

A Theory of Computation Based on Quantum Logic

Mingsheng Ying*

*State Key Laboratory of Intelligent Technology and Systems,
Department of Computer Science and Technology,
Tsinghua University, Beijing 100084, China,
Email: yingmsh@tsinghua.edu.cn*

Abstract

The (meta)logic underlying classical theory of computation is Boolean (two-valued) logic. Quantum logic was proposed by Birkhoff and von Neumann as a logic of quantum mechanics about 70 years ago. It is currently understood as a logic whose truth values are taken from an orthomodular lattice. The major difference between Boolean logic and quantum logic is that the latter does not enjoy distributivity in general. The rapid development of quantum computation in recent years stimulates us to establish a theory of computation based on quantum logic.

Finite automata and pushdown automata are two classes of the simplest mathematical models of computation. The present Chapter is a systematic exposition of automata theory based on quantum logic. We introduce the notions of orthomodular lattice-valued (quantum) finite and pushdown automaton. The classes of languages accepted by them are defined. Various properties of automata are carefully reexamined in the framework of quantum logic by employing an approach of semantic analysis, including equivalence between finite automata and regular expressions (the Kleene theorem) and equivalence between pushdown automata and context-free grammars. It is found that the universal validity of many important properties (for example, the Kleene theorem) of automata depend heavily upon the distributivity of the underlying logic. This indicates that these properties do not universally hold in the realm of quantum logic. On the other hand, we show that a local validity of them can be recovered by imposing a certain commutativity to the (atomic) statements about the automata under consideration. This reveals an essential difference between classical automata theory and automata theory based on quantum logic.

*This work was partly supported by the National Foundation of Natural Sciences of China (Grant No: 60321002, 60496321) and the Key Grant Project of Chinese Ministry of Education (Grant No: 10403)

Key Words: Quantum logic, orthomodular lattices, algebraic semantics, finite automata, regular languages, pushdown automata, context-free languages, quantum computation

Contents

1. Introduction (page 3)
2. Preliminaries (page 11)
 - 2.1. Orthomodular lattices (page 11)
 - 2.2. The syntax of quantum logic (page 20)
 - 2.3. The algebraic semantics of quantum logic (page 21)
 - 2.4. Orthomodular lattice-valued sets (page 22)
 - 2.5. Orthomodular lattice-valued relations (page 25)
 - 2.6. Orthomodular lattice-valued languages (page 26)
3. Orthomodular lattice-valued (nondeterministic) finite automata (page 28)
 - 3.1. Basic definitions and examples (page 28)
 - 3.2. Orthomodular lattice-valued deterministic finite automata (page 39)
 - 3.3. Orthomodular lattice-valued finite automata with ε -moves (page 46)
 - 3.4. Closure properties of orthomodular lattice-valued regularity (page 52)
 - 3.5. The Kleene theorem for orthomodular lattice-valued languages (page 68)
 - 3.6. The Myhill-Nerode theorem for orthomodular lattice-valued languages (page 81)
 - 3.7. The Pumping lemma for orthomodular lattice-valued regular languages (page 89)
4. Orthomodular lattice-valued pushdown automata (page 93)
 - 4.1. Orthomodular lattice-valued context-free grammars (page 93)
 - 4.2. Basic definitions of orthomodular lattice-valued pushdown automata (page 105)
 - 4.3. Equivalence of orthomodular lattice-valued context-free grammars and pushdown automata (page 111)
 - 4.4. Closure properties of orthomodular lattice-valued context-freeness (page 114)
 - 4.5. The pumping lemma for orthomodular lattice-valued context free languages (page 125)
5. Conclusion (page 128)
6. Bibliographical notes (page 133)
 - References (page 135)

1. Introduction

It is well-known that an axiomatization of a mathematical theory consists of a system of fundamental notions as well as a set of axioms about these notions. The mathematical theory is then the set of theorems which can be derived from the axioms. Obviously, one needs a certain logic to provide tools for reasoning in the derivation of these theorems from the axioms. As pointed out by A. Heyting [32] (page 5), in elementary axiomatics logic was used in a not analyzed form. Afterwards, in the studies for foundations of mathematics beginning in the early of twentieth century, it had been realized that a major part of mathematics has to exploit the full power of classical (Boolean) logic [31], the strongest one in the family of existing logics. For example, group theory is based on first-order logic, and point-set topology is built on a fragment of second-order logic. However, a few mathematicians, including the big names L. E. J. Brouwer, H. Poincare, L. Kronecker and H. Weyl, took some kind of constructive position which is in more or less explicit opposition to certain forms of mathematical reasoning used by the majority of mathematical community. Some of them even endeavored to establish so-called constructive mathematics, the part of mathematics that could be rebuilt on constructivist principles. The logic employed in the development of constructive mathematics is intuitionistic logic [77] which is truly weaker than classical logic.

Since many logics different from classical logic and intuitionistic logic have been invented in the last century, one may naturally ask the question whether we are able to establish some mathematical theories based on other nonclassical logics besides intuitionistic logic. Indeed, as early as the first nonclassical logics appeared, the possibility of building mathematics upon them was conceived. As mentioned by A. Mostowski [48], J. Lukasiewicz hoped that there would be some nonclassical logics which can be properly used in mathematics as non-Euclidean geometry does. In 1952, J. B. Rosser and A. R. Turquette [60] (page 109) proposed a similar and even more explicit idea:

“The fact that it is thus possible to generalize the ordinary two-valued logic so as not only to cover the case of many-valued statement calculi, but of many-valued quantification theory as well, naturally suggests the possibility of further extending our treatment of many-valued logic to cover the case of many-valued sets, equality, numbers, etc. Since we now have a general theory of many-valued predicate calculi, there is little doubt about the possibility of successfully developing such extended many-valued theories. ... we shall consider their careful study one of the major unsolved problems of many-valued logic.”

Unfortunately, the above idea has not attracted much attention in logical community. For such a situation, A. Mostowski [48] pointed out that most of nonclassical logics invented so far have not been really used in mathematics, and intuitionistic logic seems the unique one of nonclassical logics which still has an opportunity to

carry out the Lukasiewicz's programme. A similar opinion was also expressed by J. Dieudonne [19], and he said that mathematical logicians have been developing a variety of nonclassical logics such as second-order logic, modal logic and many-valued logic, but these logics are completely useless for mathematicians working in other research areas.

One reason for this situation might be that there is no suitable method to develop mathematics within the framework of nonclassical logics. As was pointed out above, classical logic is applied as the deduction tool in almost all mathematical theories. It should be noted that what is used in these theories is the deductive (proof-theoretical) aspect of classical logic. However, the proof theory of nonclassical logics is much more complicated than that of classical logic, and it is not an easy task to conduct reasoning in the realm of the proof theory of nonclassical logics. It is the case even for the simplest nonclassical logics, three-valued logics. This is explicitly indicated by the following excerpt from H. Hodes [33]:

“Of course three-valued logics will be somewhat more complicated than classical two-valued logic. In fact, proof-theoretically they are at least twice as complicated: But model-theoretically they are only 50 percent more complicated,....”

And much worse, some nonclassical logics were introduced only in a semantic way, and the axiomatizations of some among them are still to be found, and some of them may be not (finitely) axiomatizable. Thus, our experience in studying classical mathematics may be not suited, or at least cannot directly apply, to develop mathematics based on nonclassical logics. In the early 1990's an attempt had been made by the author [80, 81, 82] to give a partial and elementary answer in the case of point-set topology to the J. B. Rosser and A. R. Turquette's question raised above. We employed a semantical analysis approach to establish topology based on residuated lattice-valued logic, especially the Lukasiewicz system of continuous-valued logic. Roughly speaking, the semantical analysis approach transforms our intended conclusions in mathematics, which are usually expressed as implication formulas in our logical language, into certain inequalities in the truth-value lattice by truth valuation rules, and then we demonstrate these inequalities in an algebraic way and conclude that the original conclusions are semantically valid. The rich results achieved in [80, 81, 82] suggests that semantical analysis approach is an effective method to develop mathematics based on nonclassical logics.

A much more essential reason for the situation that few nonclassical logics have been applied in mathematics is absence of appealing from other subjects or applications in the real world. One major exception may be the case of quantum logic. Quantum logic was introduced by G. Birkhoff and J. von Neumann [6] in the thirties of the twentieth century as the logic of quantum mechanics. They realized that quantum mechanical systems are not governed by classical logical laws. Their proposed logic stems from von Neumann's Hilbert space formalism of quantum mechanics. The starting point was explained very well by the following excerpt from G. Birkhoff and J. von Neumann [6]:

“what logical structure one may hope to find in physical theories which, like quan-

tum mechanics, do not conform to classical logic. Our main conclusion, based on admittedly heuristic arguments, is that one can reasonably expect to find a calculus of propositions which is formally indistinguishable from the calculus of linear subspaces [of Hilbert space] with respect to set products, linear sums, and orthogonal complements - and resembles the usual calculus of propositions with respect to “and”, “or”, and “not”.”

Thus linear (closed) subspaces of Hilbert space are identified with propositions concerning a quantum mechanical system, and the operations of set product, linear sum and orthogonal complement are treated as connectives. By observing that the set of linear subspaces of a finite-dimensional Hilbert space together with these operations enjoys Dedekind’s modular law, G. Birkhoff and J. von Neumann [6] suggested to use modular lattices as the algebraic version of the logic of quantum mechanics, just like that Boolean algebras act as an algebraic counterpart of classical logic. However, the modular law does not hold in an infinite-dimensional Hilbert space. In 1937, K. Husimi [34] found a new law, called now the orthomodular law, which is valid for the set of linear subspaces of any Hilbert space. Nowadays, what is usually called quantum logic in the mathematical physics literature refers to the theory of orthomodular lattices. Obviously, this kind of quantum logic is not very logical. Indeed, there is also another much more “logical” point of view on quantum logic in which quantum logic is seen as a logic whose truth values range over an orthomodular lattice (for an excellent exposition for the latter approach of quantum logic, see [13, 14, 16]).

After the invention of quantum logic, quite a few mathematicians have tried to establish mathematics based on quantum logic. Indeed, J. von Neumann [49] himself proposed the idea of considering a quantum set theory, corresponding to quantum logic, as does classical set theory to classical logic. One important contribution in this direction was made by G. Takeuti [72]. His main idea was explained, and the nature of mathematics based on quantum logic was analyzed very well by the following citation from the introduction of [72]:

“Since quantum logic is an intrinsic logic, i.e. the logic of the quantum world, it is an important problem to develop mathematics based on quantum logic, more specifically set theory based on quantum logic. It is also a challenging problem for logicians since quantum logic is drastically different from the classical logic or the intuitionistic logic and consequently mathematics based on quantum logic is extremely difficult. On the other hand, mathematics based on quantum logic has a very rich mathematical content. This is clearly shown by the fact that there are many complete Boolean algebras inside quantum logic. For each complete Boolean algebra B , mathematics based on B has been shown by our work on Boolean valued analysis to have rich mathematical meaning. Since mathematics based on B can be considered as a sub-theory of mathematics based on quantum logic, there is no doubt about the fact that mathematics based on quantum logic is very rich. The situation seems to be the following. Mathematics based on quantum logic is too gigantic to see through clearly.”

The main technical result of G. Takeuti [72] is a construction of orthomodular lattice-valued universe $V^{P(H)}$, where H is a Hilbert space, and $P(H)$ is the orthomodular lattice consisting of all closed linear subspaces of H . He built up such an universe in a way similar to Boolean-valued models of $ZF + AC$, and showed that a reasonable set theory, including some axioms from $ZF + AC$ or their slight modifications, holds in this universe. Also, he [71] defined real numbers in $V^{P(H)}$ and showed that observables in quantum physics can be represented by such numbers. Furthermore, Titani and Kozawa [74] provided a representation of unitary operators by complex numbers in $V^{P(H)}$. Recently, another formal number theory based on quantum logic was introduced by Tokuo [75]. A different attempt of developing a theory of quantum sets was made by K. -G. Schlesinger [65] using a categorical approach in the spirit of topos theory. He started with the category of complex (pre-)Hilbert spaces and linear maps. This category was seen as the (basic) quantum set universe. Then he was able to introduce the analog of number systems and to deal with the analog of some algebraic structures in quantum set theory. Indeed, K. -G. Schlesinger's terminal goal is to build a quantum mathematics, i.e., a mathematical theory where all the ingredients (like logic and set theory) adhere to the rules of quantum mechanics. According to his proposal, quantum set theory is the quantization of the mathematical theory of pure objects, and so it is just the first step toward his goal. It is worth noting that the role of quantum logic in such a quantum mathematics is different from that in G. Takeuti's quantum set theory, and quantum logic appears as an internal logic in the former.

After a careful examination on the development of mathematics based on non-classical logics, we now come to explore the possibility of establishing a theory of computation based on a nonclassical logic. A formal formulation of the notion of computation is one of the greatest scientific achievements in the twentieth century. Since the middle of 1930's, various models of computation have been introduced, such as Turing machines, Post systems, λ -calculus and μ -recursive functions. In classical computing theory, these models of computation are investigated in the framework of classical logic; more explicitly, all properties of them are deduced by classical logic as a (meta)logical tool. So, it is reasonable to say that classical computing theory is a part of classical mathematics. Knowing the basic idea of mathematics based on nonclassical logics, we may naturally ask the question: is it possible to build a theory of computation based on a nonclassical logic, and what are the same of and difference between the properties of the models of computation in classical logic and the corresponding ones in nonclassical logics? There has been a very big population of nonclassical logics. Of course, it is unnecessary to construct models of computation in each nonclassical logic and to compare them with the ones in classical logic because some nonclassical logics are completely irrelative to behaviors of computation. Nevertheless, as will be explained shortly, it is absolutely worth studying deeply and systematically models of computation based on quantum logic.

It seems that both points of views on quantum logic mentioned above have no obvious links to computations; but appearance of the idea of quantum computers

changed dramatically the long-standing situation. The idea of quantum computation came from the studies of connections between physics and computation. The first step toward it was the understanding of the thermodynamics of classical computation. In 1973, C. H. Bennet [3] noted that a logically reversible operation does not need to dissipate any energy and found that a logically reversible Turing machine is a theoretical possibility. In 1980, further progress was made by P. A. Benioff [2] who constructed a quantum mechanical model of Turing machine. His construction is the first quantum mechanical description of computer, but it is not a real quantum computer. It should be noted that in P. A. Benioff's model between computation steps the machine may exist in an intrinsically quantum state, but at the end of each computation step the tape of the machine always goes back to one of its classical states. Quantum computers were first envisaged by R. P. Feynman [24, 25]. In 1982, he [24] conceived that no classical Turing machine could simulate certain quantum phenomena without an exponential slowdown, and so he realized that quantum mechanical effects should offer something genuinely new to computation. Although R. P. Feynman proposed the idea of universal quantum simulator, he did not give a concrete design of such a simulator. His ideas were elaborated and formalized by D. Deutsch in a seminal paper [17]. In 1985, D. Deutsch described the first true quantum Turing machine. In his machine, the tape is able to exist in quantum states too. This is different from P. A. Benioff's machine. In particular, D. Deutsch introduced the technique of quantum parallelism by which quantum Turing machine can encode many inputs on the same tape and perform a calculation on all the inputs simultaneously. Furthermore, he proposed that quantum computers might be able to perform certain types of computation that classical computers can only perform very inefficiently. One of the most striking advances was made by P. W. Shor [68] in 1994. By exploring the power of quantum parallelism, he discovered a polynomial-time algorithm on quantum computers for prime factorization of which the best known algorithm on classical computers is exponential. In 1996, L. K. Grover [28] offered another apt killer of quantum computation, and he found a quantum algorithm for searching a single item in an unsorted database in square root of the time it would take on a classical computer. Since both prime factorization and database search are central problems in computer science and the quantum algorithms for them are highly faster than the classical ones, P. W. Shor and L. K. Grover's works stimulated an intensive investigation on quantum computation. After that, quantum computation has been an extremely exciting and rapidly growing field of research.

The current studies of quantum computation may be roughly divided into five areas: (1) physical implementations; (2) physical models; (3) mathematical models; (4) quantum algorithms and complexity; and (5) quantum programming languages. Almost all pioneer works such as [2, 24, 17] in this field were devoted to build physical models of quantum computing. In 1990's, a great attention was paid to the physical implementation of quantum computation. For example, S. Lloyd [43] considered the practical implementation by using electromagnetic pulses and J. I. Cirac and P. Zoller [10] used laser manipulations of cold trapped ions to implement quantum

computing. Of course, physical implementation is still and will continue to be one of the most important problems in the area of quantum computation before quantum computers come into truth.

Quantum programming is an emerging area in recent years. Some imperative quantum programming languages was introduced by B. Ómer [53], J. W. Sanders and P. Zuliani [61], and S. Bettelli, T. Calarco and L. Serafini [5], and a functional quantum programming language was defined by P. Selinger [67]. Semantics of the quantum programming languages presented in [61, 67] have been carefully examined. More generally, it was already proposed by the UK computing research committee as one of the grand challenges for computing research to rework and to extend the whole of classical software engineering into the quantum domain, and finally, to develop a mature discipline of quantum software engineering [36].

The theoretical concerns in the computer science community have mainly been given to quantum algorithms and complexity (see [69] for a brief survey of quantum algorithms, an explanation of why quantum algorithms are so hard to be discovered, and some suggested lines of research to find new quantum algorithms). But also there have been a few attempts to develop mathematical models of quantum computation and to clarify the relationship between different models. For example, except quantum Turing machines, D. Deutsch [18] also proposed the quantum circuit model of computation, and A. C. Yao [79] showed that the quantum circuit model is equivalent to the quantum Turing machine in the sense that they can simulate each other in polynomial time. A quantum generalization of λ -calculus, another model of sequential computation, was introduced by A. V. Tonder [76], and quantum process algebras have also been proposed to model quantum concurrent computation and quantum communicating systems [27, 42]. As is well known, in classical computing theory, there are still two important classes of automata rather than Turing machines; namely, finite automata and pushdown automata. They have been widely applied in the design and implementation of programming languages. Since finite automata and pushdown automata are equipped with finite memory or finite memory with stack, respectively, they have weaker computing power than Turing machines. J. P. Crutchfield and C. Moore [11], A. Kondacs and J. Watrous [40], and S. Gudder [29] tried to introduce some quantum devices corresponding to these weaker models of computation.

In a sense, the mathematical models of quantum computation can be seen as abstractions of its physical models. It should be noted that the theoretical models of quantum computation mentioned above, including quantum Turing machines and quantum automata, are still developed in classical (Boolean) logic. Thus, their logical basis is the same as that of classical computation, and we may argue that sometimes these models might be not suitable for reasoning about quantum computing systems that obey some logical laws different from that in Boolean logic. Indeed, V. Vedral and M. B. Plenio [78] already advocated that quantum computers require “quantum logic”, something fundamentally different to classical Boolean logic. As stated above, quantum logic has been existing for a long time. So, the

point is how to apply quantum logic in the analysis and design of quantum computing systems. The background exposed above highly motivates us to explore the possibility of establishing a theory of computation based on quantum logic. Such a computing theory may be thought of as a logical foundation of quantum computation and a further abstraction of its mathematical models. We can imagine that the relation between mathematical models of quantum computation and computing theory based on quantum logic is quite similar to that between J. von Neumann's Hilbert space formalism of quantum mechanics and quantum logic.

The author and his colleagues have tried to build a theory of computation based on quantum logic for years. Since finite automata and pushdown automata are the simplest models of computation (with finite memory), they have focused their attention on developing automata theory based on quantum logic as the first step [83, 84, 44, 45, 55, 9]. The purpose of this Chapter is to give a systematic exposition of such a new theory and to clarify the relationship between it and some related works.

The present Chapter is organized as follows. Section 2 is a preliminary section. In this section, we recall some basic notions and results of orthomodular lattices. The syntax of first-order quantum logic is presented. An algebraic semantics of quantum logic is given in terms of orthomodular lattices. Then orthomodular lattice-valued (quantum) set theory is briefly reviewed. Finally, the notion of orthomodular lattice-valued language is introduced, and various operations of orthomodular lattice-valued languages are defined.

Section 3 is devoted to a systematic development of the theory of finite automata and regular languages in the framework of quantum logic. In this section, we introduce the notion of orthomodular lattice-valued (quantum) nondeterministic finite automaton and its various variants, including orthomodular lattice-valued deterministic automaton and finite automaton with ε -moves. The (orthomodular lattice-valued) languages accepted by orthomodular lattice-valued finite automata are defined according to two different principles of treating interactions between conjunction and disjunction: the depth-first one and the width-first one. It is here interesting to observe that a single notion of acceptance in the classical theory of automata splits into two nonequivalent notions in the framework of quantum logic. Indeed, both of them are natural generalizations of classical acceptance, and non-equivalence between them arises from lack of distributivity in quantum logic. It is shown that they are equivalent if and only if the underlying logic degenerates to classical Boolean logic.

With each orthomodular lattice-valued generalization of acceptance, a straightforward quantum logical generalization of regularity in the classical theory of automata can be proposed, and it is called noncommutative regularity. Unfortunately, in terms of noncommutative regularity, some important properties of finite automata cannot be generalized into the setting of quantum logic; for example, the pumping lemma. This forces us to introduce a more reasonable orthomodular lattice-valued generalization of the regularity predicate on languages: commutative regularity. It

is just noncommutative regularity plus a certain commutator. Again, a single notion of regularity in classical automata theory splits into two nonequivalent ones in quantum logic.

Orthomodular lattice-valued regularity provides us with a framework in which various properties of finite automata can be reexamined within quantum logic. The acceptance ability of orthomodular lattice-valued nondeterministic finite automata is then compared with that of their various variants. Furthermore, the closure properties of orthomodular lattice-valued regular languages are derived. We introduce the notion of orthomodular lattice-valued regular expression. A generalization of the Kleene theorem about equivalence of regular expressions and finite automata is established in quantum logic. Also, the Myhill-Nerode theorem, which characterizes regular languages in terms of certain congruence relations between words, is extended to orthomodular lattice-valued languages. A pumping lemma for orthomodular lattice-valued regular languages is finally presented.

The aim of Section 4 is to develop systematically the theory of pushdown automata and context-free languages based on quantum logic. The notions of orthomodular lattice-valued context-free grammar and pushdown automaton are proposed in this section. The languages generated by orthomodular lattice-valued context-free grammars and accepted by orthomodular lattice-valued pushdown automata are defined. Then the predicate of context-freeness may be directly generalized to orthomodular lattice-valued languages. Similar to the case of acceptance by finite automata, two different ways are allowed in defining these classes of languages, depending on the depth-first principle or the width-first principle is employed for the treatment of interactions between conjunction and disjunction. The non-equivalence of these two ways is also due to the fact that quantum logic is not distributive. Again, we are able to show that they are equivalent if and only if the meta-logic degenerates to classical Boolean logic.

We carefully rebuild various properties of context-free languages and pushdown automata within the realm of quantum logic. In particular, the closure properties of orthomodular lattice-valued context-freeness under some operations of languages are proved, and equivalence between orthomodular lattice-valued context-free grammars and pushdown automata is verified. It should be pointed out that many properties of context-free languages and pushdown automata can be easily generalized into quantum logic when the depth-first principle is applied, whereas a certain commutator must be imposed for the same purpose when we adopt the width-first principle. This is an interesting phenomenon of which we cannot find any hint in classical automata theory.

Section 5 concludes this Chapter. In particular, we try to point out some interesting implications of the essential difference between classical automata theory and automata theory based on quantum logic and to give a physical interpretation for it. Also, we provide some interesting problems for further studies.

2. Preliminaries

The aim of this section is to recall some basic notions and results about quantum logic and quantum set theory needed in the subsequent sections from the previous literature and to fix notations.

Quantum logic is in this Chapter understood as a complete orthomodular lattice-valued logic. This section is mainly concerned with the semantic aspect of such a logic, and it will be divided into six subsections. The first subsection will briefly review some fundamental results on orthomodular lattices; for more details, we refer to [39] and [8]. Some new lemmas on implication operators in orthomodular lattices will be presented too. They are crucial in the proofs of several main results in this Chapter. In the second one we shall introduce the language of first-order quantum logic. The third will present an algebraic semantics of first-order quantum logic in terms of orthomodular lattices. In the fourth subsection, orthomodular lattice-valued (quantum) sets will be introduced and some of their useful properties will be given; see [72] for details. In the fifth subsection, orthomodular lattice-valued relations and their composition and reflexive and transitive closure will be defined. Finally, in the sixth subsection we shall introduce orthomodular lattice-valued generalizations of languages in automata theory as well as some of their operations.

2.1. Orthomodular Lattices

The set of truth values of a quantum logic will be taken to be an orthomodular lattice. So we first introduce the notion of orthomodular lattice. An ortholattice is a 7-tuple $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$, where:

(1) $\langle L, \leq, \wedge, \vee, 0, 1 \rangle$ is a bounded lattice, $0, 1$ are the least and greatest elements of L , respectively, \leq is the partial ordering in L , and for any $a, b \in L$, $a \wedge b$, and $a \vee b$ stand for the greatest lower bound and the least upper bound of a and b , respectively;

(2) \perp is a unary operation on L , called orthocomplement, and required to satisfy the following conditions: for any $a, b \in L$,

$$(2.1) \quad a \wedge a^\perp = 0, \quad a \vee a^\perp = 1;$$

$$(2.2) \quad a^{\perp\perp} = a; \text{ and}$$

$$(2.3) \quad a \leq b \text{ implies } b^\perp \leq a^\perp.$$

It is easy to see that the condition (2.3) is equivalent to one of the De Morgan laws: for any $a, b \in L$,

$$(a \wedge b)^\perp = a^\perp \vee b^\perp, \quad (a \vee b)^\perp = a^\perp \wedge b^\perp.$$

Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an ortholattice, and let $a, b \in L$. We say that a commutes with b , in symbols aCb , if $a = (a \wedge b) \vee (a \wedge b^\perp)$.

An orthomodular lattice is an ortholattice $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ satisfying the orthomodular law: for all $a, b \in L$,

$$a \leq b \text{ implies } a \vee (a^\perp \wedge b) = b.$$

The orthomodular law can be replaced by the following equation:

$$a \vee (a^\perp \wedge (a \vee b)) = a \vee b \text{ for any } a, b \in L.$$

A Boolean algebra is an ortholattice $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ fulfilling the distributive law of join over meet: for all $a, b, c \in L$,

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c).$$

With the De Morgan law it is easy to know that this condition is equivalent to the distributive law of meet over join: for any $a, b, c \in L$,

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c).$$

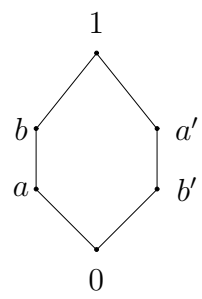
Obviously, the distributive law implies the orthomodular law, and so a Boolean algebra is an orthomodular lattice.

The following lemma gives a characterization of orthomodular lattices and it distinguishes orthomodular lattices from ortholattices.

Lemma 2.1. ([8], Propositions 2.1 and 2.2) Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an ortholattice. Then the following seven statements are equivalent:

- (1) ℓ is an orthomodular lattice;
- (2) For any $a, b \in L$, if $a \leq b$ and $a^\perp \wedge b = 0$ then $a = b$;
- (3) For any $a, b \in L$, if aCb then bCa ;
- (4) For any $a, b \in L$, if aCb then $a^\perp Cb$;
- (5) For any $a, b \in L$, if aCb then $a \vee (a^\perp \wedge b) = a \vee b$;
- (6) The benzene ring O_6 (see Figure 1) is not a subalgebra of ℓ ;
- (7) For any $a, b \in L$, if $a \leq b$ then the subalgebra $[\{a, b\}]$ of ℓ generated by a and b is a Boolean algebra. \square

The set of truth values of classical logic is a Boolean algebra; whereas quantum logic is an orthomodular lattice-valued logic. It is well-known that a Boolean algebra must be an orthomodular lattice, but the inverse is not true. Thus, quantum logic is weaker than classical logic. The major difference between a Boolean algebra and an orthomodular lattice is that distributivity is not valid in the latter. However, many cases still appeal an application of the distributivity even when we manipulate



elements in an orthomodular lattice. This requires us to regain a certain (weaker) version of distributivity in the realm of orthomodular lattices. The key technique for this purpose is commutativity which is able to provide a localization of distributivity. The following lemma together with Lemma 2.1(4) indicates that commutativity is preserved by all operations of orthomodular lattice.

Lemma 2.2. ([8], Proposition 2.4) Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let $a \in L$ and $b_i \in L$ ($i \in I$). If aCb_i for any $i \in I$, then

$$aC\left(\bigwedge_{i \in I} b_i\right) \text{ and } aC\left(\bigvee_{i \in I} b_i\right)$$

provided $\bigwedge_{i \in I} b_i$ and $\bigvee_{i \in I} b_i$ exist. \square

The local distributivity implied by commutativity is then given by the following lemma.

Lemma 2.3. ([8], Proposition 2.3) Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice. For any $a \in L$ and $b_i \in L$ ($i \in I$), if aCb_i for all $i \in I$, then

$$a \wedge \left(\bigvee_{i \in I} b_i\right) = \bigvee_{i \in I} (a \wedge b_i),$$

$$a \vee \left(\bigwedge_{i \in I} b_i\right) = \bigwedge_{i \in I} (a \vee b_i)$$

provided $\bigwedge_{i \in I} b_i$ and $\bigvee_{i \in I} b_i$ exist. \square

The above lemma is very useful, and it often enables us to recover distributivity in an orthomodular lattice. However, its condition that all elements involved commute each other is quite strong, and not easy to meet. This suggests us to find a way to weaken this condition. One solution was found by G. Takeuti [72], and he introduced the notion of commutator which can be seen as an index measuring the degree to which the commutativity is valid.

Definition 2.4. ([72], pages 305 and 307) Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let $A \subseteq L$.

(1) If A is finite, then the commutator $\gamma(A)$ of A is defined by

$$\gamma(A) = \bigvee_{a \in A} \left\{ \bigwedge_{a \in A} a^{f(a)} : f \text{ is a mapping from } A \text{ into } \{1, -1\} \right\},$$

where a^1 denotes a itself and a^{-1} denotes a^\perp .

(2) The strong commutator $\Gamma(A)$ of A is defined by

$$\Gamma(A) = \bigvee \{b : aCb \text{ for all } a \in A, \text{ and } (a_1 \wedge b)C(a_2 \wedge b) \text{ for all } a_1, a_2 \in A\}.$$

The relation between commutator and strong commutator is clarified by the following lemma. In addition, the third item of the following lemma shows that commutator is a relativization of the notion of commutativity.

Lemma 2.5. ([72], Proposition 4 and its corollary) Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice and let $A \subseteq L$. Then

- (1) $\Gamma(A) \leq \gamma(A)$.
- (2) If A is finite, then $\Gamma(A) = \gamma(A)$.
- (3) $\gamma(A) = 1$ if and only if all the members of A are mutually commutable. \square

We now can present a generalization of Lemma 2.3 by using the tool of commutator. It is easy to see from Lemmas 2.5(2) and (3) that the following lemma degenerates to Lemma 2.3 whenever aCb_i for all $i \in I$.

Lemma 2.6. ([72], Propositions 5 and 6) Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice and let $A \subseteq L$. Then for any $a \in A$ and $b_i \in A$ ($i \in I$),

$$\Gamma(A) \wedge (a \wedge \bigvee_{i \in I} b_i) \leq \bigvee_{i \in I} (a \wedge b_i),$$

$$\Gamma(A) \wedge \bigwedge_{i \in I} (a \vee b_i) \leq a \vee \bigwedge_{i \in I} b_i. \quad \square$$

Suppose that we want to use the above lemma on a formula of the form $a \wedge (\bigvee_{i \in I} b_i)$ or $a \vee (\bigwedge_{i \in I} b_i)$ in order to get a local distributivity. In many situations, the elements a and b_i ($i \in I$) may be very complicated, and the operations \perp , \wedge and \vee are involved in them. Then the above lemma cannot be applied directly, and it needs the help of the following

Lemma 2.7. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice and let $A \subseteq L$. Then for any $B \subseteq [A]$ we have $\Gamma(A) \leq \Gamma(B)$, where $[A]$ stands for the subalgebra of ℓ generated by A .

Proof. For any $X \subseteq L$, we write

$$K(X) = \{b \in L : aCb \text{ and } (a_1 \wedge b)C(a_2 \wedge b) \text{ for all } a, a_1, a_2 \in X\}$$

Furthermore, we set $A_0 = A$ and

$$A_{i+1} = A_i \cup \{a^\perp : a \in A_i\} \cup \{a_1 \wedge a_2 : a_1, a_2 \in A_i\} \quad (i = 0, 1, 2, \dots)$$

First, we prove that $K(A_i) = K(A)$ for all $i \geq 0$ by induction on i . It is obvious that $K(A_{i+1}) \subseteq K(A)$. Conversely, suppose that $b \in K(A)$ and we want to show that $b \in K(A_{i+1})$. It is easy to see that aCb for any $a \in A_{i+1}$. Thus, we only need to demonstrate the following

Claim: $(a_1 \wedge b)C(a_2 \wedge b)$ for any $a_1, a_2 \in A_{i+1}$.

The essential part of the proof of the above claim is the following two cases, and the other cases are clear, or can be treated as iterations of them:

Case 1. $a_1 \in A_i$, $a_2 = c_1 \wedge c_2$ and $c_1, c_2 \in A_i$. From the induction hypothesis we have

$$(a_1 \wedge b)C(c_1 \wedge b), \text{ and } (a_1 \wedge b)C(c_2 \wedge b).$$

This yields

$$(a_1 \wedge b)C(c_1 \wedge b) \wedge (c_2 \wedge b) = (c_1 \wedge c_2) \wedge b = a_2 \wedge b.$$

Case 2. $a_1 \in A_i$, $a_2 = c^\perp$ and $c \in A_i$. Then from the induction hypothesis we obtain $(a_1 \wedge b)C(c \wedge b)$, and further $(a_1 \wedge b)C(c \wedge b)^\perp$ by using Lemma 2.1(4). In addition, $(a_1 \wedge b)Cb$. This together with Lemma 2.2 yields $(a_1 \wedge b)C[b \wedge (c \wedge b)^\perp]$. Note that cCb and so $b^\perp Cc^\perp$. Then by Lemma 2.3 we assert that

$$b^\perp \vee (c \wedge b) = b^\perp \vee c \text{ and } b \wedge (c \wedge b)^\perp = b \wedge c^\perp.$$

Hence, it follows that $(a_1 \wedge b)Cb \wedge c^\perp = a_2 \wedge b$.

We now write

$$A_\infty = \bigcup_{i=0}^{\infty} A_i.$$

Then

$$K(A_\infty) = \bigcap_{i=0}^{\infty} K(A_i) = K(A).$$

It is easy to see that $A \subseteq A_\infty$ is a subalgebra of ℓ . So, $[A] \subseteq A_\infty$,

$$K(A) = K(A_\infty) \subseteq K([A]) \subseteq K(B),$$

and

$$\Gamma(A) = \bigvee K(A) \leq \bigvee K(B) = \Gamma(B). \quad \square$$

As stated in the introduction, the aim of this paper is to develop a theory of computation based on quantum logic. The logical language for a theory of computation has to contain the universal and existential quantifiers, and the two quantifiers are usually interpreted as (infinite) meet and join, respectively. Hence, we should assume that the lattice of the truth values of our quantum logic is complete. A

complete orthomodular lattice is an orthomodular lattice $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ in which for any $M \subseteq L$, both the greatest lower bound $\bigwedge M$ and the least upper bound $\bigvee M$ exist.

The function of a logic is to provide us with a certain reasoning ability, and the implication connective is an intrinsic representative of inference within the logic. Thus each logic should reasonably contain a connective of implication. To make a complete orthomodular lattice available as the set of truth values of quantum logic, we need to define a binary operation, called implication operator, on it such that this operation may serve as the interpretation of implication in this logic. Unfortunately, it is a very vexed problem to define a reasonable implication operator for quantum logic. All implication operators that one can reasonably introduce in an orthomodular lattice are more or less anomalous in the sense that they do not share most of the fundamental properties of the implication in classical logic. This is different from the cases of most weak logics. (For a thorough discussion on the implication problem in quantum logic, see [13], Section 3.)

An implication operator is defined to be a mapping \rightarrow from $L \times L$ into L . A minimal condition for it is the requirement proposed by G. Birkhoff and J. von Neumann [6]:

$$a \rightarrow b = 1 \text{ if and only if } a \leq b$$

for any $a, b \in L$. Usually in a logic, there are two ways in which implication is introduced. The first one is to treat implication as a derived connective; that is, implication is explicitly defined in terms of other connectives such as negation, conjunction and disjunction. All implications of this kind were found by G. Kalmbach [38], and they are presented by the following

Lemma 2.8. ([38]; see also [39], Theorem 15.3) The orthomodular lattice freely generated by two elements is isomorphic to $2^4 \times MO2$, where 2 stands for the Boolean algebra of two elements, and $MO2$ is the lattice called ‘‘Chinese lantern’’ (for a detailed description, see Example 3.8 below). The elements of $2^4 \times MO2$ satisfying the Birkhoff-von Neumann requirement are exactly the following five polynomials of two variables:

$$a \rightarrow_1 b = (a^\perp \wedge b) \vee (a^\perp \wedge b^\perp) \vee (a \wedge (a^\perp \vee b)),$$

$$a \rightarrow_2 b = (a^\perp \wedge b) \vee (a \wedge b) \vee ((a^\perp \vee b) \wedge b^\perp),$$

$$a \rightarrow_3 b = a^\perp \vee (a \wedge b),$$

$$a \rightarrow_4 b = b \vee (a^\perp \wedge b^\perp),$$

$$a \rightarrow_5 b = (a^\perp \wedge b) \vee (a \wedge b) \vee (a^\perp \wedge b^\perp). \quad \square$$

Obviously, this lemma implies that the above five polynomials are all implication operators definable in orthomodular lattices. It was shown by G. Kalmbach [38, 39] that the orthomodular lattice-valued (propositional) logic can be (finitely)

axiomatizable by using the modus ponens with implication \rightarrow_1 as the only one rule of inference, but the same conclusion does not hold for the other implications \rightarrow_i ($2 \leq i \leq 5$).

We may also define the material conditional \rightarrow_0 in an orthomodular lattice $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ by

$$a \rightarrow_0 b = a^\perp \vee b$$

for all $a, b \in L$. It is easy to see that \rightarrow_0 does not fulfil the Birkhoff-von Neumann requirement. On the other hand, the following lemma shows that the five implication operators given in Lemma 2.8 degenerate to the material conditional whenever the two operands are compatible.

Lemma 2.9. ([13], Theorem 3.2) Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice. Then for any $a, b \in L$, $a \rightarrow_i b = a \rightarrow_0 b$ if and only if aCb , where $1 \leq i \leq 5$. \square

The second way of defining an implication is to take its truth function as the adjunction (i.e., residuation) of the truth function of conjunction. Note that in this case the implication is usually not definable from negation, conjunction and disjunction, and it has to be treated as a primitive connective. Indeed, L. Herman, E. Marsden and R. Piziak [35] introduced an implication in the style of residuation. Furthermore, the following lemma shows that the five polynomial implication operators \rightarrow_i ($1 \leq i \leq 5$) cannot be defined as the residuation of the conjunction unless ℓ is a Boolean algebra.

Lemma 2.10. ([14], page 148) Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let $1 \leq i \leq 5$. Then the following two statements are equivalent:

- (i) ℓ is a Boolean algebra.
- (ii) The import-export law: for all $a, b \in L$, $a \wedge b \leq c$ if and only if $a \leq b \rightarrow_i c$. \square

Among the five orthomodular polynomial implications, \rightarrow_3 , named the Sasaki-hook, has often been preferred since it enjoys some properties resembling those in intuitionistic logic. The Sasaki-hook was originally introduced by P. D. Finch [26]. For a detailed discussion of the Sasaki-hook, see L. Román and B. Rumbos [58] and L. Román and R. E. Zuazua [59]. Here we first point out that the Sasaki-hook possesses a modification of residual characterization although it is defined as a polynomial in orthomodular lattice. A weakening of the import-export law is the resulting condition, called compatible import-export law, by restricting the import-export law for any $a, b \in L$ with aCb ; that is, if aCb , then $a \wedge b \leq c$ if and only if $a \leq b \rightarrow c$.

Lemma 2.11. ([72], Proposition 1 and its corollary; [14]) Let $\ell = \langle L, \leq$

, $\wedge, \vee, \perp, 0, 1$) be an orthomodular lattice, and let $a, b \in L$. Then

$$a \rightarrow_3 b = \bigvee \{x : xCa \text{ and } x \wedge a \leq b\}.$$

Moreover, among the five implications \rightarrow_i ($1 \leq i \leq 5$), the Sasaki-hook \rightarrow_3 is the only one satisfying the compatible import-export law. \square

Our mathematical reasoning frequently requires that implication relation is preserved by conjunction and disjunction. Also, the negation is needed to be compatible with implication in the sense that the negation can reverse the direction of implication. And, to warrant the validity of a chain of inferences, the transitivity of implication is required. However, this is not the case in general if we are working in an orthomodular lattice. Fortunately, if we adopt the Sasaki-hook, then these properties of implication can be recovered by attaching a certain commutator.

Lemma 2.12. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice. Then

(1) for any $a_i, b_i \in L$ ($i = 1, \dots, n$), let $X = \{a_1, \dots, a_n\} \cup \{b_1, \dots, b_n\}$,

$$\Gamma(X) \wedge \bigwedge_{i=1}^n (a_i \rightarrow_3 b_i) \leq \bigwedge_{i=1}^n a_i \rightarrow_3 \bigwedge_{i=1}^n b_i,$$

$$\Gamma(X) \wedge \bigwedge_{i=1}^n (a_i \rightarrow_3 b_i) \leq \bigvee_{i=1}^n a_i \rightarrow_3 \bigvee_{i=1}^n b_i.$$

(2) for any $a, b \in L$,

$$\Gamma(a, b) \wedge (a \rightarrow_3 b) \leq b^\perp \rightarrow_3 a^\perp.$$

(3) for any $a, b, c \in L$,

$$\Gamma(a, b, c) \wedge (a \rightarrow_3 b) \wedge (b \rightarrow_3 c) \leq a \rightarrow_3 c.$$

(4) for any $a, b \in L$,

$$\Gamma(a, b) \wedge b \leq a \rightarrow_3 b.$$

(5) for any $a, b \in L$,

$$\Gamma(a, b) \wedge a \wedge (a \rightarrow_3 b) \leq b.$$

Proof. (1) We only prove the first inequality, and the proof of the second is similar. With Lemmas 2.6 and 2.7 we obtain:

$$\begin{aligned}
\bigwedge_{i=1}^n a_i \rightarrow_3 \bigwedge_{i=1}^n b_i &= (\bigwedge_{i=1}^n a_i)^\perp \vee (\bigwedge_{i=1}^n a_i \wedge \bigwedge_{i=1}^n b_i) \\
&= \bigvee_{i=1}^n a_i^\perp \vee \bigwedge_{i=1}^n (a_i \wedge b_i) \\
&\geq \Gamma(X) \wedge \bigwedge_{i=1}^n (\bigvee_{j=1}^n a_j^\perp \vee (a_i \wedge b_i)) \\
&\geq \Gamma(X) \wedge \bigwedge_{i=1}^n (a_i^\perp \vee (a_i \wedge b_i)) \\
&= \Gamma(X) \wedge \bigwedge_{i=1}^n (a_i \rightarrow b_i).
\end{aligned}$$

(2) First, we note that

$$a \wedge b, a^\perp \wedge b, a^\perp \wedge b^\perp \leq b \vee (a^\perp \wedge b^\perp) = b^\perp \rightarrow_3 a^\perp.$$

Thus, it follows that

$$\begin{aligned}
\Gamma(a, b) &= (a \wedge b) \vee (a \wedge b^\perp) \vee (a^\perp \wedge b) \vee (a^\perp \wedge b^\perp) \\
&\leq (b^\perp \rightarrow_3 a^\perp) \vee (a \wedge b^\perp),
\end{aligned}$$

and furthermore with Lemmas 2.6 and 2.7 we have:

$$\begin{aligned}
\Gamma(a, b) \wedge (a \rightarrow_3 b) &= \Gamma(a, b) \wedge (a^\perp \vee (a \wedge b)) \\
&\leq \Gamma(a, b) \wedge (a^\perp \vee b) \\
&= \Gamma(a, b) \wedge \Gamma(a, b) \wedge (a^\perp \vee b) \\
&\leq \Gamma(a, b) \wedge [(b^\perp \rightarrow_3 a^\perp) \vee (a \wedge b^\perp)] \wedge (a^\perp \vee b) \\
&\leq [(b^\perp \rightarrow_3 a^\perp) \wedge (a^\perp \vee b)] \vee [(a \wedge b^\perp) \wedge (a^\perp \vee b)] \\
&\leq (b^\perp \rightarrow_3 a^\perp) \vee [(a \wedge b^\perp) \wedge (a^\perp \vee b)].
\end{aligned}$$

Note that $(a \wedge b^\perp)^\perp = a^\perp \vee b$ and $(a \wedge b^\perp) \wedge (a^\perp \vee b) = 0$. Then it holds that

$$\Gamma(a, b) \wedge (a \rightarrow_3 b) \leq b^\perp \rightarrow_3 a^\perp.$$

(3) Again, we use Lemmas 2.6 and 2.7. This enables us to assert that

$$\begin{aligned}
\Gamma(a, b, c) \wedge (a \rightarrow_3 b) \wedge (b \rightarrow_3 c) &= \Gamma(a, b, c) \wedge (a^\perp \vee (a \wedge b)) \wedge (b^\perp \vee (b \wedge c)) \\
&\leq \Gamma(a, b, c) \wedge [(a^\perp \wedge (b^\perp \vee (b \wedge c))] \vee [(a \wedge b) \wedge (b^\perp \vee (b \wedge c))] \\
&\leq \Gamma(a, b, c) \wedge (a^\perp \vee [(a \wedge b) \wedge (b^\perp \vee (b \wedge c))]).
\end{aligned}$$

We note that $\Gamma(a, b, c)Ca^\perp$ and $\Gamma(a, b, c)C[(a \wedge b) \wedge (b^\perp \vee (b \wedge c))]$. Then it yields:

$$\begin{aligned}
\Gamma(a, b, c) \wedge (a \rightarrow_3 b) \wedge (b \rightarrow_3 c) &\leq (\Gamma(a, b, c) \wedge a^\perp) \vee (\Gamma(a, b, c) \wedge [(a \wedge b) \wedge (b^\perp \vee (b \wedge c))]) \\
&\leq a^\perp \vee (\Gamma(a, b, c) \wedge [(a \wedge b) \wedge (b^\perp \vee (b \wedge c))]) \\
&\leq a^\perp \vee [(a \wedge b) \wedge b^\perp] \vee [(a \wedge b) \wedge (b \wedge c)] \\
&= a^\perp \vee [(a \wedge b) \wedge (b \wedge c)] \\
&\leq a^\perp \vee (a \wedge c) \\
&= a \rightarrow_3 c.
\end{aligned}$$

(4) Using Lemmas 2.6 and 2.7 we obtain:

$$\begin{aligned}
\Gamma(a, b) \wedge b &\leq \Gamma(a, b) \wedge (a^\perp \vee b) = \Gamma(a, b) \wedge [(a^\perp \vee a) \wedge (a^\perp \vee b)] \\
&\leq a^\perp \vee (a \wedge b) = a \rightarrow_3 b.
\end{aligned}$$

(5) Also using Lemmas 2.6 and 2.7 we have:

$$\begin{aligned}
\Gamma(a, b) \wedge a \wedge (a \rightarrow_3 b) &= \Gamma(a, b) \wedge a \wedge [a^\perp \vee (a \wedge b)] \\
&\leq (a \wedge a^\perp) \vee (a \wedge a \wedge b) = a \wedge b \leq b. \quad \square
\end{aligned}$$

For simplicity of presentation, we finally introduce an abbreviation. For each implication operator \rightarrow , the bi-implication operator on ℓ is defined as follows:

$$a \leftrightarrow b \stackrel{def}{=} (a \rightarrow b) \wedge (b \rightarrow a)$$

for any $a, b \in L$.

2.2. The Syntax of Quantum Logic

In this subsection we present the syntax of quantum logic. Given a complete orthomodular lattice $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$, together with an implication operator \rightarrow over it. We require that the language of an ℓ -valued (quantum) logic possesses a nullary connective \mathbf{a} for each $a \in L$ as well as three other primitive connectives: an unary one \neg (negation) and two binary ones \wedge (conjunction), \rightarrow (implication). The language also has a primitive quantifier \forall (universal quantifier).

It deserves an explanation for our design decision of choosing implication as a primitive connective. In the sequel, many results only need to suppose that the implication operator satisfies the Birkhoff-von Neumann requirement. It is known that there are five polynomials fulfilling the Birkhoff-von Neumann requirement. If we treated implication as a derived connective defined in terms of negation, conjunction and disjunction, then it would be necessary to assume five different connectives of implication in our logical language. This would often complicate our presentation

very much. On the other hand, in some cases, the Birkhoff-von Neumann condition is not enough and it requires the implication operator to be the Sasaki-hook. So, we decide to use implication as a primitive connective, and specify it when needed.

The syntax of ℓ -valued logic is defined in a familiar way; we omit its details. In addition, we often need a set-theoretic language in developing our theory of computation based on quantum logic. So, a binary predicate symbol \in should be added into the syntax, and it will be interpreted as the membership relation as in classical set theory. To simplify the notations in what follows, it is necessary to introduce several derived formulas:

- (i) $\varphi \vee \psi \stackrel{def}{=} \neg(\neg\varphi \wedge \neg\psi)$;
- (ii) $\varphi \leftrightarrow \psi \stackrel{def}{=} (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$;
- (iii) $(\exists x)\varphi \stackrel{def}{=} \neg(\forall x)\neg\varphi$;
- (iv) $A \subseteq B \stackrel{def}{=} (\forall x)(x \in A \rightarrow x \in B)$; and
- (v) $A \equiv B \stackrel{def}{=} (A \subseteq B) \wedge (B \subseteq A)$.

Suppose that Δ is a finite set of formulas. The commutator of Δ is defined to be

$$\gamma(\Delta) \stackrel{def}{=} \bigvee_{\varphi \in \Delta} \{ \bigwedge \varphi^{f(\varphi)} : f \in \{1, -1\}^\Delta \},$$

where φ^1, φ^{-1} express φ and $\neg\varphi$, respectively. It is obvious that the above formula is the counterpart of Definition 2.4(1) in the language of our quantum logic.

2.3. The Algebraic Semantics of Quantum Logic

We now turn to give the semantics of quantum logic. There are several different versions of semantics for quantum logic; for example, quantum logic enjoys a semantics in the Kripke style [13, 57]. What concerns us here is its algebraic semantics. Assume that $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ is an orthomodular lattice equipped with additionally a binary operation \rightarrow over it. The operation \rightarrow is required to be suited to serve as the truth function of implication connective. According to our explanation of the connective of implication in the last subsection, we leave the operation \rightarrow unspecified but suppose that it satisfies the Birkhoff-von Neumann requirement. An ℓ -valued interpretation is an interpretation in which every predicate symbol is associated with an ℓ -valued relation, i.e., a mapping from the product of some copies of the discourse universe into L , where the number of copies is exactly the arity of the predicate symbol (see Section 2.5). The other items in ℓ -valued logical language are interpreted as usual. For every (well-formed) formula φ , its truth value is denoted by $\lceil \varphi \rceil$, and it is assumed in L . The truth valuation rules for logical and set-theoretical formulas are given as follows:

- (i) $\lceil \mathbf{a} \rceil = a$;
- (ii) $\lceil \neg\varphi \rceil = \lceil \varphi \rceil^\perp$;

- (iii) $[\varphi \wedge \psi] = [\varphi] \wedge [\psi]$;
- (iv) $[\varphi \rightarrow \psi] = [\varphi] \rightarrow [\psi]$;
- (v) if U is the universe of discourse, then

$$[(\forall x)\varphi(x)] = \bigwedge_{u \in U} [\varphi(u)];$$

(vi) $[x \in A] = A(x)$, where A is a set constant (unary predicate symbol) and it is interpreted as a mapping, also denoted as A , from the universe into L , i.e., an ℓ -valued set (more exactly, an ℓ -valued subset of the universe; see Section 2.4).

Note that in the above truth valuation rule (iii), \wedge in the left-hand side is a connective in quantum logic, whereas \wedge in the right-hand side stands for an operation in the orthomodular lattice ℓ of truth values. Also, the symbol \rightarrow in the left-hand side of (iv) is a connective in the language of quantum logic, but the symbol \rightarrow in the right-hand side of (iv) is the binary operation attached to ℓ that is explained at the beginning of this subsection.

As we claimed in the introduction, quantum logic will act as our meta-logic in the theory of automata developed in this Chapter. Then we still have to introduce several meta-logical notions for quantum logic. For every orthomodular lattice $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$, if Γ is a set of formulas and φ a formula, then φ is a semantic consequence of Γ in ℓ -valued logic, written $\Gamma \models^\ell \varphi$, whenever

$$\bigwedge_{\psi \in \Gamma} [\psi] \leq [\varphi]$$

for all ℓ -valued interpretations. In particular, $\models^\ell \varphi$ means that $\emptyset \models^\ell \varphi$, i.e., $[\varphi] = 1$ always holds for every ℓ -valued interpretation; in other words, 1 is the unique designated truth value in ℓ . Furthermore, if $\Gamma \models^\ell \varphi$ (resp. $\models^\ell \varphi$) for all orthomodular lattice ℓ then we say that φ is a semantic consequence of Γ (resp. φ is valid) in quantum logic and write $\Gamma \models \varphi$ (resp. $\models \varphi$).

We here are not going to give a detailed exposition on quantum logic, but would like to point out that quantum logic gives rise to many counterexamples to some meta-logical properties which hold for classical logic and for a large class of weaker logics; for example, M. L. Dalla Chiara [12] showed that a minimal version of quantum logic fails to enjoy the Lindenbaum property, and J. Malinowski [46] found that the deduction theorem fails in quantum logic and some of its variants.

2.4. Orthomodular Lattice-Valued Sets

Beside the language of first-order quantum logic, we will also need some notations such as \in (membership) from set-theoretical language in our study of automata theory based on quantum logic. As mentioned in the introduction, a theory of quantum sets has already been developed by G. Takeuti [72]. A careful review of quantum

set theory is out of the scope of the present Chapter. What mainly concerned G. Takeuti [72] is how some axioms of classical set theory could be modified so that they will hold in the framework of quantum logic. In other words, he tried to clarify the relation of quantum set theory with the classical mathematics. Here, we instead propose, for a given orthomodular lattice $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$, some operations of ℓ -valued sets and also introduce several notations for some special ℓ -valued sets. These are needed in the subsequent sections.

We write L^X for the set of all ℓ -valued subsets of X , i.e., all mappings from X into L . For any $A \subseteq X$, its characteristic function is a mapping from X into the Boolean algebra $\mathbf{2} = \{0, 1\}$ of two elements, and so it can also be seen as a mapping from X into L , namely, an ℓ -valued subset of X . We will identify A with its characteristic function.

For any non-empty set X , if $x \in X$ and $\lambda \in L - \{0\}$, then x_λ is defined to be a mapping from X into L such that

$$x_\lambda(x') = \begin{cases} \lambda, & \text{if } x' = x, \\ 0, & \text{otherwise,} \end{cases}$$

and it is often called an ℓ -valued point in X . We write $p_\ell(X)$ for the set of all ℓ -valued points in X ; that is, $p_\ell(X) = \{x_\lambda : x \in X \text{ and } \lambda \in L - \{0\}\}$. For each $e = x_\lambda \in p_\ell(X)$, x is called the support of e and denoted $s(e)$, and λ is called the height of e and written $h(e)$. In particular, an ℓ -valued point of height 1 is always identified with its support. The predicate \in can be extended to a predicate between ℓ -valued points and ℓ -valued sets in a natural way:

$$x_\lambda \in A \stackrel{def}{=} x_\lambda \subseteq A.$$

Then it is easy to see that $[x_\lambda \in A] = \lambda \rightarrow A(x)$ for any $x \in X$, $\lambda \in L$ and $A \in L^X$, where \rightarrow is the implication operator under consideration.

The equality and inclusion between ℓ -valued sets are defined in the usual way. Let $A, B \in L^X$. Then

$$\begin{aligned} A \subseteq B &\stackrel{def}{=} (\forall x)(x \in A \rightarrow x \in B) \\ A \equiv B &\stackrel{def}{=} (A \subseteq B) \wedge (B \subseteq A). \end{aligned}$$

From the truth valuation rules and the definition of derived formulas in the ℓ -valued logical and set-theoretical language, we know that

$$\begin{aligned} [A \subseteq B] &= \bigwedge_{x \in X} (A(x) \rightarrow B(x)), \\ [A \equiv B] &= [A \subseteq B] \wedge [B \subseteq A]. \end{aligned}$$

For any $a \in L$ and $A, B \in L^X$, we define all of the scalar product aA , complement A^c , intersection $A \cap B$ and union $A \cup B$ to be ℓ -valued subsets of X , and for all

$x \in X$,

$$\begin{aligned} x \in aA &\stackrel{def}{=} a \wedge (x \in A); \\ x \in A^c &\stackrel{def}{=} \neg(x \in A); \\ x \in A \cap B &\stackrel{def}{=} (x \in A) \wedge (x \in B); \\ x \in A \cup B &\stackrel{def}{=} (x \in A) \vee (x \in B). \end{aligned}$$

In other words, for all $x \in X$, we have:

$$\begin{aligned} (aA)(x) &= a \wedge A(x); \\ (A^c)(x) &= A(x)^\perp; \\ (A \cap B)(x) &= A(x) \wedge B(x); \text{ and} \\ (A \cup B)(x) &= A(x) \vee B(x). \end{aligned}$$

It is easy to see that in the domain of ℓ -valued sets the intersection and union operations are idempotent, commutative and associative, and they have X and ϕ , respectively as their unit elements. The intersection and union together with the complement satisfy the De Morgan law, but the distributivity of intersection over union or union over intersection is no longer valid. Clearly, the laws for operations of ℓ -valued sets are essentially determined by the algebraic properties of the lattice ℓ of truth values. We can also define Cartesian product for ℓ -valued sets. Suppose X and Y are two sets, $A \in L^X$ and $B \in L^Y$. Then for any $x \in X$ and $y \in Y$,

$$(x, y) \in A \times B \stackrel{def}{=} (x \in A) \wedge (y \in B).$$

Equivalently, it holds that $(A \times B)(x, y) = A(x) \wedge B(y)$.

Assume that X and Y are two non-empty sets, and $h : X \rightarrow Y$ is a mapping. For any $A \in L^X$, its image $h(A)$ under h is defined by

$$y \in h(A) \stackrel{def}{=} (\exists x \in X)(y = f(x) \wedge x \in A),$$

and for any $B \in L^Y$, its pre-image $h^{-1}(B)$ under h is defined by

$$x \in h^{-1}(B) \stackrel{def}{=} h(x) \in B.$$

The defining equations of $h(A)$ and $h^{-1}(B)$ may be rewritten, respectively, as follows: for any $x \in X$ and $y \in Y$,

$$h(A)(y) = \bigvee \{A(x) : x \in X \text{ and } f(x) = y\}, \text{ and}$$

$$h^{-1}(B)(x) = B(h(x)).$$

The following lemma indicates that set-theoretic equality is preserved by pre-image operator.

Lemma 2.13. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, let \rightarrow enjoy the Birkhoff-von Neumann requirement, and let $h : X \rightarrow Y$ be a mapping. Then for any $A, B \in L^Y$,

$$\stackrel{\ell}{\models} A \equiv B \rightarrow h^{-1}(A) \equiv h^{-1}(B).$$

Proof. From the truth valuation rules we may assert that

$$\begin{aligned} [h^{-1}(A) \equiv h^{-1}(B)] &= \bigwedge_{x \in X} (h^{-1}(A)(x) \leftrightarrow h^{-1}(B)(x)) \\ &= \bigwedge_{x \in X} (A(h(x)) \leftrightarrow B(h(x))) \\ &\geq \bigwedge_{y \in Y} (A(y) \leftrightarrow B(y)) \\ &= [A \equiv B]. \quad \square \end{aligned}$$

2.5. Orthomodular Lattice-Valued Relations

The notion of relation can be introduced in quantum set theory too. Let X and Y be two sets. Then an ℓ -valued (binary) relation from X to Y is an ℓ -valued subset of $X \times Y$. The proposition “ $(x, y) \in R$ ” is usually abbreviated to “ xRy ”. Suppose that R is an ℓ -valued relation from X to Y and S from Y to Z . Then their composition $R \circ S$ is defined to be an ℓ -valued relation from X to Z , and for any $x \in X$ and $z \in Z$,

$$x(R \circ S)z \stackrel{def}{=} (\exists y \in Y)(xRy \wedge ySz).$$

If R is an ℓ -valued relation from X to itself, and $n \geq 0$, then we have two different ways to define its n -power: depth-first way and width-first way. The n -power $R^{n[D]}$ of R in depth-first way is defined by

$$xR^{n[D]}y \stackrel{def}{=} (\exists z_1, \dots, z_{n-1} \in X)(xRz_1 \wedge z_1Rz_2 \wedge \dots \wedge z_{n-2}Rz_{n-1} \wedge z_{n-1}Ry)$$

for any $x, y \in X$, and the n -power $R^{n[W]}$ of R in width-first way is defined by

$$\begin{cases} R^{0[W]} = Id_X, \\ R^{n+1[W]} = R \circ R^{n[W]}, \quad n \geq 0, \end{cases}$$

where Id_X is the identity relation on X . It is easy to see that in general $R^{n[D]} = R^{n[W]}$ does not hold provided $n \geq 3$. On the other hand, distributivity of meet \wedge over join \vee in the lattice ℓ of truth values implies $R^{n[D]} = R^{n[W]}$. This indicates

that the above two ways of defining power of a relation coincide when we work in classical Boolean logic, but the classical definition of power of a relation splits into two nonequivalent versions in quantum logic.

2.6. Orthomodular Lattice-Valued Languages

The notion of language in automata theory has a straightforward orthomodular lattice-valued generalization. Suppose that Σ is an alphabet, that is, a finite nonempty set (of input symbols). We write Σ^* for the set of strings over Σ :

$$\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n.$$

The empty string is usually denoted by ε . An ℓ -valued language over Σ is defined to be an ℓ -valued subset of Σ^* . Thus, the set of ℓ -valued languages over Σ is exactly L^{Σ^*} .

Let $A, B \in L^{\Sigma^*}$ be two ℓ -valued subsets of Σ^* . Then we define the concatenation $A \cdot B$ of A and B as follows: for any $s \in \Sigma^*$,

$$s \in A \cdot B \stackrel{def}{=} (\exists u, v \in \Sigma^*)(s = uv \wedge u \in A \wedge v \in B).$$

This defining equation can be translated to the following formula in the lattice of truth values by employing the truth valuation rules: for every $s \in \Sigma^*$,

$$(A \cdot B)(s) = \bigvee \{A(u) \wedge B(v) : u, v \in \Sigma^* \text{ and } s = uv\}.$$

Similar to the case of defining closure of a relation, the Kleene closure of an ℓ -valued language A over Σ also have two nonequivalent definitions. In the depth-first way, it is defined to be $A^{*[D]} \in L^{\Sigma^*}$, where

$$s \in A^{*[D]} \stackrel{def}{=} (\exists n \geq 0)(\exists s_1, \dots, s_n \in \Sigma^*)(s = s_1 \dots s_n \wedge \bigwedge_{i=1}^n (s_i \in A)),$$

that is,

$$A^{*[D]}(s) = \bigvee \left\{ \bigwedge_{i=1}^n A(s_i) : n \geq 0, s_1, \dots, s_n \in \Sigma^* \text{ and } s = s_1 \dots s_n \right\}$$

for each $s \in \Sigma^*$. In the width-first way, the Kleene closure of A is defined to be

$$A^{*[W]} = \bigcup_{n=0}^{\infty} A^n,$$

where

$$\begin{cases} A^0 = \{\varepsilon\}, \\ A^{n+1} = A^n \cdot A \text{ for all } n \geq 0. \end{cases}$$

It is easy to demonstrate that if the meet \wedge is distributive over the join \vee in ℓ (in other words, ℓ is a Boolean algebra), then we have $A^{*[D]} = A^{*[W]}$.

Let Σ and Γ be two alphabets of input symbols. Then each mapping $h : \Sigma \rightarrow \Gamma^*$ can be uniquely extended to a homomorphism $h : \Sigma^* \rightarrow \Gamma^*$ such that $h(\varepsilon) = \varepsilon$ and $h(xy) = h(x)h(y)$ for all $x, y \in \Sigma^*$. Furthermore, we may define images of ℓ -valued subsets of Σ^* under h and pre-images of ℓ -valued subsets of Γ^* under h . For any $A \in L^{\Sigma^*}$ and $B \in L^{\Gamma^*}$, $h(A) \in L^{\Gamma^*}$ and $h^{-1}(B) \in L^{\Sigma^*}$ are given as follows:

$$t \in h(A) \stackrel{def}{=} (\exists s \in \Sigma^*)(s \in A \wedge h(s) = t)$$

or equivalently

$$h(A)(t) = \bigvee \{A(s) : s \in \Sigma^* \text{ and } h(s) = t\}$$

for each $t \in \Gamma^*$, and

$$s \in h^{-1}(B) \stackrel{def}{=} h(s) \in B$$

or equivalently $h^{-1}(B)(s) = B(h(s))$ for each $s \in \Sigma^*$.

3. Orthomodular Lattice-Valued (Nondeterministic) Finite Automata

The finite automaton is a useful mathematical model of finite state systems, with discrete inputs and outputs, and it is one of the simplest models of computation. There are many finite state systems in computer science and other fields that can be described as finite automata. The theory of finite automata is an essential part of computing theory. Classical automata theory is established in the framework of classical Boolean logic. This section is devoted to a systematic development of a theory of finite automata based on quantum logic.

3.1. Basic Definitions and Examples

For convenience we first recall some basic notions in classical automata theory. Let Σ be a finite input alphabet whose elements are called input symbols or labels. Then a nondeterministic finite automaton (NFA for short) over Σ is a quadruple $\mathfrak{R} = \langle Q, \delta, I, T, \rangle$, in which:

- (i) Q is a finite set whose elements are called states;
- (ii) $I \subseteq Q$, and states in I are said to be initial;
- (iii) $T \subseteq Q$, and states in T are said to be terminal; and
- (iv) $\delta \subseteq Q \times \Sigma \times Q$, and each $(p, \sigma, q) \in \delta$ is called a transition in (or an edge of) \mathfrak{R} and it means that input σ makes state p evolves to q .

Usually, the set of initial states is taken to be a singleton $\{q_0\}$. In this case, \mathfrak{R} is simply written as $\langle Q, E, q_0, T \rangle$.

An NFA is said to be deterministic if I is a singleton, and for any p in Q and σ in Σ , there is exactly one q in Q such that $(p, \sigma, q) \in \delta$. Thus, the transition relation E in a deterministic finite automaton (DFA, for short) may be seen as a mapping from $Q \times \Sigma$ into Q , and it is called the transition function.

A path in \mathfrak{R} is a finite sequence of the form $c = q_0\sigma_1q_1\dots q_{k-1}\sigma_kq_k$ such that $(q_i, \sigma_{i+1}, q_{i+1}) \in \delta$ for each $i < k$. In this case, the sequence $\sigma_1\dots\sigma_k$ is called the label of c . A path $c = q_0\sigma_1q_1\dots q_{k-1}\sigma_kq_k$ is said to be successful if $q_0 \in I$ and $q_k \in T$. The language $L(\mathfrak{R})$ accepted by an automaton \mathfrak{R} is the set of labels of all successful paths in \mathfrak{R} . The definition of accepted language is often restated as follows. For any $P \subseteq Q$ and $\sigma \in \Sigma$, we write

$$\delta_\sigma(P) = \{q \in Q : (p, \sigma, q) \in \delta \text{ for some } p \in P\}.$$

Furthermore, we set $\delta_\varepsilon(P) = P$ and $\delta_{s\sigma}(P) = \delta_\sigma(\delta_s(P, s))$ for any $P \subseteq Q$, $s \in \Sigma^*$ and $\sigma \in \Sigma$. Then $L(\mathfrak{R}) = \{s \in \Sigma^* : \delta_s(I) \cap T \neq \emptyset\}$. For each $A \subseteq \Sigma^*$, A is said to be regular if there is an automaton \mathfrak{R} over Σ such that $A = L(\mathfrak{R})$.

The notion of orthomodular lattice-valued finite automata is a natural generalization of NFAs. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let Σ be a finite alphabet. Then an ℓ -valued (quantum) nondeterministic finite automaton over Σ is a quadruple $\mathfrak{R} = \langle Q, \delta, I, T \rangle$, where:

(i) Q is the same as in an NFA;

(ii) I is an ℓ -valued subset of Q ; that is, a mapping from Q into L . For each $q \in Q$, $I(q)$ indicates the truth value (in the underlying quantum logic) of the proposition that q is an initial state;

(iii) T is also an ℓ -valued subset of Q , and for every $q \in Q$, $T(q)$ expresses the truth value (in our quantum logic) of the proposition that q is terminal; and

(iv) δ is an ℓ -valued subset of $Q \times \Sigma \times Q$, that is, a mapping from $Q \times \Sigma \times Q$ into L , and it is called the ℓ -valued (quantum) transition relation of \mathfrak{R} . Intuitively, δ is an ℓ -valued (ternary) predicate over Q, Σ and Q , and for any $p, q \in Q$ and $\sigma \in \Sigma$, $\delta(p, \sigma, q)$ stands for the truth value (in quantum logic) of the proposition that input σ causes state p to become q .

The propositions of the form “ q is an initial state”, written “ $q \in I$ ”, “ q is a terminal state”, written “ $q \in T$ ”, and “input σ causes state p to become q according to the specification given by δ ”, written “ $p \xrightarrow{\delta, \sigma} q$ ” are assumed to be atomic propositions in our logical language designated for describing ℓ -valued automata \mathfrak{R} . The truth values of the above three propositions are respectively $I(q)$, $T(q)$ and $\delta(p, \sigma, q)$. The set of these atomic propositions is denoted $atom(\mathfrak{R})$. Formally, we have:

$$atom(\mathfrak{R}) = \{q \in I : q \in Q\} \cup \{q \in T : q \in Q\} \cup \{p \xrightarrow{\delta, \sigma} q : p, q \in Q \text{ and } \sigma \in \Sigma\}.$$

The ℓ -valued set I of initial states is often chosen to be a singleton $\{q_0\}$ in an ℓ -valued automaton \mathfrak{R} , and in this case we simply write $\mathfrak{R} = \langle Q, \delta, q_0, T \rangle$.

We write $\mathbf{NFA}(\Sigma, \ell)$ for the (proper) class of all ℓ -valued nondeterministic finite automata over Σ .

There are two nonequivalent ways of generalizing the concept of recognizability in quantum logic: the depth-first one and the width-first one. Before defining recognizability for ℓ -valued automata in the depth-first way, we need to introduce some auxiliary notions and notations. We set

$$T(Q, \Sigma) = (Q\Sigma)^*Q = \bigcup_{n=0}^{\infty} [(Q\Sigma)^n Q],$$

that is, the set of all alternative sequences of states and labels beginning at a state and also ending at a state. For any $c = q_1\sigma_1q_2\dots q_k\sigma_kq_{k+1} \in T(Q, \Sigma)$, the length of c is defined to be k and denoted by $|c|$, q_1 is the beginning of c and denoted by $b(c)$, q_{k+1} is the end of c and denoted by $e(c)$, and sequence $s = \sigma_1\dots\sigma_k$ is called the label of c and denoted by $lb(c)$.

Let $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ be an ℓ -valued automaton over Σ . Then the ℓ -valued (unary) predicate $Path_{\mathfrak{R}}$ on $T(Q, \Sigma)$ is defined as $Path_{\mathfrak{R}} \in L^{T(Q, \Sigma)}$ (the set of all mappings from $T(Q, \Sigma)$ into L):

$$Path_{\mathfrak{R}}(c) \stackrel{def}{=} \bigwedge_{i=1}^k [(q_i, \sigma_i, q_{i+1}) \in \delta]$$

for every $c = q_1\sigma_1q_1\dots q_k\sigma_kq_{k+1} \in T(Q, \Sigma)$. Thus, the truth value of the proposition that $c = q_1\sigma_1q_1\dots q_k\sigma_kq_{k+1}$ is a path in \mathfrak{R} is

$$\lceil Path_{\mathfrak{R}}(c) \rceil = \bigwedge_{i=1}^k \delta(q_i, \sigma_i, q_{i+1}).$$

Note the difference between the symbols \wedge in the above two equations: the former is a logical connective, whereas the latter is an operation on the lattice of truth values.

Now we are ready to define one of the key notions in this section, namely, recognizability for ℓ -valued automata according to the depth-first principle. It will be seen that the defining equation of ℓ -valued recognizability is the same as that in the classical theory of automata. The essential difference between the quantum recognizability and the corresponding classical notion implicitly resides in their truth values.

Definition 3.1. Let $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$. Then the recognizability $rec_{\mathfrak{R}}^{[D]}$ by \mathfrak{R} in the depth-first way is defined to be an ℓ -valued (unary) predicate $rec_{\mathfrak{R}} \in L^{\Sigma^*}$: for every $s \in \Sigma^*$,

$$rec_{\mathfrak{R}}^{[D]}(s) \stackrel{def}{=} (\exists c \in T(Q, \Sigma))(b(c) \in I \wedge e(c) \in T \wedge lb(c) = s \wedge Path_{\mathfrak{R}}(c)).$$

In other words, the truth value of the proposition that s is recognizable by \mathfrak{R} in the depth-first way is given by

$$\lceil rec_{\mathfrak{R}}(s) \rceil^{[D]} = \bigvee \{ I(b(c)) \wedge T(e(c)) \wedge \lceil Path_{\mathfrak{R}}(c) \rceil : c \in T(Q, \Sigma) \text{ and } lb(c) = s \}.$$

We note that $rec_{\mathfrak{R}}^{[D]}$ is defined above as an ℓ -valued unary predicate on Σ^* , so it may also be seen as an ℓ -valued subset of Σ^* , that is, a mapping $rec_{\mathfrak{R}}^{[D]} : \Sigma^* \rightarrow L$ with $rec_{\mathfrak{R}}^{[D]}(s) = \lceil rec_{\mathfrak{R}}^{[D]}(s) \rceil$ for all $s \in \Sigma^*$.

We now turn to introduce ℓ -valued recognizability in the width-first way. Suppose $\mathfrak{R} = \langle Q, \delta, I, T \rangle \in \mathbf{NFA}(\Sigma, \ell)$. For any $P \in L^Q$ and $\sigma \in \Sigma$, the image $\delta_{\sigma}(P) \in L^Q$ of P under δ with respect to σ is defined by

$$q \in \delta_{\sigma}(P) \stackrel{def}{=} (\exists p \in Q)(p \in P \wedge (p, \sigma, q) \in \delta),$$

that is,

$$\delta_{\sigma}(P)(q) = \bigvee_{p \in Q} (P(p) \wedge \delta(p, \sigma, q))$$

for each $q \in Q$. Furthermore, $\delta_s(P)$ for $s \in \Sigma^*$ is defined by induction on the length $|s|$ of s :

$$\begin{cases} \delta_{\varepsilon}(P) = P, \\ \delta_{s\sigma}(P) = \delta_{\sigma}(\delta_s(P)) \quad \text{for any } \sigma \in \Sigma \text{ and } s \in \Sigma^*. \end{cases}$$

Definition 3.2. Let $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$. Then recognizability by \mathfrak{R} in the width-first way is defined to be an ℓ -valued (unary) predicate $rec_{\mathfrak{R}}^{[D]} \in L^{\Sigma^*}$: for each $s \in \Sigma^*$,

$$rec_{\mathfrak{R}}^{[D]}(s) \stackrel{def}{=} (\exists q \in Q)(q \in \delta_s(I) \cap T).$$

The next lemma examines the relationship between the two notions of recognizability defined in depth-first and width-first ways.

Lemma 3.3. Let Σ be a nonempty finite alphabet.

(1) For any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and for any $s \in \Sigma^*$, we have:

$$\models^{\ell} rec_{\mathfrak{R}}^{[D]}(s) \rightarrow rec_{\mathfrak{R}}^{[W]}(s).$$

(2) The following two statements are equivalent:

(2.1) ℓ is a Boolean algebra;

(2.2) for any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and for any $s \in \Sigma^*$, we have:

$$\models^{\ell} rec_{\mathfrak{R}}^{[D]}(s) \leftrightarrow rec_{\mathfrak{R}}^{[W]}(s).$$

(3) For any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and for any $s \in \Sigma^*$, we have:

$$\models^{\ell} \gamma(\text{atom}(\mathfrak{R})) \wedge rec_{\mathfrak{R}}^{[W]}(s) \rightarrow rec_{\mathfrak{R}}^{[D]}(s),$$

and in particular if $\rightarrow = \rightarrow_3$ then

$$\models^{\ell} \gamma(\text{atom}(\mathfrak{R})) \rightarrow (rec_{\mathfrak{R}}^{[D]}(s) \leftrightarrow rec_{\mathfrak{R}}^{[W]}(s)).$$

Proof. (1) From Definitions 3.1 and 3.2 it suffices to verify that for each $c \in T(Q, \Sigma)$ with $lb(c) = s$,

$$I(b(c)) \wedge \lceil \text{Path}_{\mathfrak{R}}(c) \rceil \leq \delta_s(I)(e(c)).$$

We proceed by induction on the length $|s|$ of s . If $s = \varepsilon$, then $b(c) = e(c)$, and the inequality is valid. Suppose that $s = \sigma_1 \dots \sigma_k \sigma_{k+1}$ and $c = q_1 \sigma_1 \dots q_k \sigma_k q_{k+1} \sigma_{k+1} q_{k+2}$. Then the induction hypothesis implies

$$\begin{aligned} I(b(c)) \wedge \lceil \text{Path}_{\mathfrak{R}}(c) \rceil &= I(q_1) \wedge \bigwedge_{i=1}^{k+1} \delta(q_i, \sigma_i, q_{i+1}) \\ &\leq \delta_{\sigma_1 \dots \sigma_k}(I)(q_{k+1}) \wedge \delta(q_{k+1}, \sigma_{k+1}, q_{k+2}) \\ &\leq \delta_{\sigma_{k+1}}(\delta_{\sigma_1 \dots \sigma_k}(I))(q_{k+2}) \\ &= \delta_s(I)(e(c)). \end{aligned}$$

(2) The implication from (2.1) to (2.2) follows directly from (3). So, we merely need to prove that (2.2) implies (2.1). We pick up an element σ from the nonempty set Σ . For any $\lambda, \mu_1, \mu_2 \in L$, we set $\mathfrak{R} = \langle \{q_0, q_1, q_2, q_3, q_4\}, \delta, q_0, \{q_4\} \rangle$, $\delta(q_0, \sigma, q_1) = \mu_1$, $\delta(q_0, \sigma, q_2) = \mu_2$, $\delta(q_1, \sigma, q_3) = \delta(q_2, \sigma, q_3) = 1$ and $\delta(q_3, \sigma, q_4) = \lambda_1$, and it is assumed that δ takes value 0 for all the other arguments. A simple calculation yields $rec_{\mathfrak{R}}^{[D]}(\sigma^3) = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2)$ and $rec_{\mathfrak{R}}^{[W]}(\sigma^3) = \lambda \wedge (\mu_1 \vee \mu_2)$. Then from (2.2) we obtain $\lambda \wedge (\mu_1 \vee \mu_2) = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2)$. Thus, ℓ enjoys distributivity and it is a Boolean algebra.

(3) We first prove that

$$\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \delta_s(I)(q) \leq \bigvee \{ I(b(c)) \wedge \lceil \text{Path}_{\mathfrak{R}}(c) \rceil : c \in T(Q, \Sigma), lb(c) = s \text{ and } e(c) = q \}$$

for each $q \in Q$. This can be done by induction on $|s|$. It is obvious when $s = \varepsilon$. In general, for any $s' \in \Sigma^*$ and $\sigma \in \Sigma$, we obtain:

$$\begin{aligned} \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \delta_{s'\sigma}(I)(q) &= \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \delta_{\sigma}(\delta_{s'}(I))(q) \\ &= \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \bigvee_{p \in Q} (\delta_{s'}(I)(p) \wedge \delta(p, \sigma, q)) \\ &\leq \bigvee_{p \in Q} (\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \delta_{s'}(I)(p) \wedge \delta(p, \sigma, q)) \\ &\leq \bigvee_{p \in Q} (\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \delta(p, \sigma, q) \wedge \bigvee \{ I(b(d)) \wedge \lceil \text{Path}_{\mathfrak{R}}(d) \rceil : \\ &\quad d \in T(Q, \Sigma), lb(d) = s' \text{ and } e(d) = p \}) \\ &\leq \bigvee \{ I(b(d)) \wedge \lceil \text{Path}_{\mathfrak{R}}(d) \rceil \wedge \delta(p, \sigma, q) : d \in T(Q, \Sigma), lb(d) = s' \text{ and } e(d) = p \in Q \} \end{aligned}$$

by using Lemmas 2.6 and 2.7 twice and by using the induction hypothesis. For any $p \in Q$ and $d \in T(Q, \Sigma)$ with $lb(d) = s'$ and $e(d) = p$, we put $c = d\sigma q$. then $c \in T(Q, \Sigma)$, $lb(c) = s'\sigma$, $e(c) = q$ and

$$I(b(d)) \wedge \lceil \text{Path}_{\mathfrak{R}}(d) \rceil \wedge \delta(p, \sigma, q) = I(b(c)) \wedge \lceil \text{Path}_{\mathfrak{R}}(c) \rceil.$$

Therefore, it follows that

$$\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \delta_{s'\sigma}(I)(q) \leq \bigvee \{ I(b(c)) \wedge \lceil \text{Path}_{\mathfrak{R}}(c) \rceil : c \in T(Q, \Sigma), lb(c) = s'\sigma \text{ and } e(c) = q \}.$$

Furthermore, by using Lemmas 2.6 and 2.7 twice again and by using the above conclusion we obtain:

$$\begin{aligned} \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil rec_{\mathfrak{R}}^{[W]}(s) \rceil &= \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \bigvee_{q \in Q} (\delta_s(I)(q) \wedge I(q)) \\ &\leq \bigvee_{q \in Q} (\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \delta_s(I)(q) \wedge I(q)) \\ &\leq \bigvee_{q \in Q} (\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge T(q) \bigvee \{ I(b(c)) \wedge \lceil \text{Path}_{\mathfrak{R}}(c) \rceil : c \in T(Q, \Sigma), lb(c) = s \text{ and } e(c) = q \}) \\ &\leq \bigvee \{ I(b(c)) \wedge \lceil \text{Path}_{\mathfrak{R}}(c) \rceil \wedge T(q) : c \in T(Q, \Sigma), lb(c) = s \text{ and } e(c) = q \in Q \} \\ &= \lceil rec_{\mathfrak{R}}^{[D]}(s) \rceil. \end{aligned}$$

This completes the proof of the first part. The second part follows immediately from the first one and Lemma 2.11. \square

As a straightforward generalization of regular language, we can define regularity for ℓ -valued languages. But ℓ -valued regularity may be defined in two different ways, namely, according to the depth-first principle or the width-first one.

Definition 3.4. The two ℓ -valued (unary) regularity predicates $Reg_{\Sigma}^{[D]}$ and $Reg_{\Sigma}^{[W]}$ on L^{Σ^*} (the set of all ℓ -valued subsets of Σ^*), regularity in the depth-first way and regularity in the width-first way, are defined as $Reg_{\Sigma}^{[D]}, Reg_{\Sigma}^{[W]} \in L^{(L^{\Sigma^*})}$, respectively: for each $A \in L^{\Sigma^*}$,

$$Reg_{\Sigma}^{[D]}(A) \stackrel{def}{=} (\exists \mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell))(A \equiv rec_{\mathfrak{R}}^{[D]}),$$

$$Reg_{\Sigma}^{[W]}(A) \stackrel{def}{=} (\exists \mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell))(A \equiv rec_{\mathfrak{R}}^{[W]}).$$

Thus, the truth value of the proposition that A is regular is

$$\lceil Reg_{\Sigma}^{[D]}(A) \rceil = \bigvee \{ \lceil A \equiv rec_{\mathfrak{R}}^{[D]} \rceil : \mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell) \}.$$

A similar calculation applies to regularity in the width-first way.

It should be noted that the (automaton) variable \mathfrak{R} bounded by the existential quantifier in the right-hand side of the defining formula of $Reg_{\Sigma}^{[D]}$ ranges over the proper class $\mathbf{NFA}(\Sigma, \ell)$. Some readers who are familiar with axiomatic set theory may worry about that this definition will cause a certain set-theoretical difficulty, but we stay well away from anything genuinely problematic. Indeed, for any ℓ -valued automaton $\mathfrak{R} = \langle Q, \delta, I, T \rangle$, there is a bijection $\varsigma : Q \rightarrow |Q|$ (the cardinality of Q) $= \{0, 1, \dots, |Q| - 1\}$ and we can construct a new ℓ -valued automaton

$$\varsigma(\mathfrak{R}) = \langle |Q|, \varsigma(\delta), \varsigma(I), \varsigma(T) \rangle$$

where $\varsigma(\delta)(m, \sigma, n) = \delta(\varsigma^{-1}(m), \sigma, \varsigma^{-1}(n))$ for any $m, n \in |Q|$ and $\sigma \in \Sigma$. It is easy to see that $rec_{\mathfrak{R}}^{[D]} = rec_{\varsigma(\mathfrak{R})}^{[D]}$. Obviously, such a transformation also holds for regularity in the width-first way. Then in Definition 3.4 we may only require that the variable \mathfrak{R} bounded by the existential quantifier ranges over all ℓ -valued automata whose state sets are subsets of ω (the set of all non-negative integers). Note that the class of all ℓ -valued automata with subsets of ω as state sets is really a set, and in fact it is a subset of $(2^{\omega})^3 \times \bigcup_{Q \subseteq \omega} L^{Q \times \Sigma \times Q}$. In most situations, however, the original version of Definition 3.4 is much more convenient and compatible with the corresponding definition in classical automata theory.

Before investigating carefully various properties of ℓ -valued regular languages, we present some interesting examples. The first one indicates that every finite ℓ -valued language is regular in both the depth-first way and the width-first way. It is well-known that a similar conclusion holds in classical automata theory.

Example 3.5. For any $A \in L^{\Sigma^*}$, if A is finite, i.e., $\text{supp}A = \{s \in \Sigma^* : A(s) > 0\}$ is finite, then

$$\models^\ell \text{Reg}_\Sigma^{[D]}(A) \text{ and } \models^\ell \text{Reg}_\Sigma^{[W]}(A).$$

Indeed, suppose that $\text{supp}A = \{\sigma_{i1}\dots\sigma_{im_i} : i = 1, \dots, k\}$. Then we construct an ℓ -valued automaton $\mathfrak{R}_A = (Q_A, I_A, T_A, \delta_A)$ in the following way:

(i) $Q_A = \cup_{i=1}^k \{q_{i0}, q_{i1}, \dots, q_{im_i}\};$

(ii) $I_A = \{q_{10}, q_{20}, \dots, q_{k0}\};$

(iii) $T_A = \{q_{1m_1}, q_{2m_2}, \dots, q_{km_k}\};$

(iv) We define $\delta_A(q_{ij}, \sigma_{i(j+1)}, q_{i(j+1)}) = A(\sigma_{i1}\dots\sigma_{im_i})$ for any $1 \leq i \leq k$ and $0 \leq j < m_i$, and we define $\delta_A(p, \sigma, q) = 0$ for other $(p, \sigma, q) \in Q_A \times \Sigma \times Q_A$. Then it is easy to see that $\text{rec}_{\mathfrak{R}_A}^{[D]} = A$ and $[\text{Reg}_\Sigma^{[D]}(A)] \geq [A \equiv \text{rec}_{\mathfrak{R}_A}^{[D]}] = 1$. Similarly, we have $\text{rec}_{\mathfrak{R}_A}^{[W]} = A$ and $[\text{Reg}_\Sigma^{[W]}(A)] = 1$.

The following example may be seen as an extension of Example 3.5, and it shows that the recognizability of a quantum language is not less than the volume of its finite part.

Example 3.6. For any $A \in L^{\Sigma^*}$, we define:

$$A \downarrow \lambda = \{s \in \Sigma^* : A(s) \not\leq \lambda\} \text{ and } A \uparrow \lambda = \{s \in \Sigma^* : A(s) \not\geq \lambda\}.$$

They are called lower and upper anti- λ -cuts of A , respectively. Then it holds that

(1) $\models^\ell \mu \rightarrow \text{Reg}_\Sigma^{[D]}(A)$ and $\models^\ell \mu \rightarrow \text{Reg}_\Sigma^{[W]}(A)$, where $\mu = \vee\{\lambda^\perp : A \downarrow \lambda \text{ is finite}\};$

(2) $\models^\ell \theta \rightarrow \text{Reg}_\Sigma^{[D]}(A)$ and $\models^\ell \theta \rightarrow \text{Reg}_\Sigma^{[W]}(A)$, where $\theta = \vee\{\lambda : A \uparrow \lambda \text{ is finite}\}.$

Here \rightarrow may be interpreted as any implication operator satisfying the Birkhoff-von Neumann requirement. We only prove (1) for regularity in the depth-first way, and the other conclusions may be proven likewise. For any $\lambda \in L$, if $A \downarrow \lambda$ is finite, then we define $A \downarrow \lambda \in L^{\Sigma^*}$ as follows: for any $s \in \Sigma^*$,

$$(A \downarrow \lambda)(s) = \begin{cases} A(s), & \text{if } A(s) \not\leq \lambda, \\ 0, & \text{if } A(s) \leq \lambda. \end{cases}$$

Clearly, $A \downarrow \lambda$ is finite. Then from Example 3.5 we know that there is an ℓ -valued automata $\mathfrak{R}[\lambda]$ such that $\text{rec}_{\mathfrak{R}[\lambda]}^{[D]} = A \downarrow \lambda$, i.e., $\text{rec}_{\mathfrak{R}[\lambda]}^{[D]} = A(s)$ if $A(s) \not\leq \lambda$ and $\text{rec}_{\mathfrak{R}[\lambda]}^{[D]} = 0$ if $A(s) \leq \lambda$, and

$$\begin{aligned} [\text{Rec}_\Sigma^{[D]}(A)] &\geq [A \equiv \text{rec}_{\mathfrak{R}[\lambda]}^{[D]}] \\ &= \bigwedge \{A(s) \leftrightarrow \text{rec}_{\mathfrak{R}[\lambda]}^{[D]} : A(s) \not\leq \lambda\} \wedge \bigwedge \{A(s) \leftrightarrow 0 : A(s) \leq \lambda\} \\ &= \bigwedge \{A(s) \leftrightarrow 0 : A(s) \leq \lambda\} \geq \lambda^\perp. \end{aligned}$$

The third example gives a simple connection between recognizability in classical automata theory and the ℓ -valued predicates $Reg_{\Sigma}^{[D]}$ and $Reg_{\Sigma}^{[W]}$ introduced above.

Example 3.7. Let $A \subseteq \Sigma^*$ be a regular language (in classical automata theory), $B \in L^{\Sigma^*}$ and $suppB = \{s \in \Sigma^* : B(s) > 0\} \subseteq A$, and let

$$\lambda = \bigvee \{ \bigwedge_{s \in A} (a \leftrightarrow B(s)) : a \in L \}.$$

Then we have:

$$\models^{\ell} \lambda \rightarrow Reg_{\Sigma}^{[D]}(B) \text{ and } \models^{\ell} \lambda \rightarrow Reg_{\Sigma}^{[W]}(B).$$

In particular, if $A \subseteq \Sigma^*$ is regular then for every $a \in L$,

$$\models^{\ell} Reg_{\Sigma}^{[D]}(A[a]) \text{ and } \models^{\ell} Reg_{\Sigma}^{[W]}(A[a]),$$

where $A[a] \in L^{\Sigma^*}$ is given as

$$A[a](s) = \begin{cases} a, & \text{if } s \in A, \\ 0, & \text{otherwise.} \end{cases}$$

This conclusion is not difficult to prove. In fact, since A is regular, there must be an automaton $\mathfrak{R} = \langle Q, \delta, I, T \rangle$ that accepts the language A . Now, for each $a \in L$, we construct an ℓ -valued automaton $\mathfrak{R}_a = \langle Q, I, T, \delta_a \rangle$ such that

$$\delta_a(p, \sigma, q) = \begin{cases} a, & \text{if } (p, \sigma, q) \in \delta, \\ 0, & \text{otherwise.} \end{cases}$$

Then it is easy to know that for all $s \in \Sigma^*$,

$$[rec_{\mathfrak{R}_a}^{[D]}(s)] = \begin{cases} a, & \text{if } s \in A, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$[B \equiv rec_{\mathfrak{R}_a}^{[D]}] = \bigwedge_{s \in A} (a \leftrightarrow B(s)).$$

Therefore, we have

$$[Reg_{\Sigma}^{[D]}(B)] \geq \bigvee \{ [B \equiv rec_{\mathfrak{R}_a}^{[D]}] : a \in L \} = \lambda.$$

A similar argument proves the conclusion for regularity in the depth-first way.

The fourth example demonstrates that the ℓ -valued predicate $Reg_{\Sigma}^{[D]}$ defined above is not trivial, that is, it does not in general degenerate into a two-valued

(Boolean) predicate. We can also construct an example for the same purpose with respect to $Reg_{\Sigma}^{[W]}$.

Example 3.8. We consider a canonical orthomodular lattice. This lattice has a clear interpretation in quantum physics. One pasts together observables of the spin one-half system. Then he will obtain an orthomodular lattice $L(x) \oplus L(\bar{x})$, where $L(x) = \{0, p_-, p_+, 1\}$ corresponds to the outcomes of a measurement of the spin states along the x -axis and $L(\bar{x}) = \{\bar{0} = 1, \bar{p}_-, \bar{p}_+, \bar{1} = 0\}$ is obtained by measuring the spin states along a different spatial direction; and $L(x) \oplus L(\bar{x})$ may be visualized as the ‘‘Chinese lantern’’ (see Figure 2, and for a more detailed description of $L(x) \oplus L(\bar{x})$ see [70]).

In this example, we set $\rightarrow = \rightarrow_3$ (the Sasaki-hook). By a routine calculation we have:

$$p_- \leftrightarrow p_+ = p_- \leftrightarrow \bar{p}_- = p_- \leftrightarrow \bar{p}_+ = 0$$

and $p_- \leftrightarrow 1 = p_-$. Thus, for each $\lambda \in L(x) \oplus L(\bar{x})$, $\lambda \not\leq p_-$ implies $p_- \leftrightarrow \lambda \leq p_-$. Furthermore, let $\Sigma = \{\sigma, \tau\}$ and $A = \{\sigma^n \tau^n : n \in \omega\}$, and for any $t \in L(x) \oplus L(\bar{x})$, let $A_t \in L^{\Sigma^*}$ be given as follows:

$$A_t(s) = \begin{cases} 1, & \text{if } s \in A, \\ t, & \text{otherwise,} \end{cases}$$

Then it holds that

$$\models_{\ell} \mathbf{p}_- \leftrightarrow Reg_{\Sigma}^{[D]}(A_{p_-});$$

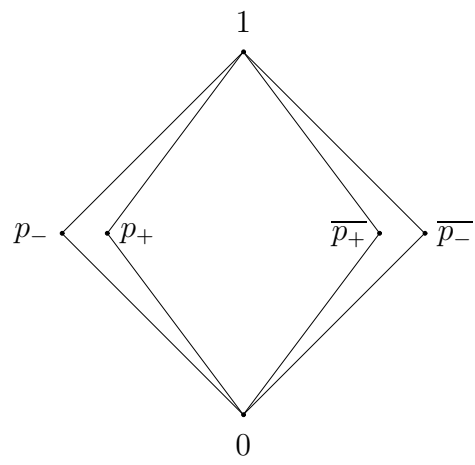
that is, $[Reg_{\Sigma}^{[D]}(A_{p_-})] = p_-$. In fact, we know that Σ^* is regular (see [23], Example II.2.3), and with Example 3.7 it is easy to see that $[Reg_{\Sigma}^{[D]}(A_{p_-})] \geq p_-$. Conversely, for any ℓ -valued automaton $\mathfrak{R} = \langle Q, \delta, I, T \rangle$, if $|Q| = n$ then

$$\begin{aligned} [A_{p_-} \equiv rec_{\mathfrak{R}}^{[D]}] &\leq [A_{p_-}(\sigma^n \tau^n) \leftrightarrow rec_{\mathfrak{R}}^{[D]}(\sigma^n \tau^n)] \wedge \bigwedge_{k,l \in \omega \text{ s.t. } k \neq l} [A_{p_-}(\sigma^k \tau^l) \leftrightarrow rec_{\mathfrak{R}}^{[D]}(\sigma^k \tau^l)] \\ &= rec_{\mathfrak{R}}^{[D]}(\sigma^n \tau^n) \wedge \bigwedge_{k,l \in \omega \text{ s.t. } k \neq l} [p_- \leftrightarrow rec_{\mathfrak{R}}^{[D]}(\sigma^k \tau^l)]. \end{aligned}$$

If $rec_{\mathfrak{R}}^{[D]}(\sigma^n \tau^n) \leq p_-$, then $[A_{p_-} \equiv rec_{\mathfrak{R}}^{[D]}] \leq p_-$. Now, we consider the case of $rec_{\mathfrak{R}}^{[D]}(\sigma^n \tau^n) \not\leq p_-$. For any $c \in T(Q, \Sigma)$, if $b(c) \in I$, $e(c) \in T$ and $lb(c) = \sigma^n \tau^n$, then c must be of the form $c = p_0 \sigma p_1 \dots p_{n-1} \sigma p_n \tau q_1 \dots q_{n-1} \tau q_n$. Since $|Q| = n$, there are i, j such that $i < j \leq n$ and $p_i = p_j$. We put

$$c^+ = p_0 \sigma p_1 \dots p_{j-1} \sigma p_j (= p_i) \sigma p_{i+1} \dots p_{j-1} \sigma p_j \sigma p_{j+1} \dots p_{n-1} \sigma p_n \tau q_1 \dots q_{n-1} \tau q_n.$$

Then $b(c^+) \in I$, $e(c^+) \in T$, $lb(c^+) = \sigma^{n+(j-i)} \tau^n$ and $[Path_{\mathfrak{R}}(c^+)] = [Path_{\mathfrak{R}}(c)]$.



Therefore, it holds that

$$\begin{aligned} rec_{\mathfrak{R}}^{[D]}(\sigma^{n+(j-i)}\tau^n) &\geq \bigvee \{ [Path_{\mathfrak{R}}(c^+)] : b(c) \in I, e(c) \in T \text{ and } lb(c) = \sigma^n\tau^n \} \\ &= \bigvee \{ [Path_{\mathfrak{R}}(c)] : b(c) \in I, e(c) \in T \text{ and } lb(c) = \sigma^n\tau^n \} \\ &= rec_{\mathfrak{R}}(\sigma^n\tau^n), \end{aligned}$$

and $rec_{\mathfrak{R}}^{[D]}(\sigma^{n+(j-i)}\tau^n) \not\leq p_-$. Furthermore, we have:

$$[A_{p_-} \equiv rec_{\mathfrak{R}}^{[D]}] \leq p_- \leftrightarrow rec_{\mathfrak{R}}^{[D]}(\sigma^{n+(j-i)}\tau^n) \leq p_-.$$

So, for all ℓ -valued automata \mathfrak{R} we have $[A_{p_-} \equiv rec_{\mathfrak{R}}^{[D]}] \leq p_-$, and it follows that

$$[Rec_{\Sigma}^{[D]}(A_{p_-})] = \bigvee \{ [A \equiv rec_{\mathfrak{R}}^{[D]}] : \mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell) \} \leq p_-.$$

This together with $[Reg_{\Sigma}^{[D]}(A_{p_-})] \geq p_-$ obtained before leads to $[Reg_{\Sigma}^{[D]}(A_{p_-})] = p_-$.

Similarly, we have $[Reg_{\Sigma}^{[D]}(A_t)] = t$ for $t = p_+, \overline{p_-}$ and $\overline{p_+}$. \square

Motivated by the above example, we propose the open problem: how to describe orthomodular lattices $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ which satisfy that $\{ [Reg_{\Sigma}^{[D]}(A)] : A \in L^{\Sigma^*} \} = L$, i.e., the truth values of recognizability traverse all over L , or more explicitly, for every $\lambda \in L$, there is $A \in L^{\Sigma^*}$ such that $[Reg_{\Sigma}^{[D]}(A)] = \lambda$? We may ask the same question for $Reg_{\Sigma}^{[W]}$. It seems that this is a difficult problem.

The ℓ -valued regularity predicates $Reg_{\Sigma}^{[D]}$ and $Reg_{\Sigma}^{[W]}$ in Definition 3.4 are a direct generalization of the notion of regular language in classical automata theory. In what follows, we will see that the predicate $Reg_{\Sigma}^{[D]}$ and $Reg_{\Sigma}^{[W]}$ do not work well in many cases. Why this happens? Note that $Reg_{\Sigma}^{[D]}$ and $Reg_{\Sigma}^{[W]}$ are merely a simple mimic of the classical concept of regular language, and an essential feature of quantum logic is missing here. In the defining equations of $Reg_{\Sigma}^{[D]}$ and $Reg_{\Sigma}^{[W]}$, the language A to be recognized and the automaton \mathfrak{R} for recognizing A are left completely irrelevant. In the case of classical logic, this does not causes any difficulty in manipulating regular languages. Nevertheless, the thing changes when we work in quantum logic. After an analysis it was found that a suitable link between A and \mathfrak{R} is a commutativity of them. This suggests the following:

Definition 3.9. The ℓ -valued (unary and partial) predicates $CReg_{\Sigma}^{[D]}$ and $CReg_{\Sigma}^{[W]}$ on L^{Σ^*} are called commutative regularity in the depth-first way and width-first way, respectively, and they are defined as $CReg_{\Sigma}^{[D]}, CReg_{\Sigma}^{[W]} \in L^{(L^{\Sigma^*})}$: for any $A \in L^{\Sigma^*}$ with finite $Range(A) = \{A(s) : s \in \Sigma^*\}$,

$$CReg_{\Sigma}^{[D]}(A) \stackrel{def}{=} (\exists \mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)) (\gamma(atom(\mathfrak{R}) \cup r(A)) \wedge (A \equiv rec_{\mathfrak{R}}^{[D]})),$$

$$CReg_{\Sigma}^{[W]}(A) \stackrel{def}{=} (\exists \mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell))(\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge (A \equiv \text{rec}_{\mathfrak{R}}^{[W]})),$$

where $r(A) = \{\mathbf{a} : a \in \text{Range}(A)\}$, and \mathbf{a} is the nullary predicate corresponding to element a in L .

The exposition concerning the automata variable \mathfrak{R} in the defining equations of $Reg_{\Sigma}^{[D]}$ and $Reg_{\Sigma}^{[W]}$ in Definition 3.4 also applies to $CReg_{\Sigma}^{[D]}$ and $CReg_{\Sigma}^{[W]}$ in the above definition.

It is obvious that the notion of commutative regularity is stronger than (non-commutative) regularity. In other words, we have for any $A \in L^{\Sigma^*}$,

$$\models^{\ell} CReg_{\Sigma}^{[D]}(A) \rightarrow Reg_{\Sigma}^{[D]}(A) \text{ and } \models^{\ell} CReg_{\Sigma}^{[W]}(A) \rightarrow Reg_{\Sigma}^{[W]}(A).$$

On the other hand, if ℓ is a Boolean algebra; that is, the underlying logic is the classical Boolean logic, then these two notions are equivalent; or formally, for all $A \in L^{\Sigma^*}$, it holds that

$$\models^{\ell} CReg_{\Sigma}^{[D]}(A) \leftrightarrow Reg_{\Sigma}^{[D]}(A) \text{ and } \models^{\ell} CReg_{\Sigma}^{[W]}(A) \leftrightarrow Reg_{\Sigma}^{[W]}(A).$$

This is just why both the predicate $Reg_{\Sigma}^{[D]}$ and $Reg_{\Sigma}^{[W]}$ work very well in classical automata theory but not in the theory of automata based on quantum logic.

As a direct corollary of Lemma 3.3 we see that $CReg_{\Sigma}^{[D]}$ and $CReg_{\Sigma}^{[W]}$ are equivalent if we use the Sasaki implication \rightarrow_3 .

Corollary 3.10. If $\rightarrow = \rightarrow_3$ then for each $A \in L^{\Sigma^*}$, we have:

$$\models^{\ell} CReg_{\Sigma}^{[D]}(A) \leftrightarrow CReg_{\Sigma}^{[W]}(A).$$

Proof. For any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$, using Lemmas 2.12(3) and 3.4 we obtain:

$$\begin{aligned} [\gamma(\text{atom}(\mathfrak{R}) \cup r(A))] \wedge [A \equiv \text{rec}_{\mathfrak{R}}^{[D]}] &\leq [\gamma(\text{atom}(\mathfrak{R}) \cup r(A))] \wedge [A \equiv \text{rec}_{\mathfrak{R}}^{[D]}] \wedge [\text{rec}_{\mathfrak{R}}^{[D]} \equiv \text{rec}_{\mathfrak{R}}^{[W]}] \\ &\leq [\gamma(\text{atom}(\mathfrak{R}) \cup r(A))] \wedge [A \equiv \text{rec}_{\mathfrak{R}}^{[W]}] \\ &\leq [CReg_{\Sigma}^{[W]}(A)]. \end{aligned}$$

Thus, $[CReg_{\Sigma}^{[D]}(A)] \leq [CReg_{\Sigma}^{[W]}(A)]$. Conversely, we also have $[CReg_{\Sigma}^{[W]}(A)] \leq [CReg_{\Sigma}^{[D]}(A)]$. \square

3.2. Orthomodular Lattice-Valued Deterministic Finite Automata

The notion of nondeterminism plays a central role in the theory of computation. The nondeterministic mechanism enables a device to change its states in a way that

is only partially determined by the current state and input symbol. Obviously, the concept of ℓ -valued automaton introduced in the last section is a generalization of nondeterministic finite automaton. In classical theory of automata, each nondeterministic finite automaton is equivalent to a deterministic one; more exactly, there exists a deterministic finite automaton which accepts the same language as the originally given nondeterministic one does. The aim of this section is just to see whether this result is still valid in the framework of quantum logic. To this end, we first introduce the concept of deterministic ℓ -valued automaton.

Let $\mathfrak{R} = \langle Q, \delta, I, T \rangle \in \mathbf{NFA}(\Sigma, \ell)$ be an ℓ -valued automata over Σ . If

(i) there is a unique q_0 in Q with $I(q_0) > 0$; and

(ii) for all q in Q and σ in Σ , there is a unique state p in Q such that $\delta(q, \sigma, p) > 0$, then M is called an ℓ -valued (quantum) deterministic finite automaton (ℓ -valued DFA for short). The ℓ -valued transition relation δ in an ℓ -valued DFA may be equivalently represented by a mapping from $Q \times \Sigma$ into $Q \times (L - \{0\})$. For any q in Q and σ in Σ , if p is the unique element in Q with $\delta(q, \sigma, p) > 0$, then $\delta(q, \sigma) = (p, \delta(q, \sigma, p)) \in Q \times (L - \{0\})$.

The class of ℓ -valued DFAs over Σ is denoted $\mathbf{DFA}(\Sigma, \ell)$.

Suppose that \mathfrak{R} is an ℓ -valued DFA, $\delta(q_0, \sigma_1) = (q_1, \lambda_1)$ and $\delta(q_i, \sigma_{i+1}) = (q_{i+1}, \lambda_{i+1})$ for all $i = 1, 2, \dots, n-1$. Then it is easy to see that

$$[rec_{\mathfrak{R}}^{[D]}(\sigma_1 \dots \sigma_n)] = [rec_{\mathfrak{R}}^{[W]}(\sigma_1 \dots \sigma_n)] = I(q_0) \wedge T(q_n) \wedge \bigwedge_{i=1}^n \lambda_i.$$

Consequently, for any $\mathfrak{R} \in \mathbf{DFA}(\Sigma, \ell)$, it holds that $rec_{\mathfrak{R}}^{[D]} = rec_{\mathfrak{R}}^{[W]}$. Thus, we shall drop the superscripts and simply write $rec_{\mathfrak{R}}$ for $rec_{\mathfrak{R}}^{[D]}$ (and $rec_{\mathfrak{R}}^{[W]}$) for an ℓ -valued deterministic finite automaton \mathfrak{R} .

Throughout this section, we always suppose that the lattice ℓ of truth values is finite. The reason is that otherwise the set L^Q of states in the ℓ -valued power set construction $\ell^{\mathfrak{R}}$ of \mathfrak{R} , defined below, will be an infinite set, and the assumption that the set of states in an ℓ -valued automaton is finite will be violated.

The proof of the equivalence between classical deterministic finite and nondeterministic finite automata is carried out by building the subset construction of a nondeterministic finite automaton that is deterministic and can simulate the given nondeterministic one. The subset construction can be naturally extended into the case of ℓ -valued automata.

Let $\mathfrak{R} = \langle Q, \delta, I, T \rangle \in \mathbf{NFA}(\Sigma, \ell)$ be an ℓ -valued nondeterministic finite automaton over Σ . We define the ℓ -valued subset construction of \mathfrak{R} to be ℓ -valued automaton $\ell^{\mathfrak{R}} = \langle L^Q, \bar{\delta}, I_1, \bar{T} \rangle$ over Σ , where:

- (i) L^Q is the set of all ℓ -valued subsets of Q , that is, mappings from Q into L ;
- (ii) I_1 is an ℓ -valued point with height 1, that is, $I_1 \in L^{(L^Q)}$ and for all $X \in L^Q$,

$$I_1(X) = \begin{cases} 1, & \text{if } X = I, \\ 0, & \text{otherwise;} \end{cases}$$

(iii) $\bar{T} \in L(L^Q)$, that is, \bar{T} is an ℓ -valued subset of L^Q , and for any $X \in L^Q$,

$$\bar{T}(X) = \bigvee_{q \in Q} [X(q) \wedge T(q)];$$

(iv) $\bar{\delta}$ is a mapping from $L^Q \times \Sigma$ into L^Q , and for each $X \in L^Q$, we have $\bar{\delta}(X, \sigma) \in L^Q$ and for every $q \in Q$,

$$\bar{\delta}(X, \sigma)(q) = \bigvee_{p \in Q} [X(p) \wedge \delta(p, \sigma, q)].$$

Since L is assumed to be finite, L^Q is finite too. Thus, it is easy to see that $\ell^{\mathfrak{R}}$ is an ℓ -valued DFA. Moreover, both the set of the initial states and the transition relation are two-valued, namely, their truth values are either 0 or 1, and only the set of terminal states carries ℓ -valued information.

The following theorem compares the abilities of an ℓ -valued automaton and its subset construction according to the ℓ -valued languages recognized by them in the depth-first way.

Theorem 3.11. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be a finite orthomodular lattice, and let \rightarrow be an implication operator satisfying the Birkhoff-von Neumann requirement.

(1) For any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and $s \in \Sigma^*$, we have:

$$\models^{\ell} \text{rec}_{\mathfrak{R}}^{[D]}(s) \rightarrow \text{rec}_{\ell^{\mathfrak{R}}}(s).$$

(2) For any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and $s \in \Sigma^*$, it holds that

$$\models^{\ell} \gamma(\text{atom}(\mathfrak{R})) \wedge \text{rec}_{\ell^{\mathfrak{R}}}(s) \rightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s),$$

and in particular if $\rightarrow = \rightarrow_3$, then

$$\models^{\ell} \gamma(\text{atom}(\mathfrak{R})) \rightarrow (\text{rec}_{\ell^{\mathfrak{R}}}(s) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s)).$$

(3) The following two statements are equivalent to each other:

(3.1) ℓ is a Boolean algebra.

(3.2) For any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and $s \in \Sigma^*$,

$$\models^{\ell} \text{rec}_{\mathfrak{R}}^{[D]}(s) \longleftrightarrow \text{rec}_{\ell^{\mathfrak{R}}}(s).$$

Proof. The proof of (1) is easy, and we omit it here.

(2) Suppose that $\mathfrak{R} = \langle Q, \delta, I, T \rangle \in \mathbf{NFA}(\Sigma, \ell)$, and $\ell^{\mathfrak{R}} = \langle L^Q, \bar{\delta}, I_1, \bar{T} \rangle$ is the ℓ -valued subset construction of \mathfrak{R} . Our aim is to demonstrate that

$$[\gamma(\text{atom}(\mathfrak{R}))] \wedge [\text{rec}_{\ell^{\mathfrak{R}}}(s)] \leq [\text{rec}_{\mathfrak{R}}^{[D]}(s)]$$

for all $s \in \Sigma^*$. To this end, we first prove the following

$$\text{Claim : } [\gamma(\text{atom}(\mathfrak{R}))] \wedge \bar{\delta}(I, \sigma_1 \dots \sigma_n)(q_n)$$

$$\leq \bigvee \{ I(q_0) \wedge \bigwedge_{i=0}^{n-1} \delta(q_i, \sigma_{i+1}, q_{i+1}) : q_0, q_1, \dots, q_{n-1} \in Q \}$$

for any $\sigma_1, \dots, \sigma_n \in \Sigma$ and $q \in Q$. We proceed by induction on n . For $n = 0$, it is clear. The definition of $\bar{\delta}$ yields

$$\begin{aligned} \bar{\delta}(I, \sigma_1 \dots \sigma_n)(q_n) &= \bar{\delta}(\bar{\delta}(I, \sigma_1 \dots \sigma_{n-1}), \sigma_n)(q_n) \\ &= \bigvee_{q_{n-1} \in Q} [\bar{\delta}(I, \sigma_1 \dots \sigma_{n-1})(q_{n-1}) \wedge \delta(q_{n-1}, \sigma_n, q_n)]. \end{aligned}$$

We write $[\text{atom}(\mathfrak{R})] = \{[\varphi] : \varphi \in \text{atom}(\mathfrak{R})\}$. Then it holds that $[\gamma(\text{atom}(\mathfrak{R}))] = \gamma([\text{atom}(\mathfrak{R})])$. Note that the symbol γ in the left-hand side applies to a set of logical formulas, whereas the one in the right-hand side applies to a subset of L . Furthermore, it is easy to see that $\delta(q_{n-1}, \sigma_n, q_n)$, $\bar{\delta}(I, \sigma_1 \dots \sigma_{n-1})(q_{n-1})$ and $[\gamma(\text{atom}(\mathfrak{R}))]$ are all in $[[\text{atom}(\mathfrak{R})]]$ (the subalgebra of ℓ generated by $[\text{atom}(\mathfrak{R})]$). Thus, with Lemmas 2.6 and 2.7 and the induction hypothesis we obtain:

$$\begin{aligned} [\gamma(\text{atom}(\mathfrak{R}))] \wedge \bar{\delta}(I, \sigma_1 \dots \sigma_n)(q_n) &= [\gamma(\text{atom}(\mathfrak{R}))] \wedge [\gamma(\text{atom}(\mathfrak{R}))] \\ &\quad \wedge \bigvee_{q_{n-1} \in Q} [\bar{\delta}(I, \sigma_1 \dots \sigma_{n-1})(q_{n-1}) \wedge \delta(q_{n-1}, \sigma_n, q_n)] \\ &\leq \bigvee_{q_{n-1} \in Q} [[\gamma(\text{atom}(\mathfrak{R}))] \wedge (\bigvee \{ I(q_0) \wedge \bigwedge_{i=0}^{n-2} \delta(q_i, \sigma_{i+1}, q_{i+1}) : \\ &\quad q_0, q_1, \dots, q_{n-2} \in Q \}) \wedge \delta(q_{n-1}, \sigma_n, q_n)]. \end{aligned}$$

Using Lemmas 2.6 and 2.7 again, we complete the proof of the above claim.

Now with this claim, we can use Lemmas 2.6 and 2.7 twice and derive that

$$\begin{aligned} [\gamma(\text{atom}(\mathfrak{R}))] \wedge [\text{rec}_{\ell\mathfrak{R}}(\sigma_1 \dots \sigma_n)] &= [\gamma(\text{atom}(\mathfrak{R}))] \wedge \bar{T}(\bar{\delta}(I, \sigma_1 \dots \sigma_n)) \\ &= [\gamma(\text{atom}(\mathfrak{R}))] \wedge \bigvee_{q_n \in Q} [\bar{\delta}(I, \sigma_1 \dots \sigma_n)(q_n) \wedge T(q_n)] \\ &\leq \bigvee_{q_n \in Q} [[\gamma(\text{atom}(\mathfrak{R}))] \wedge \bar{\delta}(I, \sigma_1 \dots \sigma_n)(q_n) \wedge T(q_n)] \\ &\leq \bigvee_{q_n \in Q} [[\gamma(\text{atom}(\mathfrak{R}))] \wedge (\bigvee \{ I(q_0) \wedge \bigwedge_{i=0}^{n-1} \delta(q_i, \sigma_{i+1}, q_{i+1}) : q_0, q_1, \dots, q_{n-1} \in Q \}) \wedge T(q_n)] \\ &\leq \bigvee \{ I(q_0) \wedge \bigwedge_{i=0}^{n-1} \delta(q_i, \sigma_{i+1}, q_{i+1}) \wedge T(q_n) : q_0, q_1, \dots, q_{n-1} \in Q \} \\ &= [\text{rec}_{\mathfrak{R}}^{[D]}(\sigma_1 \dots \sigma_n)]. \end{aligned}$$

For the case of $\rightarrow = \rightarrow_3$, what we want to prove is

$$\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \leq \lceil \text{rec}_{\ell\mathfrak{R}}(s) \rceil \rightarrow_3 \lceil \text{rec}_{\mathfrak{R}}^{[D]}(s) \rceil.$$

With the above conclusion and Lemma 2.11, it suffices to show that $\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil C \lceil \text{rec}_{\ell\mathfrak{R}}^{[D]}(s) \rceil$. We observe that

$$\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil = \bigvee \left\{ \bigwedge_{\varphi \in \lceil \text{atom}(\mathfrak{R}) \rceil} \varphi^{f(\varphi)} : f \in \{0, 1\}^{\lceil \text{atom}(\mathfrak{R}) \rceil} \right\}.$$

Then Lemma 2.2 tells us that we only need to prove

$$\bigwedge_{\varphi \in \lceil \text{atom}(\mathfrak{R}) \rceil} \varphi^{f(\varphi)} C \lceil \text{rec}_{\ell\mathfrak{R}}^{[D]}(s) \rceil$$

for all $f \in \{0, 1\}^{\lceil \text{atom}(\mathfrak{R}) \rceil}$. For every $\psi \in \lceil \text{atom}(\mathfrak{R}) \rceil$, note that

$$\bigwedge_{\varphi \in \lceil \text{atom}(\mathfrak{R}) \rceil} \varphi^{f(\varphi)} \leq \psi^{f(\psi)}.$$

Then we have

$$\bigwedge_{\varphi \in \lceil \text{atom}(\mathfrak{R}) \rceil} \varphi^{f(\varphi)} C \psi^{f(\psi)},$$

and furthermore it follows that

$$\bigwedge_{\varphi \in \lceil \text{atom}(\mathfrak{R}) \rceil} \varphi^{f(\varphi)} C \psi$$

from Lemmas 2.1(3) and (4). Since $\lceil \text{rec}_{\ell\mathfrak{R}}^{[D]}(s) \rceil$ is calculated from some elements in $\lceil \text{atom}(\mathfrak{R}) \rceil$ by applying a finite number of meets or unions, we complete the proof with Lemma 2.2.

(3) Note that $\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil = 1$ is always valid when ℓ is a Boolean algebra. Thus, it is proved that (3.1) implies (3.2). We now turn to show that (3.2) implies (3.1). It suffices to show that the meet \wedge is distributive over the union \vee , that is, $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ for all $a, b, c \in L$. Let $a, b, c \in L$. We construct an ℓ -valued automaton $\mathfrak{R} = \langle \{u, v, w\}, \delta, \{u, v\}, \{w\} \rangle$ over Σ which has at least one element σ , where $\delta(u, \sigma, u) = a$, $\delta(u, \sigma, w) = c$, $\delta(v, \sigma, u) = b$, and δ takes the value 0 for other cases. It may be visualized by Figure 3.

In the automaton \mathfrak{R} we have:

$$\begin{aligned} \lceil \text{rec}_{\mathfrak{R}}^{[D]}(\sigma\sigma) \rceil &= \bigvee \{ I(q_0) \wedge T(q_2) \wedge \delta(q_0, \sigma, q_1) \wedge \delta(q_1, \sigma, q_2) : q_0, q_1, q_2 \in Q \} \\ &= \bigvee \{ \delta(u, \sigma, q_1) \wedge \delta(q_1, \sigma, w) : q_1 \in Q \} \vee \bigvee \{ \delta(v, \sigma, q_1) \wedge \delta(q_1, \sigma, u) : q_1 \in Q \} \\ &= [\delta(u, \sigma, u) \wedge \delta(u, \sigma, w)] \vee [\delta(v, \sigma, u) \wedge \delta(u, \sigma, w)] \\ &= (a \wedge c) \vee (b \wedge c). \end{aligned}$$

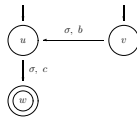


Figure 3: Automaton a

Consider the ℓ -valued subset construction $\ell^{\mathfrak{R}}$ of \mathfrak{R} . Then

$$\begin{aligned}\bar{\delta}(I, \sigma)(u) &= \bigvee_{q \in Q} [I(q) \wedge \delta(q, \sigma, u)] \\ &= \delta(u, \sigma, u) \vee \delta(v, \sigma, u) = a \vee b.\end{aligned}$$

Similarly, we obtain $\bar{\delta}(I, \sigma)(v) = 0$ and $\bar{\delta}(I, \sigma)(w) = c$. It follows that for any $q \in Q$,

$$\begin{aligned}\bar{\delta}(I, \sigma\sigma)(q) &= \bar{\delta}(\bar{\delta}(I, \sigma), \sigma)(q) \\ &= \bigvee_{q' \in Q} [\bar{\delta}(I, \sigma)(q') \wedge \delta(q', \sigma, q)] \\ &= [\bar{\delta}(I, \sigma)(u) \wedge \delta(u, \sigma, q)] \vee [\bar{\delta}(I, \sigma)(w) \wedge \delta(w, \sigma, q)] \\ &= (a \vee b) \wedge \delta(u, \sigma, q).\end{aligned}$$

Thus, it follows that $\bar{\delta}(I, \sigma\sigma)(u) = (a \vee b) \wedge a = a$, $\bar{\delta}(I, \sigma\sigma)(v) = 0$ and $\bar{\delta}(I, \sigma\sigma)(w) = (a \vee b) \wedge c$. Therefore, we obtain:

$$\begin{aligned}[\text{rec}_{\ell^{\mathfrak{R}}}(\sigma\sigma)] &= \bar{T}(\bar{\delta}(I, \sigma\sigma)) \\ &= \bigvee_{q \in Q} [\bar{\delta}(I, \sigma\sigma)(q) \wedge T(q)] \\ &= \bar{\delta}(I, \sigma\sigma)(w) = (a \vee b) \wedge c.\end{aligned}$$

Finally, from the assumption (3.2) we assert that

$$(a \wedge c) \vee (b \wedge c) = [\text{rec}_{\mathfrak{R}}^{[D]}(\sigma\sigma)] = [\text{rec}_{\ell^{\mathfrak{R}}}(\sigma\sigma)] = (a \vee b) \wedge c. \quad \square$$

Many results in this Chapter appear in the same scheme as the above theorem. So, we here give a detailed explanation of this theorem. The above theorem points out that the ability of an ℓ -valued nondeterministic automaton for recognizing language according to the depth-first principle is always weaker than that of its subset construction. On the other hand, in order to warrant that an ℓ -valued automaton \mathfrak{R} and its subset construction have the same ability of accepting language, the condition $\gamma(\text{atom}(\mathfrak{R}))$ has to be imposed. The intuitive meaning of this condition is that (the truth values of) any two atomic propositions describing \mathfrak{R} should commute. (See also the physical interpretation of commutativity presented in the concluding section.) The third part of Theorem 3.11 indicates that the equivalence between a nondeterministic finite automaton and its subset construction is universally valid if and only if the underlying logic degenerates to the classical Boolean logic. In other words, if the meta-logic that we use in our reasoning does not enjoy distributivity, then such a meta-logic is not strong enough to guarantee the universal validity of equivalence between a nondeterministic finite automaton and its subset construction, and we can always find a nondeterministic finite automaton such that the

equivalence between it and its subset construction is not derivable with the mere inference power provided by such a meta-logic. However, the next theorem indicates that any ℓ -valued nondeterministic finite automaton can be simulated by its power set construction whenever the width-first principle is employed.

Theorem 3.12. For each $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and for each $s \in \Sigma^*$, it holds that

$$\models^\ell \text{rec}_{\mathfrak{R}}^{[W]}(s) \leftrightarrow \text{rec}_{\ell\mathfrak{R}}(s).$$

Proof. Immediate from Definition 3.2 and the definition of $\ell\mathfrak{R}$. \square

We can give a simpler proof of Theorem 3.11 by using the above theorem and Lemma 3.3.

Comparing the above two theorems, we see an interesting phenomenon: recognizability in classical automata theory is naturally split into two nonequivalent notions in quantum logic so that the simulation of nondeterministic finite automata by deterministic ones is valid for one of them but not for the other.

In Section 3.1, we introduced four notions of ℓ -valued regularity. They are all given with respect to nondeterministic ℓ -valued automata. Now we propose a restricted version of them based on the smaller class of deterministic ℓ -valued finite automata. Note that here we do not need to distinguish the depth-first and width-first ways.

Definition 3.13. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice. Then the ℓ -valued (unary) predicates $DReg_\Sigma$ and (unary and partial) predicate $CDReg_\Sigma$ on L^{Σ^*} are called deterministic regularity and commutative deterministic regularity, respectively, and they are defined as $DReg_\Sigma, CDReg_\Sigma \in L^{(L^{\Sigma^*})}$: for any $A \in L^{\Sigma^*}$,

$$DReg_\Sigma(A) \stackrel{def}{=} (\exists \mathfrak{R} \in \mathbf{DA}(\Sigma, \ell))(A \equiv \text{rec}_{\mathfrak{R}}),$$

and for any $A \in L^{\Sigma^*}$ with finite $\text{Range}(A) = \{A(s) : s \in \Sigma^*\}$,

$$CDReg_\Sigma(A) \stackrel{def}{=} (\exists \mathfrak{R} \in \mathbf{DA}(\Sigma, \ell))(\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge (A \equiv \text{rec}_{\mathfrak{R}})),$$

where $r(A) = \{\mathbf{a} : a \in \text{Range}(A)\}$, and \mathbf{a} is the nullary predicate associated with element a in L .

It is similar to the relation between $Reg_\Sigma^{[D]}$ and $CReg_\Sigma^{[D]}$ (as well as that between $Reg_\Sigma^{[W]}$ and $CReg_\Sigma^{[W]}$) that $CDReg_\Sigma$ is stronger than $DReg_\Sigma$. In other words, it holds that for any $A \in L^{\Sigma^*}$,

$$\models^\ell CDReg_\Sigma(A) \rightarrow DReg_\Sigma(A).$$

The following corollary shows that a certain commutativity condition guarantees that these two predicates are equivalent. Furthermore, if ℓ is a Boolean algebra, then the six notions $Reg_{\Sigma}^{[D]}$, $Reg_{\Sigma}^{[W]}$, $CReg_{\Sigma}^{[D]}$, $CReg_{\Sigma}^{[W]}$, $DReg_{\Sigma}$ and $CDReg_{\Sigma}$ all coincide.

Corollary 3.14. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be a finite orthomodular lattice. Then for any $A \in L^{\Sigma^*}$, we have:

$$\models^{\ell} Reg_{\Sigma}^{[W]}(A) \leftrightarrow DReg_{\Sigma}(A) \text{ and } \models^{\ell} CReg_{\Sigma}^{[W]}(A) \leftrightarrow CDReg_{\Sigma}(A),$$

and if $\rightarrow = \rightarrow_3$, then

$$\models^{\ell} CReg_{\Sigma}^{[D]}(A) \leftrightarrow CDReg_{\Sigma}(A).$$

Furthermore, if ℓ is a Boolean algebra, then for any $A \in L^{\Sigma^*}$,

$$\models^{\ell} Reg_{\Sigma}^{[D]}(A) \leftrightarrow DReg_{\Sigma}(A).$$

Proof. The first part is immediate from Theorem 3.12. The second part comes from Corollary 3.10 and the first part. The third part is a simple corollary of the second one. \square

3.3. Orthomodular Lattice-Valued Finite Automata with ε -Moves

Finite automata with ε -moves are nondeterministic finite automata in which transitions on the empty input ε are included, and they have the same power for accepting languages. In the classical theory of automata, automata with ε -moves are very convenient tools in building complex automata from simple ones and in proving the closure properties of regular languages. The aim of this section is to introduce an orthomodular lattice-valued extension of automaton with ε -moves.

Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice. Then an ℓ -valued finite automaton with ε -moves over Σ is a quadruple $\mathfrak{R} = \langle Q, \delta, I, T \rangle$ in which all components are the same as in an ℓ -valued nondeterministic finite automaton (without ε -moves), but the domain of the quantum transition relation δ is changed to $Q \times (\Sigma \cup \{\varepsilon\}) \times Q$, that is, δ is a mapping from $Q \times (\Sigma \cup \{\varepsilon\}) \times Q$ into L , where ε stands for the empty string of input symbols. Thus, in an ℓ -valued finite automaton with ε -moves, transitions of the form “ $p \xrightarrow{\delta, \varepsilon} q$ ” are allowed. So, $atom(\mathfrak{R})$ contains the atomic propositions “ $p \xrightarrow{\delta, \varepsilon} q$ ”, and their truth values are given as $\delta(p, \varepsilon, q)$ for all $p, q \in Q$.

Now let $\mathfrak{R} = \langle Q, \delta, I, T \rangle$ be an ℓ -valued finite automaton with ε -moves. We put

$$T_{\varepsilon}(Q, \Sigma) = (Q(\Sigma \cup \{\varepsilon\}))^* Q = \bigcup_{n=0}^{\infty} [(Q(\Sigma \cup \{\varepsilon\}))^n Q].$$

The difference between $T(Q, \Sigma)$ and $T_\varepsilon(Q, \Sigma)$ is that in the latter the empty string may be used as labels. For any $c = q_0\sigma_1q_1\dots q_{k-1}\sigma_kq_k \in T_\varepsilon(Q, \Sigma)$, $lb_\varepsilon(c)$ is defined to be the sequence $\sigma_1\dots\sigma_k$ with all occurrences of ε deleted. Note that it is possible that the length of $lb_\varepsilon(c)$ is strictly smaller than k . Then the recognizability $rec_{\mathfrak{R}}^{[D]}$ in the depth-first way is also defined as an ℓ -valued unary predicate over Σ^* , and it is given by

$$rec_{\mathfrak{R}}^{[D]}(s) \stackrel{def}{=} (\exists c \in T_\varepsilon(Q, \Sigma))(b(c) \in I \wedge e(c) \in T \wedge lb_\varepsilon(c) = s \wedge Path_{\mathfrak{R}}(c))$$

for all $s \in \Sigma^*$, where $Path_{\mathfrak{R}}$ is defined in the same way as in an ℓ -valued automaton without ε -moves. The defining equation of $rec_{\mathfrak{R}}^{[D]}$ may be rewritten in terms of truth valued as follows:

$$[rec_{\mathfrak{R}}^{[D]}(s)] = \bigvee \{I(b(c)) \wedge T(e(c)) \wedge [Path_{\mathfrak{R}}(c)] : c \in T_\varepsilon(Q, \Sigma) \text{ and } lb_\varepsilon(c) = s\},$$

where

$$[Path_{\mathfrak{R}}(c)] = \bigwedge_{i=0}^{k-1} \delta(q_i, \sigma_{i+1}, q_{i+1})$$

if $c = q_0\sigma_1q_1\dots q_{k-1}\sigma_kq_k$.

To introduce recognizability $rec_{\mathfrak{R}}^{[W]} \in L^{\Sigma^*}$ in the width-first way, we define $s^{-\varepsilon} \in \Sigma^*$ to be the sequence of input symbols obtained from s by deleting all occurrences of ε for each $s \in (\Sigma \cup \{\varepsilon\})^*$. Then for each $s \in \Sigma^*$,

$$rec_{\mathfrak{R}}^{[W]}(s) \stackrel{def}{=} (\exists t \in (\Sigma \cup \{\varepsilon\})^*, q \in Q)(t^{-\varepsilon} = s \wedge q \in \delta_t(I) \wedge q \in T),$$

where δ_t is defined in the same way as in an ℓ -valued automaton without ε (that is, ε -moves are treated as usual moves with input symbols).

For any ℓ -valued finite automaton $\mathfrak{R} = \langle Q, \delta, I, T \rangle$ with ε -moves, its ε -reduction is defined to be the ℓ -valued finite automaton $\mathfrak{R}^{-\varepsilon} = \langle Q, \delta', I, T' \rangle$ (without ε -moves) in which

(i) for any $q \in Q$,

$$q \in T' \stackrel{def}{=} (q \in T) \vee (q \in I \wedge (\exists p \in Q, m \geq 0)(p \in T \wedge \delta(q, \varepsilon^m, p))),$$

that is,

$$T'(q) = T(q) \vee [I(q) \wedge \bigvee_{p \in Q, m \geq 0} (T(p) \wedge \delta(q, \varepsilon^m, p))];$$

(ii) for any $p, q \in Q$ and $\sigma \in \Sigma$,

$$\delta'(p, \sigma, q) \stackrel{def}{=} (\exists m, n \geq 0) \delta(p, \varepsilon^m \sigma \varepsilon^n, q),$$

that is,

$$\delta'(p, \sigma, q) = \bigvee_{m, n \geq 0} \delta(p, \varepsilon^m \sigma \varepsilon^n, q),$$

where for all $k \geq 1$, $q_0, q_k \in Q$ and $\sigma_1, \dots, \sigma_k \in \Sigma$,

$$\delta(q_0, \sigma_1 \dots \sigma_k, q_k) \stackrel{def}{=} (\exists q_1, \dots, q_{k-1} \in Q) (\delta(q_0, \sigma_1, q_1) \wedge \delta(q_1, \sigma_2, q_2) \wedge \dots \wedge \delta(q_{k-1}, \sigma_k, q_k)),$$

or equivalently,

$$\delta(q_0, \sigma_1 \dots \sigma_k, q_k) = \bigvee \{ (\delta(q_0, \sigma_1, q_1) \wedge \delta(q_1, \sigma_2, q_2) \wedge \dots \wedge \delta(q_{k-1}, \sigma_k, q_k)) : q_1, \dots, q_{k-1} \in Q \}.$$

The following theorem gives a clear relation between the language accepted by an ℓ -valued automaton with ε -moves and that accepted by its ε -reduction. In general, the ε -reduction of an automaton with ε -moves has a stronger power of acceptance than itself. A certain commutativity between basic actions of the automaton implies the equivalence between an automaton with ε -moves and its ε -reduction. However, an universal validity of such an equivalence requires that the underlying logic degenerates to the classical Boolean logic.

Theorem 3.15. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let \rightarrow be an implication operator satisfying the Birkhoff-von Neumann requirement.

(1) For any ℓ -valued automaton \mathfrak{R} with ε -moves over Σ , and for any $s \in \Sigma^*$, we have:

$$\models^\ell \text{rec}_{\mathfrak{R}}^{[D]}(s) \rightarrow \text{rec}_{\mathfrak{R}^\varepsilon}^{[D]}(s).$$

(2) For any ℓ -valued automaton \mathfrak{R} with ε -moves over Σ , and for any $s \in \Sigma^*$,

$$\models^\ell \gamma(\text{atom}(\mathfrak{R})) \wedge \text{rec}_{\mathfrak{R}^\varepsilon}^{[D]}(s) \rightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s),$$

$$\models^\ell \gamma(\text{atom}(\mathfrak{R})) \wedge \text{rec}_{\mathfrak{R}}^{[W]}(s) \rightarrow \text{rec}_{\mathfrak{R}^\varepsilon}^{[W]}(s),$$

$$\models^\ell \gamma(\text{atom}(\mathfrak{R})) \wedge \text{rec}_{\mathfrak{R}^\varepsilon}^{[W]}(s) \rightarrow \text{rec}_{\mathfrak{R}}^{[W]}(s),$$

and in particular if $\rightarrow = \rightarrow_3$ then

$$\models^\ell \gamma(\text{atom}(\mathfrak{R})) \rightarrow (\text{rec}_{\mathfrak{R}}^{[D]}(s) \leftrightarrow \text{rec}_{\mathfrak{R}^\varepsilon}^{[D]}(s)),$$

$$\models^\ell \gamma(\text{atom}(\mathfrak{R})) \rightarrow (\text{rec}_{\mathfrak{R}}^{[W]}(s) \leftrightarrow \text{rec}_{\mathfrak{R}^\varepsilon}^{[W]}(s)).$$

(3) The following four statements are equivalent:

(3.1) ℓ is a Boolean algebra;

(3.2) for all ℓ -valued automaton \mathfrak{R} with ε -moves over Σ , and for all $s \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}}^{[D]}(s) \leftrightarrow \text{rec}_{\mathfrak{R}^\varepsilon}^{[D]}(s);$$

(3.3) for all ℓ -valued automaton \mathfrak{R} with ε -moves over Σ , and for all $s \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}}^{[W]}(s) \rightarrow \text{rec}_{\mathfrak{R}^\varepsilon}^{[W]}(s);$$

(3.4) for all ℓ -valued automaton \mathfrak{R} with ε -moves over Σ , and for all $s \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}^{-\varepsilon}}^{[W]}(s) \rightarrow \text{rec}_{\mathfrak{R}}^{[W]}(s).$$

Proof. The proof of (1) is similar to that of (2), so we omit it. We now prove (2). First, we use induction on the length $|c|$ of c to show that for any $c \in T(Q, \Sigma)$,

$$\begin{aligned} \text{Claim : } \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil \text{Path}_{\mathfrak{R}^{-\varepsilon}}(c) \rceil &\leq \bigvee \{ \lceil \text{Path}_{\mathfrak{R}}(c') \rceil : c' \in T_\varepsilon(Q, \Sigma), b(c') = b(c), \\ &e(c') = e(c) \text{ and } lb_\varepsilon(c') = lb(c) \}. \end{aligned}$$

For the case of $|c| = 1$, it is immediate from the definition of transition relation δ' in $\mathfrak{R}^{-\varepsilon}$. If $c = c'\sigma q$, then with the induction hypothesis and Lemmas 2.6 and 2.7, we have:

$$\begin{aligned} \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil \text{Path}_{\mathfrak{R}^{-\varepsilon}}(c) \rceil &= \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil \text{Path}_{\mathfrak{R}^{-\varepsilon}}(c') \rceil \wedge \delta'(e(c'), \sigma, q) \\ &= \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil \text{Path}_{\mathfrak{R}^{-\varepsilon}}(c') \rceil \wedge \bigvee_{m,n \geq 0} \delta(e(c'), \varepsilon^m \sigma \varepsilon^n, q) \\ &\leq \bigvee_{m,n \geq 0} (\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil \text{Path}_{\mathfrak{R}^{-\varepsilon}}(c') \rceil \wedge \delta(e(c'), \varepsilon^m \sigma \varepsilon^n, q)) \\ &\leq \bigvee_{m,n \geq 0} [\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \bigvee \{ \lceil \text{Path}_{\mathfrak{R}}(d') \rceil : d' \in T_\varepsilon(Q, \Sigma), b(d') = b(c'), e(d') = e(c') \\ &\quad \text{and } lb_\varepsilon(d') = lb(c') \} \wedge \delta(e(c'), \varepsilon^m \sigma \varepsilon^n, q)] \\ &\leq \bigvee \{ \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil \text{Path}_{\mathfrak{R}}(d') \rceil \wedge \delta(e(c'), \varepsilon^m \sigma \varepsilon^n, q) : m, n \geq 0, d' \in T_\varepsilon(Q, \Sigma), \\ &\quad b(d') = b(c'), e(d') = e(c'), lb_\varepsilon(d') = lb(c') \}. \end{aligned}$$

Furthermore, we know that

$$\begin{aligned} \delta(e(c'), \varepsilon^m \sigma \varepsilon^n, q) &= \bigvee \{ \delta(e(c'), \varepsilon, p_1) \wedge \delta(p_1, \varepsilon, p_2) \wedge \dots \wedge \delta(p_{m-1}, \varepsilon, p_m) \wedge \delta(p_m, \sigma, q_n) \\ &\quad \wedge \delta(q_n, \varepsilon, q_{n-1}) \wedge \dots \wedge \delta(q_2, \varepsilon, q_1) \wedge \delta(q_1, \varepsilon, q) : p_1, \dots, p_m, q_1, \dots, q_n \in Q \}. \end{aligned}$$

Again, we use Lemmas 2.6 and 2.7 and obtain

$$\begin{aligned} \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil \text{Path}_{\mathfrak{R}^{-\varepsilon}}(c) \rceil &\leq \bigvee \{ \lceil \text{Path}_{\mathfrak{R}}(d') \rceil \wedge \delta(e(c'), \varepsilon, p_1) \wedge \delta(p_1, \varepsilon, p_2) \wedge \dots \\ &\quad \wedge \delta(p_{m-1}, \varepsilon, p_m) \wedge \delta(p_m, \sigma, q_n) \wedge \delta(q_n, \varepsilon, q_{n-1}) \wedge \dots \wedge \delta(q_2, \varepsilon, q_1) \wedge \delta(q_1, \varepsilon, q) : \\ &\quad m, n \geq 0, d' \in T_\varepsilon(Q, \Sigma) \text{ with } b(d') = b(c'), e(d') = e(c') \\ &\quad \text{and } lb_\varepsilon(d') = lb(c'), p_1, \dots, p_m, q_1, \dots, q_n \in Q \}. \end{aligned}$$

We put $d = d' \varepsilon p_1 \varepsilon p_2 \dots \varepsilon p_{m-1} \varepsilon p_m \sigma q_n \varepsilon q_{n-1} \dots \varepsilon q_2 \varepsilon q_1 \varepsilon q$. Then $b(d) = b(d') = b(c')$, $e(d) = q = e(c)$, $lb_\varepsilon(d) = lb_\varepsilon(d') \sigma = lb(c') \sigma = lb(c)$, and

$$\begin{aligned} \lceil \text{Path}_{\mathfrak{R}}(d) \rceil &= \lceil \text{Path}_{\mathfrak{R}}(d') \rceil \wedge \delta(e(c'), \varepsilon, p_1) \wedge \delta(p_1, \varepsilon, p_2) \wedge \dots \wedge \delta(p_{m-1}, \varepsilon, p_m) \\ &\quad \wedge \delta(p_m, \sigma, q_n) \wedge \delta(q_n, \varepsilon, q_{n-1}) \wedge \dots \wedge \delta(q_2, \varepsilon, q_1) \wedge \delta(q_1, \varepsilon, q). \end{aligned}$$

Therefore,

$$\begin{aligned} [\gamma(\text{atom}(\mathfrak{R}))] \wedge [\text{Path}_{\mathfrak{R}-\varepsilon}(c)] &\leq \bigvee \{ [\text{Path}_{\mathfrak{R}}(d)] : d \in T_\varepsilon(Q, \Sigma), b(d) = b(c), \\ &\quad e(d) = e(c) \text{ and } lb_\varepsilon(d) = lb(c) \} \end{aligned}$$

and the claim holds for the case of $|c| = |c'| + 1$.

Now it follows from the above claim and Lemmas 2.6 and 2.7 that

$$\begin{aligned} [\gamma(\text{atom}(\mathfrak{R}))] \wedge [\text{rec}_{\mathfrak{R}-\varepsilon}^{[D]}(s)] &= [\gamma(\text{atom}(\mathfrak{R}))] \wedge [\gamma(\text{atom}(\mathfrak{R}))] \wedge \bigvee \{ I(b(c)) \wedge T'(e(c)) \\ &\quad \wedge [\text{Path}_{\mathfrak{R}-\varepsilon}(c)] : c \in T(Q, \Sigma), lb(c) = s \} \\ &\leq \bigvee \{ [\gamma(\text{atom}(\mathfrak{R}))] \wedge I(b(c)) \wedge T'(e(c)) \wedge [\text{Path}_{\mathfrak{R}-\varepsilon}(c)] : c \in T(Q, \Sigma), lb(c) = s \} \\ &\leq \bigvee \{ [\gamma(\text{atom}(\mathfrak{R}))] \wedge I(b(c)) \wedge T'(e(c)) \wedge \bigvee \{ [\text{Path}_{\mathfrak{R}}(c')] : c' \in T_\varepsilon(Q, \Sigma), b(c') = b(c), \\ &\quad e(c') = e(c) \text{ and } lb_\varepsilon(c') = lb(c) \} : c \in T(Q, \Sigma), lb(c) = s \} \\ &\leq \bigvee \{ [\gamma(\text{atom}(\mathfrak{R}))] \wedge I(b(c)) \wedge T'(e(c)) \wedge [\text{Path}_{\mathfrak{R}}(c')] : c \in T(Q, \Sigma), \\ &\quad c' \in T_\varepsilon(Q, \Sigma), b(c') = b(c), e(c') = e(c) \text{ and } lb_\varepsilon(c') = lb(c) = s \}. \end{aligned}$$

Note that for any $c \in T(Q, \Sigma)$ and $c' \in T_\varepsilon(Q, \Sigma)$ with $b(c') = b(c)$, $e(c') = e(c)$ and $lb_\varepsilon(c') = lb(c) = s$, we have:

$$\begin{aligned} &[\gamma(\text{atom}(\mathfrak{R}))] \wedge I(b(c)) \wedge \bigvee_{q \in Q, m \geq 0} (T(q) \wedge \delta(e(c), \varepsilon^m, q)) \wedge [\text{Path}_{\mathfrak{R}}(c')] \\ &\leq \bigvee_{q \in Q, m \geq 0} (I(b(c)) \wedge T(q) \wedge [\text{Path}_{\mathfrak{R}}(c')] \wedge \delta(e(c), \varepsilon^m, q)) \\ &\leq \bigvee_{q, q_1, \dots, q_{m-1} \in Q, m \geq 0} \{ [\gamma(\text{atom}(\mathfrak{R}))] \wedge I(b(c)) \wedge T(q) \wedge [\text{Path}_{\mathfrak{R}}(c')] \\ &\quad \wedge \delta(e(c), \varepsilon, q_1) \wedge \delta(q_1, \varepsilon, q_2) \wedge \dots \wedge \delta(q_{m-2}, \varepsilon, q_{m-1}) \wedge \delta(q_{m-1}, \varepsilon, q) \}. \end{aligned}$$

If we write $d = c' \varepsilon q_1 \varepsilon q_2 \dots \varepsilon q_{m-2} \varepsilon q_{m-1} \varepsilon q$, then $b(d) = b(c') = b(c)$, $e(d) = q$, $lb_\varepsilon(d) = lb(c') = s$ and

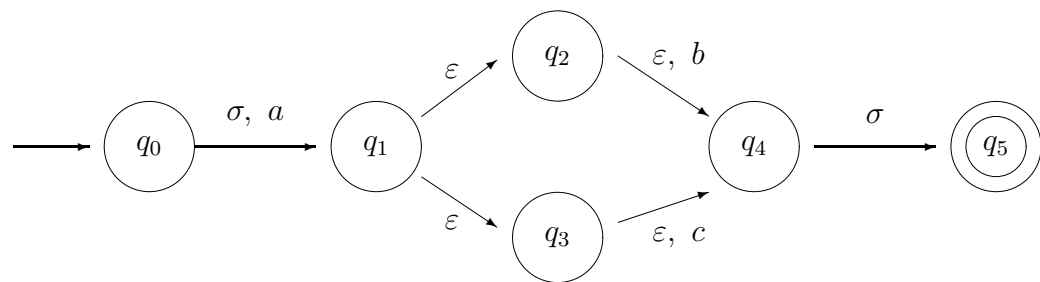
$$[\text{Path}_{\mathfrak{R}}(d)] = [\text{Path}_{\mathfrak{R}}(c')] \wedge \delta(e(c), \varepsilon, q_1) \wedge \delta(q_1, \varepsilon, q_2) \wedge \dots \wedge \delta(q_{m-2}, \varepsilon, q_{m-1}) \wedge \delta(q_{m-1}, \varepsilon, q).$$

Thus, by the definition of T' it is easy to see that

$$\begin{aligned} [\gamma(\text{atom}(\mathfrak{R}))] \wedge [\text{rec}_{\mathfrak{R}-\varepsilon}^{[D]}(s)] &\leq \bigvee \{ I(b(d)) \wedge T(e(d)) \wedge [\text{Path}_{\mathfrak{R}}(d)] : d \in T_\varepsilon(Q, \Sigma), lb_\varepsilon(d) = s \} \\ &= [\text{rec}_{\mathfrak{R}}^{[D]}(s)]. \end{aligned}$$

The conclusions for recognizability in the width-first way may also be proved by repeated applications of Lemmas 2.6 and 2.7, and we omit the details here.

For (3), the part from (3.1) to (3.2), (3.3) or (3.4) is immediate from (2) by noting that $[\gamma(\text{atom}(\mathfrak{R}))] = 1$ always holds in a Boolean algebra ℓ . Conversely, we



demonstrate that (3.2) implies (3.1). For any $a, b, c \in L$, consider ℓ -valued automaton $\mathfrak{R} = \langle \{q_0, q_1, \dots, q_5\}, \delta, \{q_0\}, \{q_5\} \rangle$ with ε -moves in which $\sigma \in \Sigma$, $\delta(q_0, \sigma, q_1) = a$, $\delta(q_1, \varepsilon, q_2) = \delta(q_1, \varepsilon, q_3) = \delta(q_4, \sigma, q_5) = 1$, $\delta(q_2, \varepsilon, q_4) = b$, $\delta(q_3, \varepsilon, q_4) = c$, and δ takes values 0 for other arguments (see Figure 4).

By a routine calculation we know that its ε -reduction is $\mathfrak{R}^{-\varepsilon} = \langle \{q_0, q_1, \dots, q_5\}, \delta', \{q_0\}, \{q_5\} \rangle$ where $\delta'(q_0, \sigma, q_1) = \delta'(q_0, \sigma, q_2) = \delta'(q_0, \sigma, q_3) = a$, $\delta'(q_0, \sigma, q_4) = (a \wedge b) \vee (a \wedge c)$, $\delta'(q_1, \sigma, q_5) = b \vee c$, $\delta'(q_2, \sigma, q_5) = b$, $\delta'(q_3, \sigma, q_5) = c$, $\delta'(q_4, \sigma, q_5) = 1$, and δ takes value 0 for other arguments (see Figure 5). Then it follows from (3.2) that

$$\begin{aligned} a \wedge (b \vee c) &= [a \wedge (b \vee c)] \vee (a \wedge b) \vee (a \wedge c) \vee [(a \wedge b) \vee (a \wedge c)] \\ &= [\text{rec}_{\mathfrak{R}^{-\varepsilon}}^{[D]}(\sigma\sigma)] \\ &= [\text{rec}_{\mathfrak{R}}^{[D]}(\sigma\sigma)] = (a \wedge b) \vee (a \wedge c). \end{aligned}$$

This shows that ℓ enjoys the distributivity of meet over union, and it is a Boolean algebra.

To show that (3.3) implies (3.1), let $\mathfrak{R} = \langle \{q_0, q_1, q_2, q_3, q_4\}, \delta, \{q_0\}, \{q_4\} \rangle$, where $\delta(q_0, \varepsilon, q_1) = b$, $\delta(q_0, \varepsilon, q_2) = c$, $\delta(q_1, \varepsilon, q_3) = \delta(q_2, \varepsilon, q_3) = 1$ and $\delta(q_3, \sigma, q_4) = a$. Then $\text{rec}_{\mathfrak{R}}^{[W]}(\sigma^2) = a \wedge (b \vee c)$ and $\text{rec}_{\mathfrak{R}^{-\varepsilon}}^{[W]}(\sigma^2) = (a \wedge b) \vee (a \wedge c)$.

The implication from (3.4) to (3.1) can be proved by a similar construction. Put $\mathfrak{R} = \langle \{q_0, q_1, q_2, q_3, p_1, p_2\}, \delta, \{q_0\}, \{q_3\} \rangle$, $\delta(q_0, \sigma, q_1) = a$, $\delta(q_1, \sigma, q_2) = \delta(p_1, \varepsilon, q_3) = \delta(p_2, \varepsilon, q_3) = 1$, $\delta(q_2, \varepsilon, p_1) = b$ and $\delta(q_2, \varepsilon, p_2) = c$, and let δ take value 0 for other cases. Then $\text{rec}_{\mathfrak{R}}^{[W]}(\sigma^2) = (a \wedge b) \vee (a \wedge c)$ and $\text{rec}_{\mathfrak{R}^{-\varepsilon}}^{[W]}(\sigma^2) = a \wedge (b \vee c)$. \square

3.4. Closure Properties of Orthomodular Lattice-Valued Regularity

It was shown in the classical automata theory that the class of regular languages is closed under various operations such as union, intersection, complement, concatenation, the Kleene closure, substitution and homomorphism. In this section, we are going to examine the closure properties of orthomodular lattice-valued regular languages under these operations.

We first consider the inverse of an ℓ -valued language. Let $A \in L^{\Sigma^*}$. Then the inverse $A^{-1} \in L^{\Sigma^*}$ of A is defined as follows: $A^{-1}(\sigma_1 \dots \sigma_m) = A(\sigma_m \dots \sigma_1)$ for any $m \in \omega$ and for any $\sigma_1, \dots, \sigma_m \in \Sigma$.

The following proposition shows that regularity and commutative regularity, both in the depth-first way and the width-first way, are preserved by the inverse operation.

Proposition 3.16. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be a complete orthomodular lattice, and let \rightarrow fulfil the property that $a \leftrightarrow a = 1$ for any $a \in L$. Then for any

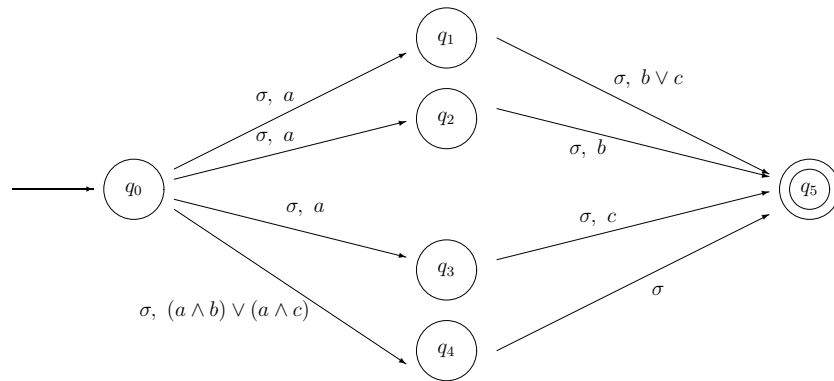


Figure 5: Automaton c

$A \in L^{\Sigma^*}$, we have:

$$\begin{aligned} \models^\ell \text{Reg}_\Sigma^{[D]}(A) &\leftrightarrow \text{Reg}_\Sigma^{[D]}(A^{-1}) \text{ and } \models^\ell \text{Reg}_\Sigma^{[W]}(A) \leftrightarrow \text{Reg}_\Sigma^{[W]}(A^{-1}); \\ \models^\ell \text{CReg}_\Sigma^{[D]}(A) &\leftrightarrow \text{CReg}_\Sigma^{[D]}(A^{-1}) \text{ and } \models^\ell \text{CReg}_\Sigma^{[W]}(A) \leftrightarrow \text{CReg}_\Sigma^{[W]}(A^{-1}). \end{aligned}$$

Proof. We first consider $\text{Reg}_\Sigma^{[D]}$. Noting that $A = (A^{-1})^{-1}$, it suffices to show that

$$[\text{Reg}_\Sigma^{[D]}(A)] \leq [\text{Reg}_\Sigma^{[D]}(A^{-1})].$$

For any ℓ -valued automaton $\mathfrak{R} = \langle Q, \delta, I, T \rangle$, we define the inverse of $\mathfrak{R} = \langle Q, \delta, I, T \rangle$ to be the ℓ -valued automaton $\mathfrak{R}^{-1} = \langle Q, \delta^{-1}, T, I, \rangle$, where $\delta^{-1}(p, \sigma, q) = \delta(q, \sigma, p)$ for any $p, q \in Q$ and $\sigma \in \Sigma$. Then it is easy to see that $\text{rec}_{\mathfrak{R}^{-1}}^{[D]} = (\text{rec}_{\mathfrak{R}}^{[D]})^{-1}$, and furthermore we have:

$$\begin{aligned} [\text{Reg}_\Sigma^{[D]}(A)] &= \bigvee \{ [A \equiv \text{rec}_{\mathfrak{R}}^{[D]}] : \mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell) \} \\ &= \bigvee \{ [A^{-1} \equiv (\text{rec}_{\mathfrak{R}}^{[D]})^{-1}] : \mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell) \} \\ &= \bigvee \{ [A^{-1} \equiv \text{rec}_{\mathfrak{R}^{-1}}^{[D]}] : \mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell) \} \\ &\leq \bigvee \{ [A^{-1} \equiv \text{rec}_{\emptyset}^{[D]}] : \emptyset \in \mathbf{NFA}(\Sigma, \ell) \} = [\text{Rec}_\Sigma^{[D]}(A^{-1})]. \end{aligned}$$

The proof for other versions of regularity is similar. \square

The commutative regularity is preserved by the complement operation, but it is not the case for the (noncommutative) regularity predicate.

Proposition 3.17. If $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ is a finite orthomodular lattice, and $\rightarrow = \rightarrow_3$, then for any $A \in L^{\Sigma^*}$, we have:

$$\models^\ell \text{CReg}_\Sigma^{[D]}(A) \rightarrow \text{CReg}_\Sigma^{[D]}(A^c).$$

The same conclusion is valid for $\text{CReg}_\Sigma^{[W]}$.

Proof. For any $\mathfrak{R} = \langle Q, \delta, I, T \rangle \in \mathbf{NFA}(\Sigma, \ell)$, we observe that $\ell^{\mathfrak{R}} = \langle L^Q, \bar{\delta}, I_1, \bar{T} \rangle$ is an ℓ -valued deterministic automaton and only \bar{T} carries ℓ -valued information. Then we set $(\ell^{\mathfrak{R}})^c = \langle L^Q, \bar{\delta}, I_1, \bar{T}^c \rangle$, where for any $X \in L^Q$, $\bar{T}^c(X) = (\bar{T}(X))^c$. It is easy to see that for all $s \in \Sigma^*$, $\text{rec}_{(\ell^{\mathfrak{R}})^c}^{[D]}(s) = (\text{rec}_{\ell^{\mathfrak{R}}}^{[D]}(s))^\perp$.

Now by using Theorem 3.11 and Lemmas 2.6 and 2.7 we obtain:

$$\begin{aligned}
& [\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge [A \equiv \text{rec}_{\mathfrak{R}}^{[D]}]] \leq [\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge [A \equiv \text{rec}_{\mathfrak{R}}^{[D]}]] \wedge [\text{rec}_{\mathfrak{R}}^{[D]} \equiv \text{rec}_{\ell^{\mathfrak{R}}}^{[D]}] \\
& = \bigwedge_{s \in \Sigma^*} ([\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge (A(s) \rightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s)) \wedge (\text{rec}_{\mathfrak{R}}^{[D]}(s) \rightarrow \text{rec}_{\ell^{\mathfrak{R}}}^{[D]}(s))] \wedge \\
& \quad \bigwedge_{s \in \Sigma^*} ([\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge (\text{rec}_{\ell^{\mathfrak{R}}}^{[D]}(s) \rightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s)) \wedge (\text{rec}_{\mathfrak{R}}^{[D]}(s) \rightarrow A(s))] \\
& \leq \bigwedge_{s \in \Sigma^*} ([\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge (A(s) \rightarrow \text{rec}_{\ell^{\mathfrak{R}}}^{[D]}(s))] \wedge \\
& \quad \bigwedge_{s \in \Sigma^*} ([\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge (\text{rec}_{\ell^{\mathfrak{R}}}^{[D]}(s) \rightarrow A(s))].
\end{aligned}$$

Then Lemma 2.12(2) yields:

$$\begin{aligned}
[\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge [A \equiv \text{rec}_{\mathfrak{R}}^{[D]}]] & \leq \bigwedge_{s \in \Sigma^*} ((\text{rec}_{\ell^{\mathfrak{R}}}^{[D]}(s))^{\perp} \rightarrow A^{\perp}(s)) \wedge \bigwedge_{s \in \Sigma^*} (A^{\perp}(s) \rightarrow (\text{rec}_{\ell^{\mathfrak{R}}}^{[D]}(s))^{\perp}) \\
& = \bigwedge_{s \in \Sigma^*} (A^{\perp}(s) \leftrightarrow (\text{rec}_{\ell^{\mathfrak{R}}}^{[D]}(s))^{\perp}) \\
& = \bigwedge_{s \in \Sigma^*} (A^{\perp}(s) \leftrightarrow \text{rec}_{(\ell^{\mathfrak{R}})^c}^{[D]}(s)) \\
& = [A^c \equiv \text{rec}_{(\ell^{\mathfrak{R}})^c}^{[D]}].
\end{aligned}$$

In addition, we have

$$[\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \leq [\gamma(\text{atom}(\mathfrak{R}) \cup r(A^c))]$$

from Lemma 2.6. Therefore,

$$\begin{aligned}
[\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge [A \equiv \text{rec}_{\mathfrak{R}}^{[D]}]] & \leq [\gamma(\text{atom}(\mathfrak{R}) \cup r(A^c)) \wedge [A^c \equiv \text{rec}_{(\ell^{\mathfrak{R}})^c}^{[D]}]] \\
& \leq [CReg_{\Sigma}^{[D]}(A^c)].
\end{aligned}$$

Finally, since \mathfrak{R} is allowed to be arbitrary, it follows that

$$\begin{aligned}
[CReg_{\Sigma}^{[D]}(A)] & = \bigvee_{\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)} [\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge [A \equiv \text{rec}_{\mathfrak{R}}^{[D]}]] \\
& \leq [CReg_{\Sigma}^{[D]}(A^c)]. \quad \square
\end{aligned}$$

We now turn to deal with recognizability of the union of two ℓ -valued languages. For this purpose, we first introduce the union operation of ℓ -valued finite automata. Let $\mathfrak{R} = \langle Q_A, \delta_A, I_A, T_A \rangle$ and $\wp = \langle Q_B, \delta_B, I_B, T_B \rangle \in \mathbf{NFA}(\Sigma, \ell)$ be two ℓ -valued finite automata over Σ . We assume that $Q_A \cap Q_B = \emptyset$. Then the (disjoint) union $\mathfrak{R} \cup \wp$ of \mathfrak{R} and \wp is defined to be $\mathfrak{S} = \langle Q_C, \delta_C, I_C, T_C \rangle$, where:

- (i) $Q_C = Q_A \cup Q_B$;
- (ii) $I_C = I_A \cup I_B$;
- (iii) $T_C = T_A \cup T_B$; and
- (iv) $\delta_C : Q_C \times \Sigma \times Q_C \rightarrow L$ is given as follows: for any $p, q \in Q_C$ and $\sigma \in \Sigma$,

$$\delta_C(p, \sigma, q) = \begin{cases} \delta_A(p, \sigma, q), & \text{if } p, q \in Q_A, \\ \delta_B(p, \sigma, q), & \text{if } p, q \in Q_B, \\ 0, & \text{otherwise.} \end{cases}$$

The following proposition describes the recognizability of the union of two ℓ -valued automata. As in the classical automata theory, a word s in Σ^* is recognized by the union of two ℓ -valued automata if and only if s is recognized by one of them, no matter the depth-first or width-first principle is adopted.

Proposition 3.18. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be a complete orthomodular lattice. If the implication operator \rightarrow satisfies that $a \leftrightarrow a = 1$ for any $a \in L$, then for any $\mathfrak{R}, \wp \in \mathbf{NFA}(\Sigma, \ell)$ and for any $s \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R} \cup \wp}(s) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s) \vee \text{rec}_{\wp}^{[D]}(s) \text{ and } \models^\ell \text{rec}_{\mathfrak{R} \cup \wp}(s) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[W]}(s) \vee \text{rec}_{\wp}^{[W]}(s).$$

Proof. We only prove the conclusion in the depth-first way, and the other case is left for the reader. Let $s = \sigma_1 \dots \sigma_k$. Then

$$\begin{aligned} [\text{rec}_{\mathfrak{R} \cup \wp}^{[D]}(s)] &= \bigvee \{ (I_A \cup I_B)(q_0) \wedge (T_A \cup T_B)(q_k) \wedge \bigwedge_{i=0}^{k-1} \delta_{A \cup B}(q_i, \sigma_{i+1}, q_{i+1}) : q_0, q_1, \dots, q_k \in Q_A \cup Q_B \} \\ &= [\bigvee \{ (I_A \cup I_B)(q_0) \wedge (T_A \cup T_B)(q_k) \wedge \bigwedge_{i=0}^{k-1} \delta_{A \cup B}(q_i, \sigma_{i+1}, q_{i+1}) : q_0, q_1, \dots, q_k \in Q_A \}] \\ &\quad \vee [\bigvee \{ (I_A \cup I_B)(q_0) \wedge (T_A \cup T_B)(q_k) \wedge \bigwedge_{i=0}^{k-1} \delta_{A \cup B}(q_i, \sigma_{i+1}, q_{i+1}) : q_0, q_1, \dots, q_k \in Q_B \}] \\ &\quad \vee [\bigvee \{ (I_A \cup I_B)(q_0) \wedge (T_A \cup T_B)(q_k) \wedge \bigwedge_{i=0}^{k-1} \delta_{A \cup B}(q_i, \sigma_{i+1}, q_{i+1}) : q_0, q_1, \dots, q_k \in Q_A \cup Q_B, \\ &\quad \text{and there are } i, j \text{ such that } 0 \leq i, j \leq k \text{ and } q_i \in Q_A \text{ and } q_j \in Q_B \}]. \end{aligned}$$

From the definition of $\mathfrak{R} \cup \wp$, we know that for any $q_0, q_1, \dots, q_k \in Q_A$,

$$\begin{aligned} (I_A \cup I_B)(q_0) &= I_A(q_0), \\ (T_A \cup T_B)(q_k) &= T_A(q_k), \\ \bigwedge_{i=0}^{k-1} \delta_{A \cup B}(q_i, \sigma_{i+1}, q_{i+1}) &= \bigwedge_{i=0}^{k-1} \delta_A(q_i, \sigma_{i+1}, q_{i+1}), \end{aligned}$$

and for any $q_0, q_1, \dots, q_k \in Q_B$,

$$(I_A \cup I_B)(q_0) = I_B(q_0),$$

$$(T_A \cup T_B)(q_k) = T_B(q_k),$$

$$\bigwedge_{i=0}^{k-1} \delta_{A \cup B}(q_i, \sigma_{i+1}, q_{i+1}) = \bigwedge_{i=0}^{k-1} \delta_B(q_i, \sigma_{i+1}, q_{i+1}).$$

If $q_0, q_1, \dots, q_k \in Q_A \cup Q_B$, and there are i, j such that $0 \leq i, j \leq k$ and $q_i \in Q_A$ and $q_j \in Q_B$, then we can find some $m \in \{0, 1, \dots, k-1\}$ such that $q_m \in Q_A$ and $q_{m+1} \in Q_B$, or $q_m \in Q_B$ and $q_{m+1} \in Q_A$. Then $\delta_{A \cup B}(q_m, \sigma_{m+1}, q_{m+1}) = 0$, and

$$\bigwedge_{i=0}^{k-1} \delta_{A \cup B}(q_i, \sigma_{i+1}, q_{i+1}) = 0.$$

Therefore, it follows that

$$\begin{aligned} [\text{rec}_{\mathfrak{R} \cup \wp}^{[D]}(s)] &= [\bigvee \{ I_A(q_0) \wedge T_A(q_k) \wedge \bigwedge_{i=0}^{k-1} \delta_A(q_i, \sigma_{i+1}, q_{i+1}) : q_0, q_1, \dots, q_k \in Q_A \}] \\ &\quad \vee [\bigvee \{ I_B(q_0) \wedge T_B(q_k) \wedge \bigwedge_{i=0}^{k-1} \delta_B(q_i, \sigma_{i+1}, q_{i+1}) : q_0, q_1, \dots, q_k \in Q_B \}] \\ &= [\text{rec}_{\mathfrak{R}}^{[D]}(s) \vee \text{rec}_{\wp}^{[D]}(s)]. \quad \square \end{aligned}$$

The following corollary slightly generalizes Example 3.5 as well as the last part of Example 3.7.

Corollary 3.19. If $\text{Range}(A) = \{A(s) : s \in \Sigma^*\}$ is finite, and $A_\lambda = \{s \in \Sigma^* : A(s) \geq \lambda\}$ is a regular language (in classical automata theory) for every $\lambda \in \text{Range}(A)$, then it holds that

$$\models^\ell \text{Reg}_\Sigma^{[D]}(A).$$

Proof. Suppose that $\text{Range}(A) = \{\lambda_1, \dots, \lambda_n\}$. Then it is easy to see that

$$A = \bigcup_{i=1}^n \lambda_i A_{\lambda_i}.$$

From Example 3.7 we know that there exists an ℓ -valued automaton \mathfrak{R}_i such that $\text{rec}_{\mathfrak{R}_i} = \lambda_i A_{\lambda_i}$ for each $i \leq n$. Thus, by proposition 3.18 we obtain

$$\text{rec}_{\bigcup_{i=1}^n \mathfrak{R}_i}^{[D]} = \bigcup_{i=1}^n \lambda_i A_{\lambda_i} = A$$

and complete the proof. \square

To present the fact that regularity is preserved by union operation of ℓ -valued languages, we need to introduce a new notion of conformal and commutative regularity.

Definition 3.20. Let Σ be a nonempty finite alphabet. The conformal and commutative regularity of depth-first is defined to be a binary ℓ -valued predicate on Σ^* , $ConCReg_{\Sigma}^{[D]} \in L^{\Sigma^* \times \Sigma^*}$, and for each $A, B \in L^{\Sigma^*}$,

$$ConCReg_{\Sigma}^{[D]}(A, B) \stackrel{def}{=} (\exists \mathfrak{R}, \wp \in \mathbf{NFA}(\Sigma, \ell)) (\gamma(atom(\mathfrak{R}) \cup atom(\wp) \cup r(A) \cup r(B)) \wedge A \equiv rec_{\mathfrak{R}}^{[D]} \wedge B \equiv rec_{\wp}^{[D]}).$$

We can also define a notion of conformal and commutative regularity in the width-first way, but we may note that it is equivalent to the one in the depth-first way whenever the implication operator is taken to be the Sasaki hook \rightarrow_3 by Lemma 3.3.

Corollary 3.21. If $\rightarrow = \rightarrow_3$, then for any $A, B \in L^{\Sigma^*}$, we have:

$$\models^{\ell} ConCReg_{\Sigma}^{[D]}(A, B) \rightarrow CReg_{\Sigma}^{[D]}(A \cup B).$$

Proof. It suffices to show that for $\mathfrak{R}, \wp \in \mathbf{NFA}(\Sigma, \ell)$,

$$[\gamma(Atom)] \wedge [A \equiv rec_{\mathfrak{R}}^{[D]}] \wedge [B \equiv rec_{\wp}^{[D]}] \leq [CReg_{\Sigma}^{[D]}(A \cup B)],$$

where $Atom = atom(\mathfrak{R}) \cup atom(\wp) \cup r(A) \cup r(B)$. In fact, it follows from Lemmas 2.7 and 2.12(1) and Proposition 3.18 that

$$\begin{aligned} & [\gamma(Atom)] \wedge [A \equiv rec_{\mathfrak{R}}^{[D]}] \wedge [B \equiv rec_{\wp}^{[D]}] = [\gamma(Atom)] \wedge \\ & \quad \bigwedge_{s \in \Sigma^*} (A(s) \leftrightarrow rec_{\mathfrak{R}}^{[D]}(s)) \wedge \bigwedge_{s \in \Sigma^*} (B(s) \leftrightarrow rec_{\wp}^{[D]}(s)) \\ & = [\gamma(Atom)] \wedge \bigwedge_{s \in \Sigma^*} [(A(s) \leftrightarrow rec_{\mathfrak{R}}^{[D]}(s)) \wedge (B(s) \leftrightarrow rec_{\wp}^{[D]}(s))] \\ & \leq [\gamma(Atom)] \wedge \bigwedge_{s \in \Sigma^*} [(A \cup B)(s) \leftrightarrow rec_{\mathfrak{R} \cup \wp}^{[D]}(s)] \\ & \leq [\gamma(atom(\mathfrak{R} \cup \wp) \cup r(A \cup B))] \wedge [A \cup B \equiv rec_{\mathfrak{R} \cup \wp}^{[D]}] \\ & \leq [CReg_{\Sigma}^{[D]}(A \cup B)]. \quad \square \end{aligned}$$

We now consider the product of two ℓ -valued automata. Let $\mathfrak{R} = \langle Q_A, \delta_A, I_A, T_A \rangle$ and $\wp = \langle Q_B, \delta_B, I_B, T_B \rangle \in \mathbf{NFA}(\Sigma, \ell)$ be two ℓ -valued automata over Σ . Then their product $\mathfrak{R} \times \wp$ is defined to be $\mathfrak{S} = \langle Q_C, \delta_C, I_C, T_C \rangle$, where:

- (i) $Q_C = Q_A \times Q_B$;
- (ii) $I_C = I_A \times I_B$;
- (iii) $T_C = T_A \times T_B$; and
- (iv) $\delta_C : Q_C \times \Sigma \times Q_C \rightarrow L$ and for any $p_a, q_a \in Q_A, p_b, q_b \in Q_B$ and $\sigma \in \Sigma$,

$$\delta_C((p_a, p_b), \sigma, (q_a, q_b)) = \delta_A(p_a, \sigma, q_a) \wedge \delta_B(p_b, \sigma, q_b).$$

It is well-known in the classical automata theory that the language accepted by the union of two finite automata is the union of the languages accepted by these two automata, and the language accepted by the product of two finite automata is the intersection of the languages accepted by the factor automata. Proposition 3.18 shows that the conclusion about the union of two automata can be generalized into the theory of automata based on quantum logic without appealing any additional condition. One may naturally expect that the conclusion for product of automata can also be easily generalized into the framework of quantum logic. However, the case for the product of two automata is much more complicated, and the following proposition tells us that in order to make the above conclusion about product of automata still valid in the automata theory based on quantum logic, a certain commutativity is necessary to be added on the basic actions of the factor automata.

Proposition 3.22. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be a complete orthomodular lattice.

- (1) For any $\mathfrak{R}, \wp \in \mathbf{NFA}(\Sigma, \ell)$, and for any $s \in \Sigma^*$, we have:

$$\models^\ell \text{rec}_{\mathfrak{R} \times \wp}^{[D]}(s) \rightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s) \wedge \text{rec}_{\wp}^{[D]}(s) \text{ and } \models^\ell \text{rec}_{\mathfrak{R} \times \wp}^{[W]}(s) \rightarrow \text{rec}_{\mathfrak{R}}^{[W]}(s) \wedge \text{rec}_{\wp}^{[W]}(s).$$

- (2) For any $\mathfrak{R}, \wp \in \mathbf{NFA}(\Sigma, \ell)$, and for any $s \in \Sigma^*$, we have:

$$\begin{aligned} & \models^\ell \gamma(\text{atom}(\mathfrak{R}) \cup \text{atom}(\wp)) \wedge \text{rec}_{\mathfrak{R}}^{[D]}(s) \wedge \text{rec}_{\wp}^{[D]}(s) \rightarrow \text{rec}_{\mathfrak{R} \times \wp}^{[D]}(s), \\ & \models^\ell \gamma(\text{atom}(\mathfrak{R}) \cup \text{atom}(\wp)) \wedge \text{rec}_{\mathfrak{R}}^{[W]}(s) \wedge \text{rec}_{\wp}^{[W]}(s) \rightarrow \text{rec}_{\mathfrak{R} \times \wp}^{[W]}(s), \end{aligned}$$

and in particular if $\rightarrow = \rightarrow_3$, then

$$\begin{aligned} & \models^\ell \gamma(\text{atom}(\mathfrak{R}) \cup \text{atom}(\wp)) \rightarrow (\text{rec}_{\mathfrak{R}}^{[D]}(s) \wedge \text{rec}_{\wp}^{[D]}(s) \leftrightarrow \text{rec}_{\mathfrak{R} \times \wp}^{[D]}(s)), \\ & \models^\ell \gamma(\text{atom}(\mathfrak{R}) \cup \text{atom}(\wp)) \rightarrow (\text{rec}_{\mathfrak{R}}^{[W]}(s) \wedge \text{rec}_{\wp}^{[W]}(s) \leftrightarrow \text{rec}_{\mathfrak{R} \times \wp}^{[W]}(s)). \end{aligned}$$

- (3) The following three statements are equivalent:

- (3.1) ℓ is a Boolean algebra.

(3.2) for all $\mathfrak{R}, \wp \in \mathbf{NFA}(\Sigma, \ell)$, and for all $s \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}}^{[D]}(s) \wedge \text{rec}_{\wp}^{[D]}(s) \leftrightarrow \text{rec}_{\mathfrak{R} \times \wp}^{[D]}(s).$$

(3.3) for all $\mathfrak{R}, \wp \in \mathbf{NFA}(\Sigma, \ell)$, and for all $s \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}}^{[W]}(s) \wedge \text{rec}_{\wp}^{[W]}(s) \leftrightarrow \text{rec}_{\mathfrak{R} \times \wp}^{[W]}(s).$$

Proof. We first prove (1) and (2) for recognizability in the depth-first way. We have directly:

$$\begin{aligned} [\text{rec}_{\mathfrak{R} \times \wp}^{[D]}(s)] &= \bigvee \left\{ (I_A \times I_B)(q_{a0}, q_{b0}) \wedge (T_A \times T_B)(q_{ak}, q_{bk}) \wedge \bigwedge_{i=0}^{k-1} \delta_{A \times B}((q_{ai}, q_{bi}), \right. \\ &\quad \left. \sigma_{i+1}, (q_{a(i+1)}, q_{b(i+1)})) : q_{a0}, q_{a1}, \dots, q_{ak} \in Q_A \text{ and } q_{b0}, q_{b1}, \dots, q_{bk} \in Q_B \right\} \\ &= \bigvee \left\{ I_A(q_{a0}) \wedge I_B(q_{b0}) \wedge T_A(q_{ak}) \wedge T_B(q_{bk}) \wedge \bigwedge_{i=0}^{k-1} \delta_A(q_{ai}, \sigma_{i+1}, q_{a(i+1)}) \wedge \right. \\ &\quad \left. \bigwedge_{i=0}^{k-1} \delta_B(q_{bi}, \sigma_{i+1}, q_{b(i+1)}) : q_{a0}, q_{a1}, \dots, q_{ak} \in Q_A \text{ and } q_{b0}, q_{b1}, \dots, q_{bk} \in Q_B \right\}, \end{aligned}$$

and

$$\begin{aligned} [\text{rec}_{\mathfrak{R}}^{[D]}(s) \wedge \text{rec}_{\wp}^{[D]}(s)] &= \left[\bigvee \left\{ I_A(q_0) \wedge T_A(q_k) \wedge \bigwedge_{i=0}^{k-1} \delta_A(q_i, \sigma_{i+1}, q_{i+1}) : q_0, q_1, \dots, q_k \in Q_A \right\} \right] \\ &\quad \wedge \left[\bigvee \left\{ I_B(q_0) \wedge T_B(q_k) \wedge \bigwedge_{i=0}^{k-1} \delta_B(q_i, \sigma_{i+1}, q_{i+1}) : q_0, q_1, \dots, q_k \in Q_B \right\} \right] \end{aligned}$$

from the definitions of product and recognizability of ℓ -valued automata. It is clear that

$$[\text{rec}_{\mathfrak{R} \times \wp}^{[D]}(s)] \leq [\text{rec}_{\mathfrak{R}}^{[D]}(s) \wedge \text{rec}_{\wp}^{[D]}(s)].$$

This indicates that (1) is true for recognizability of depth-first. By using Lemmas 2.5(2), 2.6 and 2.7 twice, we can deduce that

$$[\gamma(\text{atom}(\mathfrak{R}) \cup \text{atom}(\wp)) \wedge \text{rec}_{\mathfrak{R}}^{[D]}(s) \wedge \text{rec}_{\wp}^{[D]}(s)] \leq [\text{rec}_{\mathfrak{R} \times \wp}^{[D]}(s)].$$

Thus, (2) is proved for recognizability of depth-first.

Similarly, we are able to prove (1) and (2) for recognizability of width-first.

The part of (3) that (3.1) implies (3.2) and (3.3) is immediately derivable from (2) because we have $[\gamma(\text{atom}(\mathfrak{R}) \cup \text{atom}(\wp))] = 1$ provided ℓ is a Boolean algebra. Conversely, we show that (3.2) implies (3.1) by constructing two ℓ -valued automata and examining the behavior of their product. For any $a, b, c \in L$, we choose some

$\sigma_0 \in \Sigma$ and set $\mathfrak{R} = \langle \{p\}, \delta_A, \{p\}, \{p\} \rangle$, where $\delta_A(p, \sigma, p) = a$ if $\sigma = \sigma_0$ and 0 otherwise, and $\wp = \langle \{q, r, s\}, \delta_B, \{q\}, \{r, s\} \rangle$, where $\delta_B(x, \sigma, y) = b$ if $x = q, y = r$, and $\sigma = \sigma_0$; c if $x = q, y = s$, and $\sigma = \sigma_0$; 0 otherwise. Then $\mathfrak{R}, \wp \in \mathbf{NFA}(\Sigma, \ell)$, and it is easy to see that

$$\mathfrak{R} \times \wp = \langle \{(p, q), (p, r), (p, s)\}, \delta_{A \times B}, \{(p, q)\}, \{(p, r), (p, s)\} \rangle,$$

where $\delta_{A \times B}((p, x), \sigma, (p, y)) = a \wedge b$ if $x = q, y = r$ and $\sigma = \sigma_0$; $a \wedge c$ if $x = q, y = s$ and $\sigma = \sigma_0$; and 0 otherwise (see Figure 6). Furthermore, by a routine calculation we have:

$$\begin{aligned} [\text{rec}_{\mathfrak{R}}^{[D]}(\sigma_0)] &= a, \\ [\text{rec}_{\wp}^{[D]}(\sigma_0)] &= b \vee c, \text{ and} \\ [\text{rec}_{\mathfrak{R} \times \wp}^{[D]}(\sigma_0)] &= (a \wedge b) \vee (a \wedge c). \end{aligned}$$

Therefore, with (3.2) we finally obtain

$$\begin{aligned} a \wedge (b \vee c) &= [\text{rec}_{\mathfrak{R}}^{[D]}(\sigma_0)] \wedge [\text{rec}_{\wp}^{[D]}(\sigma_0)] \\ &= [\text{rec}_{\mathfrak{R} \times \wp}^{[D]}(\sigma_0)] = (a \wedge b) \vee (a \wedge c). \end{aligned}$$

Note that for any $\mathfrak{R} = \langle Q, \delta, I, T \rangle \in \mathbf{NFA}(\Sigma, \ell)$, it holds that $[\text{rec}_{\mathfrak{R}}^{[W]}(s)] = [\text{rec}_{\mathfrak{R}}^{[D]}(s)]$ if $|s| = 1$ and T is a classical (two-valued) subset of Q . Thus, the above construction can also be used to show that (3.3) implies (3.1). \square

Corollary 3.23. If $\rightarrow = \rightarrow_3$ then for any $A, B \in L^{\Sigma^*}$, we have:

$$\models^{\ell} \text{ConCReg}_{\Sigma}^{[D]}(A, B) \rightarrow \text{CReg}_{\Sigma}^{[D]}(A \cap B).$$

Proof. Similar to Corollary 3.21. \square

To prove the closure property of orthomodular lattice-valued regularity under the concatenation operation of languages, we propose the concept of concatenation of two orthomodular lattice-valued automata. Suppose that $\mathfrak{R}_1 = \langle Q_1, \delta_1, I_1, T_1 \rangle$, $\mathfrak{R}_2 = \langle Q_2, \delta_2, I_2, T_2 \rangle \in \mathbf{NFA}(\Sigma, \ell)$ be two ℓ -valued finite automata, and $Q_1 \cap Q_2 = \emptyset$. We define the concatenation of \mathfrak{R}_1 and \mathfrak{R}_2 to be ℓ -valued automaton $\mathfrak{R}_1 \mathfrak{R}_2 = \langle Q_1 \cup Q_2, \delta, I_1, T_2 \rangle$ with ε -moves, where $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times Q \rightarrow L$ is given by

$$\delta(p, \sigma, q) = \begin{cases} \delta_1(p, \sigma, q), & \text{if } p, q \in Q_1 \text{ and } \sigma \neq \varepsilon, \\ \delta_2(p, \sigma, q), & \text{if } p, q \in Q_2 \text{ and } \sigma \neq \varepsilon, \\ T_1(p) \wedge I_2(q), & \text{if } p \in Q_1, q \in Q_2 \text{ and } \sigma = \varepsilon, \\ 0, & \text{otherwise.} \end{cases}$$

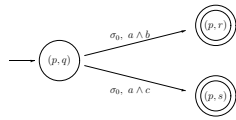


Figure 6: Automaton d

The following proposition clarifies the relation between the language recognized by the concatenation of two orthomodular lattice-valued automata and the concatenation of the languages recognized by the two automata.

Proposition 3.24. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice and \rightarrow fulfil the Birkhoff-von Neumann requirement.

(1) For all $\mathfrak{R}_1, \mathfrak{R}_2 \in \mathbf{NFA}(\Sigma, \ell)$, and for each $s \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}_1 \mathfrak{R}_2}^{[D]}(s) \rightarrow (\text{rec}_{\mathfrak{R}_1}^{[D]} \cdot \text{rec}_{\mathfrak{R}_2}^{[D]})(s) \text{ and } \models^\ell \text{rec}_{\mathfrak{R}_1 \mathfrak{R}_2}^{[W]}(s) \rightarrow (\text{rec}_{\mathfrak{R}_1}^{[W]} \cdot \text{rec}_{\mathfrak{R}_2}^{[W]})(s).$$

(2) For all $\mathfrak{R}_1, \mathfrak{R}_2 \in \mathbf{NFA}(\Sigma, \ell)$, and for each $s \in \Sigma^*$,

$$\models^\ell \gamma(\text{atom}(\mathfrak{R}_1) \cup \text{atom}(\mathfrak{R}_2)) \wedge (\text{rec}_{\mathfrak{R}_1}^{[D]} \cdot \text{rec}_{\mathfrak{R}_2}^{[D]})(s) \rightarrow \text{rec}_{\mathfrak{R}_1 \mathfrak{R}_2}^{[D]}(s),$$

$$\models^\ell \gamma(\text{atom}(\mathfrak{R}_1) \cup \text{atom}(\mathfrak{R}_2)) \wedge (\text{rec}_{\mathfrak{R}_1}^{[W]} \cdot \text{rec}_{\mathfrak{R}_2}^{[W]})(s) \rightarrow \text{rec}_{\mathfrak{R}_1 \mathfrak{R}_2}^{[W]}(s),$$

and if $\rightarrow = \rightarrow_3$ then

$$\models^\ell \gamma(\text{atom}(\mathfrak{R}_1) \cup \text{atom}(\mathfrak{R}_2)) \rightarrow (\text{rec}_{\mathfrak{R}_1 \mathfrak{R}_2}^{[D]}(s) \leftrightarrow (\text{rec}_{\mathfrak{R}_1}^{[D]} \cdot \text{rec}_{\mathfrak{R}_2}^{[D]})(s)),$$

$$\models^\ell \gamma(\text{atom}(\mathfrak{R}_1) \cup \text{atom}(\mathfrak{R}_2)) \rightarrow (\text{rec}_{\mathfrak{R}_1 \mathfrak{R}_2}^{[W]}(s) \leftrightarrow (\text{rec}_{\mathfrak{R}_1}^{[W]} \cdot \text{rec}_{\mathfrak{R}_2}^{[W]})(s)).$$

(3) The following three statements are equivalent:

(3.1) ℓ is a Boolean algebra;

(3.2) for all $\mathfrak{R}_1, \mathfrak{R}_2 \in \mathbf{NFA}(\Sigma, \ell)$, and for each $s \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}_1 \mathfrak{R}_2}^{[D]}(s) \leftrightarrow (\text{rec}_{\mathfrak{R}_1}^{[D]} \cdot \text{rec}_{\mathfrak{R}_2}^{[D]})(s).$$

(3.3) for all $\mathfrak{R}_1, \mathfrak{R}_2 \in \mathbf{NFA}(\Sigma, \ell)$, and for each $s \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}_1 \mathfrak{R}_2}^{[W]}(s) \leftrightarrow (\text{rec}_{\mathfrak{R}_1}^{[W]} \cdot \text{rec}_{\mathfrak{R}_2}^{[W]})(s).$$

Proof. (1) We first consider the case of depth-first recognizability. For any $q_0, q_1, \dots, q_m \in Q_1 \cup Q_2$, $\sigma_1, \dots, \sigma_m \in \Sigma \cup \{\varepsilon\}$ with $\sigma_1 \dots \sigma_m = s$ (note that it is possible that $|s| < m$ since $\sigma_1, \dots, \sigma_m$ may contain ε 's), if

$$I_1(q_0) \wedge T_2(q_m) \wedge \bigwedge_{i=1}^m \delta(q_{i-1}, \sigma_i, q_i) > 0,$$

then there exists $j \leq m$ such that $\sigma_j = \varepsilon$, $\sigma_i \neq \varepsilon$ ($i \neq j$), $q_0, \dots, q_{j-1} \in Q_1$,

$q_j, \dots, q_m \in Q_2$. Thus, $s = \sigma_1 \dots \sigma_{j-1} \sigma_{j+1} \dots \sigma_n$, and

$$\begin{aligned}
I_1(q_0) \wedge T_2(q_m) \wedge \bigwedge_{i=1}^m \delta(q_{i-1}, \sigma_i, q_i) &= I_1(q_0) \wedge T_2(q_m) \wedge \bigwedge_{i=1}^{j-1} \delta_1(q_{i-1}, \sigma_i, q_i) \\
&\quad \wedge T_1(q_{j-1}) \wedge I_2(q_j) \wedge \bigwedge_{i=j+1}^m \delta_2(q_{i-1}, \sigma_i, q_i) \\
&= [I_1(q_0) \wedge T_1(q_{j-1}) \wedge \bigwedge_{i=1}^{j-1} \delta_1(q_{i-1}, \sigma_i, q_i)] \wedge [I_2(q_j) \wedge T_2(q_m) \wedge \bigwedge_{i=j+1}^m \delta_2(q_{i-1}, \sigma_i, q_i)] \\
&\leq \text{rec}_{\mathfrak{R}_1}^{[D]}(\sigma_1 \dots \sigma_{j-1}) \wedge \text{rec}_{\mathfrak{R}_2}^{[D]}(\sigma_{j+1} \dots \sigma_m) \\
&\leq \bigvee \{ \text{rec}_{\mathfrak{R}_1}^{[D]}(s_1) \wedge \text{rec}_{\mathfrak{R}_2}^{[D]}(s_2) : s_1 s_2 = s \} \\
&= \lceil (\text{rec}_{\mathfrak{R}_1}^{[D]} \cdot \text{rec}_{\mathfrak{R}_2}^{[D]})(s) \rceil.
\end{aligned}$$

For width-first recognizability, we note that

$$\lceil \text{rec}_{\mathfrak{R}_1 \mathfrak{R}_2}^{[W]}(s) \rceil = \bigvee_{t^{-\varepsilon} = s, q \in Q_2} [\delta_t(I_1)(q) \wedge T_2(q)].$$

So, it suffices to show that for each $t \in (\Sigma \cup \{\varepsilon\})^*$ with $t^{-\varepsilon} = s$, and for each $q \in Q_2$,

$$\delta_t(I_1)(q) \wedge T_2(q) \leq \bigvee_{s_1 s_2 = s} (\lceil \text{rec}_{\mathfrak{R}_1}^{[W]}(s_1) \rceil \wedge \lceil \text{rec}_{\mathfrak{R}_2}^{[W]}(s_2) \rceil).$$

From the definitions of δ and $\delta_t(\cdot)$ it may be observed that $\delta_t(I_1)(q) = 0$ whenever t contains zero or more than two occurrences of ε . Thus, we can assume that $t = t_1 \varepsilon t_2$ and $t_1, t_2 \in \Sigma^*$. For any $q \in Q_1$, it is obvious that $\delta_{t_1 \varepsilon}(I_1)(q) = 0$. If $q \in Q_2$, then

$$\begin{aligned}
\delta_{t_1 \varepsilon}(I_1)(q) &= \bigvee_{p \in Q_1} [\delta_{t_1}(I_1)(p) \wedge \delta(p, \varepsilon, q)] \\
&= \bigvee_{p \in Q_1} [\delta_{t_1}(I_1)(p) \wedge T_1(p) \wedge I_2(q)] \\
&\leq \bigvee_{p \in Q_1} (\delta_{t_1}(I_1)(p) \wedge T_1(p)) \wedge I_2(q) \\
&= \lceil \text{rec}_{\mathfrak{R}_1}^{[W]}(t_1) \rceil \wedge I_2(q).
\end{aligned}$$

Furthermore, for any $q \in Q_2$, we can show that

$$\delta_t(I_1)(q) = \delta_{t_2}(\delta_{t_1 \varepsilon}(I_1))(q) \leq \lceil \text{rec}_{\mathfrak{R}_1}^{[W]}(t_1) \rceil \wedge \delta_{t_2}(I_2)(q)$$

by induction on the length $|t_2|$ of t_2 . Since $t^{-\varepsilon} = s$, we have $t_1 t_2 = s$. Therefore,

$$\begin{aligned}
\delta_t(I_1)(q) \wedge T_2(q) &\leq [\text{rec}_{\mathfrak{R}_1}^{[W]}(t_1)] \wedge \delta_{t_2}(I_2)(q) \wedge T_2(q) \\
&\leq [\text{rec}_{\mathfrak{R}_1}^{[W]}(t_1)] \wedge \bigvee_{p \in Q_2} [\delta_{t_2}(I_2)(p) \wedge T_2(p)] \\
&= [\text{rec}_{\mathfrak{R}_1}^{[W]}(t_1)] \wedge [\text{rec}_{\mathfrak{R}_2}^{[W]}(t_2)] \\
&\leq \bigvee_{s_1 s_2 = s} ([\text{rec}_{\mathfrak{R}_1}^{[W]}(s_1)] \wedge [\text{rec}_{\mathfrak{R}_2}^{[W]}(s_2)]) \\
&= [(\text{rec}_{\mathfrak{R}_1}^{[W]} \cdot \text{rec}_{\mathfrak{R}_2}^{[W]})(s)].
\end{aligned}$$

(2) We only consider the depth-first case. First, we can use Lemmas 2.6 and 2.7 to derive that

$$\begin{aligned}
&[\gamma(\text{atom}(\mathfrak{R}_1) \cup \text{atom}(\mathfrak{R}_2)) \wedge (\text{rec}_{\mathfrak{R}_1}^{[D]} \cdot \text{rec}_{\mathfrak{R}_2}^{[D]})(s)] \\
&= [\gamma(\text{atom}(\mathfrak{R}_1) \cup \text{atom}(\mathfrak{R}_2)) \wedge (\exists s_1, s_2 \in \Sigma^*)(s_1 s_2 = s \wedge \text{rec}_{\mathfrak{R}_1}^{[D]}(s_1) \wedge \text{rec}_{\mathfrak{R}_2}^{[D]}(s_2))] \\
&= [\gamma(\text{atom}(\mathfrak{R}_1) \cup \text{atom}(\mathfrak{R}_2))] \wedge \bigvee_{s_1 s_2 = s} (\text{rec}_{\mathfrak{R}_1}^{[D]}(s_1) \wedge \text{rec}_{\mathfrak{R}_2}^{[D]}(s_2)) \\
&\leq \bigvee_{s_1 s_2 = s} ([\gamma(\text{atom}(\mathfrak{R}_1) \cup \text{atom}(\mathfrak{R}_2))] \wedge \text{rec}_{\mathfrak{R}_1}^{[D]}(s_1) \wedge \text{rec}_{\mathfrak{R}_2}^{[D]}(s_2)).
\end{aligned}$$

For any $s_1, s_2 \in \Sigma^*$ with $s_1 s_2 = s$, we use Lemmas 2.6 and 2.7 again, and this yields:

$$\begin{aligned}
&[\gamma(\text{atom}(\mathfrak{R}_1) \cup \text{atom}(\mathfrak{R}_2))] \wedge \text{rec}_{\mathfrak{R}_1}^{[D]}(s_1) \wedge \text{rec}_{\mathfrak{R}_2}^{[D]}(s_2) = [\gamma(\text{atom}(\mathfrak{R}_1) \cup \text{atom}(\mathfrak{R}_2))] \wedge \\
&\quad \bigvee_{lb(c_1)=s_1} (I_1(b(c_1)) \wedge T_1(e(c_1)) \wedge [\text{Path}_{\mathfrak{R}_1}(s_1)]) \wedge \bigvee_{lb(c_2)=s_2} (I_2(b(c_2)) \wedge T_2(e(c_2)) \wedge [\text{Path}_{\mathfrak{R}_2}(s_2)]) \\
&\leq \bigvee_{lb(c_1)=s_1, lb(c_2)=s_2} (I_1(b(c_1)) \wedge T_1(e(c_1)) \wedge [\text{Path}_{\mathfrak{R}_1}(s_1)] \wedge I_2(b(c_2)) \wedge T_2(e(c_2)) \wedge [\text{Path}_{\mathfrak{R}_2}(s_2)]).
\end{aligned}$$

Furthermore, for any $c_1 = p_0 \sigma_1 p_1 \dots p_{m-1} \sigma_m p_m$ and $c_2 = q_0 \tau_1 q_1 \dots q_{n-1} \tau_n q_n$ with $s_1 = \sigma_1 \dots \sigma_m$ and $s_2 = \tau_1 \dots \tau_n$, we have:

$$\begin{aligned}
&I_1(b(c_1)) \wedge T_1(e(c_1)) \wedge [\text{Path}_{\mathfrak{R}_1}(s_1)] \wedge I_2(b(c_2)) \wedge T_2(e(c_2)) \wedge [\text{Path}_{\mathfrak{R}_2}(s_2)] = \\
&\quad I_1(p_0) \wedge T_2(q_m) \wedge \bigwedge_{i=1}^m \delta_1(p_{i-1}, \sigma_i, p_i) \wedge T_1(p_m) \wedge I_2(q_0) \wedge \bigwedge_{j=1}^n \delta_2(q_{j-1}, \tau_j, q_j) \\
&= I_1(p_0) \wedge T_2(q_m) \wedge [\text{Path}_{\mathfrak{R}_1 \mathfrak{R}_2}(p_0 \sigma_1 p_1 \dots p_{m-1} \sigma_m p_m \varepsilon q_0 \tau_1 q_1 \dots q_{n-1} \tau_n q_n)] \\
&\leq \text{rec}_{\mathfrak{R}_1 \mathfrak{R}_2}^{[D]}(s).
\end{aligned}$$

(3) The part that (3.1) implies (3.2) and (3.3) is a simple corollary of (2). Conversely, to prove the implication from (3.2) to (3.1), it suffices to show that ℓ enjoys distributivity, that is, for any $a, b, c \in L$, $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$. Let

$\mathfrak{R}_1 = \langle \{p_0, p_1\}, \delta_1, \{p_0\}, \{p_1\} \rangle$ and $\mathfrak{R}_2 = \langle \{q_0, q_1, q_2\}, \{q_0\}, \delta_2, \{q_1, q_2\} \rangle$, where $\sigma \in \Sigma$, $\delta_1(p_0, \sigma, p_1) = a$, $\delta_2(q_0, \sigma, q_1) = b$, $\delta_2(q_0, \sigma, q_2) = c$, and δ_1, δ_2 take value 0 for other arguments (see Figure 7). Then it follows that

$$\begin{aligned} a \wedge (b \vee c) &= [(\exists s_1, s_2 \in \Sigma^*)(s_1 s_2 = \sigma \sigma \wedge \text{rec}_{\mathfrak{R}_1}^{[D]}(s_1) \wedge \text{rec}_{\mathfrak{R}_2}^{[D]}(s_2))] \\ &= [\text{rec}_{\mathfrak{R}_1 \mathfrak{R}_2}^{[D]}(\sigma \sigma)] \\ &= (a \wedge b) \vee (a \wedge c). \end{aligned}$$

The above automata show that (3.3) implies (3.1) too. \square

Corollary 3.25. Let $\rightarrow = \rightarrow_3$. Then for any $A, B \in L^{\Sigma^*}$, it holds that

$$\models^\ell \text{ConCReg}_\Sigma^{[D]}(A, B) \rightarrow \text{CReg}_\Sigma^{[D]}(A \cdot B).$$

Proof. Similar to Corollary 3.21. \square

We now turn to consider the Kleene closure of an orthomodular lattice-valued language. For this purpose, we need to introduce the fold construction of an orthomodular lattice-valued automaton. Let $\mathfrak{R} = \langle Q, \delta, I, T \rangle \in \mathbf{NFA}(\Sigma, \ell)$ be an ℓ -valued automaton, and let $q_0 \notin Q$ be a new state. We define the fold of \mathfrak{R} to be ℓ -valued automaton $\mathfrak{R}^* = \langle Q \cup \{q_0\}, \delta^*, \{q_0\}, T \cup \{q_0\} \rangle$ with ε -moves, where

$$\delta^* : (Q \cup \{q_0\}) \times (\Sigma \cup \{\varepsilon\}) \times (Q \cup \{q_0\}) \rightarrow L$$

is given by

$$\delta^*(p, \sigma, q) = \begin{cases} I(q), & \text{if } p = q_0 \text{ and } \sigma = \varepsilon, \\ \delta(p, \sigma, q), & \text{if } p, q \in Q \text{ and } \sigma \neq \varepsilon, \\ T(p) \wedge I(q), & \text{if } p, q \in Q \text{ and } \sigma = \varepsilon, \\ 0, & \text{otherwise.} \end{cases}$$

The language accepted by the fold of an orthomodular lattice-valued automaton is then clearly presented by the following proposition.

Proposition 3.26. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let \rightarrow enjoy the Birkhoff-von Neumann requirement.

(1) For any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and for all $s \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}^*}^{[D]}(s) \rightarrow (\text{rec}_{\mathfrak{R}}^{[D]})^{*[D]}(s) \text{ and } \models^\ell \text{rec}_{\mathfrak{R}^*}^{[W]}(s) \rightarrow (\text{rec}_{\mathfrak{R}}^{[W]})^{*[W]}(s).$$

(2) For any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and for each $s \in \Sigma^*$,

$$\models^\ell \gamma(\text{atom}(\mathfrak{R})) \wedge (\text{rec}_{\mathfrak{R}}^{[D]})^{*[D]}(s) \rightarrow \text{rec}_{\mathfrak{R}^*}^{[D]}(s),$$

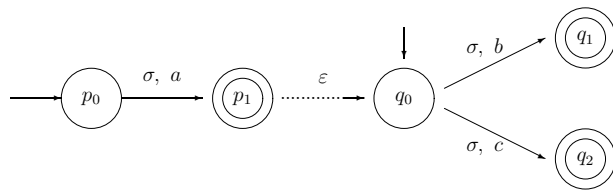


Figure 7: Automaton e

$$\models^\ell \gamma(\text{atom}(\mathfrak{R})) \wedge (\text{rec}_{\mathfrak{R}^*}^{[W]})^{*[W]}(s) \rightarrow \text{rec}_{\mathfrak{R}^*}^{[W]}(s),$$

and in particular if $\rightarrow = \rightarrow_3$, then

$$\begin{aligned} \models^\ell \gamma(\text{atom}(\mathfrak{R})) &\rightarrow (\text{rec}_{\mathfrak{R}^*}^{[D]}(s) \leftrightarrow (\text{rec}_{\mathfrak{R}}^{[D]})^{*[D]}(s)), \\ \models^\ell \gamma(\text{atom}(\mathfrak{R})) &\rightarrow (\text{rec}_{\mathfrak{R}^*}^{[W]}(s) \leftrightarrow (\text{rec}_{\mathfrak{R}}^{[W]})^{*[W]}(s)). \end{aligned}$$

(3) The following three statements are equivalent:

(3.1) ℓ is a Boolean algebra;

(3.2) for all $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and $s \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}^*}^{[D]}(s) \leftrightarrow (\text{rec}_{\mathfrak{R}}^{[D]})^{*[D]}(s);$$

(3.3) for all $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and $s \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}^*}^{[W]}(s) \leftrightarrow (\text{rec}_{\mathfrak{R}}^{[W]})^{*[W]}(s).$$

Proof. We only prove (1) for width-first recognizability and that each of (3.2) and (3.3) implies (3.1). For the other part, the proof is similar to that of Proposition 3.24, and here we omit the details.

To show that $[\text{rec}_{\mathfrak{R}^*}^{[W]}(s)] \leq [(\text{rec}_{\mathfrak{R}}^{[W]})^{*[W]}(s)]$, we observe that

$$[\text{rec}_{\mathfrak{R}^*}^{[W]}(s)] = \bigvee_{t^{-\varepsilon}=s, q \in Q \cup \{q_0\}} [\delta_t^*(q_0)(q) \wedge (T \cup \{q_0\})(q)].$$

Consequently, it suffices to demonstrate that for each $t \in (\Sigma \cup \{\varepsilon\})^*$ with $t^{-\varepsilon} = s$, and for each $q \in Q \cup \{q_0\}$, there exists $n \geq 0$ such that

$$\delta_t^*(q_0)(q) \wedge (T \cup \{q_0\})(q) \leq (\text{rec}_{\mathfrak{R}}^{[W]})^n(s).$$

It is clear for the case of $|t| = 0$ or 1. Note that $\delta_t^*(q_0)(q_0) = 0$ if $|t| \geq 1$. Thus we can assume that $q \in Q$. In this case, $(T \cup \{q_0\})(q) = T(q)$, and we only need to show that

$$\delta_t^*(q_0)(q) \wedge T(q) \leq (\text{rec}_{\mathfrak{R}}^{[W]})^n(s)$$

for some $n \geq 0$. If ε is not the first symbol of t , it is easy to see that $\delta_t^*(q_0)(q) = 0$. Thus, it suffices to consider the case of $t = \varepsilon u$ for some $u \in (\Sigma \cup \{\varepsilon\})^*$. If $u \in \Sigma^*$, then $u = s$, and from $\delta_\varepsilon^*(q_0) = I$ we know that

$$\delta_t^*(q_0)(q) \wedge T(q) \leq \delta_s(I)(q) \wedge T(q) \leq \text{rec}_{\mathfrak{R}}^{[W]}(s).$$

If u contains ε , we suppose that $u = u_1\varepsilon u'$ and u_1 does not contain ε . Then for any $p \in Q$ we have:

$$\begin{aligned}
\delta_\varepsilon^*(\delta_{\varepsilon u_1}^*(q_0))(p) &= \bigvee_{p' \in Q \cup \{q_0\}} [\delta_{\varepsilon u_1}^*(q_0)(p') \wedge \delta^*(p', \varepsilon, p)] \\
&= \bigvee_{p' \in Q} [\delta_{u_1}^*(I)(p') \wedge T(p') \wedge I(p)] \\
&\leq \bigvee_{p' \in Q} [\delta_{u_1}^*(I)(p') \wedge T(p')] \wedge I(p) \\
&= \text{rec}_{\mathfrak{R}}^{[W]}(u_1) \wedge I(p).
\end{aligned}$$

Since $\delta_t^*(q_0)(q) = \delta_{u_2}^*(\delta_\varepsilon^*(\delta_{\varepsilon u_1}^*(q_0)))(q)$, we can repeat this procedure for u' . The above inequality will be used in this repetition. Note that $|u_2| < |t|$. Such a repetition should stop in n steps for some $n < \infty$, and we finally obtain $u_1, u_2, \dots, u_n \in \Sigma^*$ such that $s = u_1 u_2 \dots u_n$ and

$$\delta_t^*(q_0)(q) \wedge T(q) \leq (\text{rec}_{\mathfrak{R}}^{[W]})^n(u_1 u_2 \dots u_n) = (\text{rec}_{\mathfrak{R}}^{[W]})^n(s).$$

To show that (3.2) implies (3.1), we assume that $a, b, c \in L$ and want to construct an ℓ -valued automaton for which the validity of (3.2) leads to $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$. Let $\mathfrak{R} = \langle \{q_1, q_2, \dots, q_6\}, \delta, \{q_1, q_2, q_3\}, \{q_6\} \rangle$ in which $\delta(q_1, \sigma, q_4) = \delta(q_3, \sigma, q_5) = 1$, $\delta(q_2, \sigma, q_6) = a$, $\delta(q_4, \sigma, q_6) = b$, $\delta(q_5, \sigma, q_6) = c$, and δ takes value 0 for the other arguments. Then \mathfrak{R}^* is visualized as Figure 8.

We now have

$$\begin{aligned}
a \wedge (b \vee c) &= [(\exists m \geq 0, s_1, \dots, s_m \in \Sigma^*)(s_1 \dots s_m = \sigma^3 \wedge \bigwedge_{i=1}^m \text{rec}_{\mathfrak{R}}^{[D]}(s_i))] \\
&= \text{rec}_{\mathfrak{R}^*}^{[D]}(\sigma^3) \\
&= (a \wedge b) \vee (a \wedge c).
\end{aligned}$$

A similar construction shows that (3.3) implies (3.1). \square

From the above proposition, we are able to demonstrate that the predicates $CReg_\Sigma^{[D]}$ and $CReg_\Sigma^{[W]}$ are preserved by the Kleene closure. The corresponding result for the predicate $Reg_\Sigma^{[D]}$ or $Reg_\Sigma^{[W]}$ is not true in general.

Corollary 3.27. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let $\rightarrow = \rightarrow_3$. Then for any $A \in L^{\Sigma^*}$, we have:

$$\models^\ell CReg_\Sigma^{[D]}(A) \rightarrow CReg_\Sigma^{[D]}(A^{*[D]}) \text{ and } \models^\ell CReg_\Sigma^{[W]}(A) \rightarrow CReg_\Sigma^{[W]}(A^{*[W]}).$$

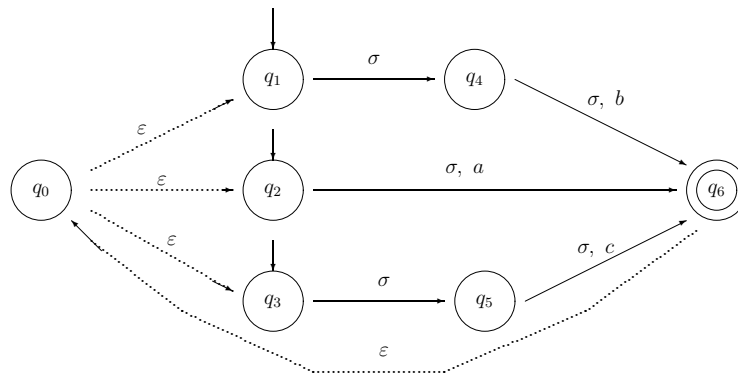


Figure 8: Automaton f

Proof. We only prove the first conclusion, and the second is left for the reader. The proof is similar to that of Proposition 3.17. The point here is to show the following inequality:

$$\lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil \wedge \lceil A \equiv \text{rec}_{\mathfrak{R}}^{[D]} \rceil \leq \lceil A^* \equiv \text{rec}_{\mathfrak{R}^*}^{[D]} \rceil$$

for any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$. In fact, by using Lemma 2.12(1) we have:

$$\begin{aligned} \lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil \wedge \lceil A \equiv \text{rec}_{\mathfrak{R}}^{[D]} \rceil &= \lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil \wedge \bigwedge_{s \in \Sigma^*} (A(s) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s)) \\ &\leq \lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil \wedge \bigwedge_{s \in \Sigma^*} \left(\bigvee_{s_1 \dots s_m = s} \bigwedge_{i=1}^m A(s_i) \leftrightarrow \bigvee_{s_1 \dots s_m = s} \bigwedge_{i=1}^m \text{rec}_{\mathfrak{R}}^{[D]}(s_i) \right) \\ &= \lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil \wedge \lceil A^* \equiv (\text{rec}_{\mathfrak{R}}^{[D]})^{*[D]} \rceil. \end{aligned}$$

On the other hand, it follows from Proposition 3.25 that

$$\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \leq \lceil (\text{rec}_{\mathfrak{R}}^{[D]})^{*[D]} \equiv \text{rec}_{\mathfrak{R}^*}^{[D]} \rceil.$$

Then with Lemma 2.12(3) we obtain:

$$\begin{aligned} \lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil \wedge \lceil A \equiv \text{rec}_{\mathfrak{R}}^{[D]} \rceil &\leq \lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil \wedge \lceil A^* \equiv (\text{rec}_{\mathfrak{R}}^{[D]})^{*[D]} \rceil \\ &\quad \wedge \lceil (\text{rec}_{\mathfrak{R}}^{[D]})^{*[D]} \equiv \text{rec}_{\mathfrak{R}^*}^{[D]} \rceil \\ &\leq \lceil A^* \equiv \text{rec}_{\mathfrak{R}^*}^{[D]} \rceil. \quad \square \end{aligned}$$

To conclude this section, we show that both the predicates $\text{Reg}_{\Sigma}^{[D]}$, $\text{Reg}_{\Sigma}^{[W]}$, $C\text{Reg}_{\Sigma}^{[D]}$ and $C\text{Reg}_{\Sigma}^{[W]}$ are preserved by the pre-image of a homomorphism between two languages. But the closure property of an orthomodular lattice-valued regular language under homomorphism is postponed to be examined in the next section, after the notion of orthomodular lattice-valued regular expression is proposed.

Let $\mathfrak{R} = \langle Q, \delta, I, T \rangle \in \mathbf{NFA}(\Gamma, \ell)$ be an ℓ -valued automaton over Γ . Then the pre-image of \mathfrak{R} under h is defined to be an ℓ -valued automaton $h^{-1}(\mathfrak{R}) = \langle Q, h^{-1}(\delta), I, T \rangle \in \mathbf{NFA}(\Sigma, \ell)$ over Σ , where for any $p, q \in Q$ and $\sigma \in \Sigma$,

$$h^{-1}(\delta)(p, \sigma, q) = \delta(p, h(\sigma), q).$$

The pre-image of a homomorphism has a very nice compatibility with the predicates $\text{Reg}_{\Sigma}^{[D]}$, $\text{Reg}_{\Sigma}^{[W]}$, $C\text{Reg}_{\Sigma}^{[D]}$ and $C\text{Reg}_{\Sigma}^{[W]}$, and no commutativity is needed here.

Proposition 3.28. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, let \rightarrow enjoy the Birkhoff-von Neumann requirement, and let $h : \Sigma \rightarrow \Gamma^*$ be a mapping. Then for any $\mathfrak{R} \in \mathbf{NFA}(\Gamma, \ell)$ and for any $s \in \Sigma^*$, it holds that

$$\models^{\ell} \text{rec}_{h^{-1}(\mathfrak{R})}^{[D]}(s) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[D]}(h(s)) \quad \text{and} \quad \models^{\ell} \text{rec}_{h^{-1}(\mathfrak{R})}^{[W]}(s) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[W]}(h(s)).$$

Proof. We only consider depth-first recognizability, and the case of width-first recognizability is similar. Suppose that $s = \sigma_1\sigma_2\dots\sigma_n$. Then

$$\begin{aligned}
[\text{rec}_{h^{-1}(\mathfrak{R})}^{[D]}(s)] &= \bigvee \{ I(q_0) \wedge T(q_n) \wedge \bigwedge_{i=0}^{n-1} h^{-1}(\delta)(q_i, \sigma_{i+1}, q_{i+1}) : q_0, q_1, \dots, q_n \in Q \} \\
&= \bigvee \{ I(q_0) \wedge T(q_n) \wedge \bigwedge_{i=0}^{n-1} \delta(q_i, h(\sigma_{i+1}), q_{i+1}) : q_0, q_1, \dots, q_n \in Q \} \\
&= [\text{rec}_{\mathfrak{R}}^{[D]}(h(\sigma_1)h(\sigma_2)\dots h(\sigma_n))] \\
&= [\text{rec}_{\mathfrak{R}}^{[D]}(h(s))]. \quad \square
\end{aligned}$$

Corollary 3.29. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, let \rightarrow enjoy the Birkhoff-von Neumann requirement, and let $h : \Sigma \rightarrow \Gamma^*$ be a mapping. Then for any $B \in L^{\Gamma^*}$,

$$\begin{aligned}
&\models^\ell \text{Reg}_\Gamma^{[D]}(B) \rightarrow \text{Reg}_\Sigma^{[D]}(h^{-1}(B)) \text{ and } \models^\ell \text{Reg}_\Gamma^{[W]}(B) \rightarrow \text{Reg}_\Sigma^{[W]}(h^{-1}(B)), \\
&\models^\ell \text{CReg}_\Gamma^{[D]}(B) \rightarrow \text{CReg}_\Sigma^{[D]}(h^{-1}(B)) \text{ and } \models^\ell \text{CReg}_\Gamma^{[W]}(B) \rightarrow \text{CReg}_\Sigma^{[W]}(h^{-1}(B)).
\end{aligned}$$

Proof. From the above proposition we have

$$h^{-1}(\text{rec}_{\mathfrak{R}}^{[D]}(s)) = \text{rec}_{\mathfrak{R}}^{[D]}(h(s)) = \text{rec}_{h^{-1}(\mathfrak{R})}^{[D]}(s)$$

for all $s \in \Sigma^*$. Then with Lemma 2.13 we obtain

$$\begin{aligned}
[\text{Reg}_\Gamma^{[D]}(B)] &= \bigvee \{ [B \equiv \text{rec}_{\mathfrak{R}}^{[D]}] : \mathfrak{R} \in \mathbf{NFA}(\Gamma, \ell) \} \\
&\leq \bigvee \{ [h^{-1}(B) \equiv h^{-1}(\text{rec}_{\mathfrak{R}}^{[D]})] : \mathfrak{R} \in \mathbf{NFA}(\Gamma, \ell) \} \\
&= \bigvee \{ [h^{-1}(B) \equiv \text{rec}_{h^{-1}(\mathfrak{R})}^{[D]}] : \mathfrak{R} \in \mathbf{NFA}(\Gamma, \ell) \} \\
&\leq \bigvee \{ [h^{-1}(B) \equiv \text{rec}_{\wp}^{[D]}] : \wp \in \mathbf{NFA}(\Sigma, \ell) \} \\
&= [\text{Reg}_\Sigma^{[D]}(h^{-1}(B))].
\end{aligned}$$

It is similar for the other cases. \square

3.5. The Kleene Theorem for Orthomodular Lattice-Valued Languages

One of the most interesting results in classical automata theory is the Kleene theorem which shows the equivalence between finite automata and regular expressions. The main aim of this section is to present an orthomodular lattice-valued generalization of the Kleene theorem.

Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let Σ be a nonempty set of input symbols. Then the language of ℓ -valued regular expressions over Σ has the alphabet $(\Sigma \cup \{\varepsilon, \phi\}) \cup (L \cup \{+, \cdot, *\})$. The symbols in $\Sigma \cup \{\varepsilon, \phi\}$ will be used to stand for atomic expressions, and the symbols in $L \cup \{+, \cdot, *\}$ will be used to denote operators for building up compound expressions: $*$ and all $\lambda \in L$ are unary operators, and $+, \cdot$ are binary ones. We use α, β, \dots to act as meta-symbols for regular expressions. For any expression α , $L^{[D]}(\alpha)$ and $L^{[W]}(\alpha)$ denote the languages generated by α in the depth-first way and the width-first way, respectively. Thus, both $L^{[D]}(\alpha)$ and $L^{[W]}(\alpha)$ will be used to denote an ℓ -valued subset of Σ^* ; that is, $L^{[D]}(\alpha), L^{[W]}(\alpha) \in L^{\Sigma^*}$. Orthomodular lattice-valued regular expressions and the orthomodular lattice-valued languages denoted by them are formally defined as follows:

(i) For each $a \in \Sigma$, a is a regular expression, and $L^{[D]}(a) = L^{[W]}(a) = \{a\}$; ε and ϕ are regular expressions, and $L^{[D]}(\varepsilon) = L^{[W]}(\varepsilon) = \{\varepsilon\}$, $L^{[D]}(\phi) = L^{[W]}(\phi) = \emptyset$.

(ii) If both α and β are regular expressions, then for each $\lambda \in L$, $\lambda\alpha$ is a regular expression, and

$$L^{[D]}(\lambda\alpha) = \lambda L^{[D]}(\alpha), \quad L^{[W]}(\lambda\alpha) = \lambda L^{[W]}(\alpha),$$

and $\alpha + \beta$, $\alpha \cdot \beta$ and α^* are all regular expressions, and

$$L^{[D]}(\alpha + \beta) = L^{[D]}(\alpha) \cup L^{[D]}(\beta), \quad L^{[W]}(\alpha + \beta) = L^{[W]}(\alpha) \cup L^{[W]}(\beta),$$

$$L^{[D]}(\alpha \cdot \beta) = L^{[D]}(\alpha) \cdot L^{[D]}(\beta), \quad L^{[W]}(\alpha \cdot \beta) = L^{[W]}(\alpha) \cdot L^{[W]}(\beta),$$

$$L^{[D]}(\alpha^*) = L(\alpha)^{*[D]}, \quad L^{[W]}(\alpha^*) = L(\alpha)^{*[W]}.$$

It is easy to see that the only difference between orthomodular lattice-valued regular expressions and the classical ones is that the additional unary (scalar) operators $\lambda \in L$ are permitted to occur in the former.

The central part of the Kleene theorem is a mechanism to transform a finite automaton into a regular expression. This mechanism has a straightforward extension in the framework of orthomodular lattice-valued automata. Let $\mathfrak{R} = \langle Q, \delta, I, T \rangle \in \mathbf{NFA}(\Sigma, \ell)$ be an ℓ -valued automaton over Σ . For any $u, v \in Q$ and $X \subseteq Q$, α_{uv}^X is defined by induction on the cardinality $|X|$ of X :

(1)

$$\alpha_{uv}^\emptyset = \begin{cases} \sum_{\sigma \in \Sigma} \delta(u, \sigma, v)\sigma, & \text{if } u \neq v, \\ \varepsilon + \sum_{\sigma \in \Sigma} \delta(u, \sigma, v)\sigma, & \text{if } u = v. \end{cases}$$

(2) if $X \neq \emptyset$, then we choose any $q \in X$ and define

$$\alpha_{uv}^X = \alpha_{uv}^{X-\{q\}} + \alpha_{uq}^{X-\{q\}} \cdot (\alpha_{qq}^{X-\{q\}})^* \cdot \alpha_{qv}^{X-\{q\}}.$$

Then the ℓ -valued regular expression

$$k(\mathfrak{R}) = \sum_{u, v \in Q} (I(u) \wedge T(v)) \alpha_{uv}^Q$$

is called a Kleene representation of \mathfrak{R} .

The following theorem describes properly the relationship between the language recognized by an orthomodular lattice-valued automaton and the language expressed by its Kleene representation.

Theorem 3.30. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let \rightarrow satisfy the Birkhoff-von Neumann requirement.

(1) For any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and $s \in \Sigma^*$, if $k(\mathfrak{R})$ is a Kleene representation of \mathfrak{R} , then

$$\models^\ell \text{rec}_{\mathfrak{R}}^{[D]}(s) \rightarrow s \in L^{[D]}(k(\mathfrak{R})) \text{ and } \models^\ell \text{rec}_{\mathfrak{R}}^{[W]}(s) \rightarrow s \in L^{[W]}(k(\mathfrak{R}))$$

(2) For any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and $s \in \Sigma^*$, and for any Kleene representation $k(\mathfrak{R})$ of \mathfrak{R} , we have:

$$\begin{aligned} \models^\ell \gamma(\text{atom}(\mathfrak{R})) \wedge s \in L^{[D]}(k(\mathfrak{R})) &\rightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s), \\ \models^\ell \gamma(\text{atom}(\mathfrak{R})) \wedge s \in L^{[W]}(k(\mathfrak{R})) &\rightarrow \text{rec}_{\mathfrak{R}}^{[W]}(s), \end{aligned}$$

and especially if $\rightarrow = \rightarrow_3$, then

$$\begin{aligned} \models^\ell \gamma(\text{atom}(\mathfrak{R})) &\rightarrow (\text{rec}_{\mathfrak{R}}^{[D]}(s) \leftrightarrow s \in L^{[D]}(k(\mathfrak{R}))), \\ \models^\ell \gamma(\text{atom}(\mathfrak{R})) &\rightarrow (\text{rec}_{\mathfrak{R}}^{[W]}(s) \leftrightarrow s \in L^{[W]}(k(\mathfrak{R}))). \end{aligned}$$

(3) The following three statements are equivalent:

(3.1) ℓ is a Boolean algebra.

(3.2) for any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and $s \in \Sigma^*$, and for any Kleene representation $k(\mathfrak{R})$ of \mathfrak{R} ,

$$\models^\ell \text{rec}_{\mathfrak{R}}^{[D]}(s) \leftrightarrow s \in L^{[D]}(k(\mathfrak{R})).$$

(3.3) for any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$ and $s \in \Sigma^*$, and for any Kleene representation $k(\mathfrak{R})$ of \mathfrak{R} ,

$$\models^\ell \text{rec}_{\mathfrak{R}}^{[W]}(s) \leftrightarrow s \in L^{[W]}(k(\mathfrak{R})).$$

Proof. We only consider the depth-first case, and the width-first case is left for the reader. We prove (1) and (2) together. To this end, we have to demonstrate that for any $u, v \in Q$, $X \subseteq Q$ and $s \in \Sigma^*$,

$$(a) \bigvee \{ [\text{Path}_{\mathfrak{R}}(c)] : c \in T(Q, \Sigma), b(c) = u, e(c) = v, M(c) \subseteq X, lb(c) = s \} \leq L^{[D]}(\alpha_{uv}^X)(s),$$

$$(b) [\gamma(\text{atom}(\mathfrak{R}))] \wedge L^{[D]}(\alpha_{uv}^X)(s) \leq$$

$$\bigvee \{ [\text{Path}_{\mathfrak{R}}(c)] : c \in T(Q, \Sigma), b(c) = u, e(c) = v, M(c) \subseteq X, lb(c) = s \},$$

where $M(c)$ stands for the set of states along c except u and v ; more exactly, $M(c) = \{q_1, \dots, q_{k-1}\}$ if $c = u\sigma_1q_1\dots q_{k-1}\sigma_k v$. This claim may be proved by induction on $|X|$. For the case of $X = \phi$, it is easy. We now suppose that $q \in X \neq \emptyset$ and

$$\alpha_{uv}^X = \alpha_{uv}^{X-\{q\}} + [\alpha_{uq}^{X-\{q\}}(\alpha_{qq}^{X-\{q\}})^*]\alpha_{qv}^{X-\{q\}}.$$

We first show that (a) is valid in this case. From the induction hypothesis we have

$$(c) \bigvee \{[Path_{\mathfrak{R}}(c)] : c \in T(Q, \Sigma), b(c) = e(c) = q,$$

$$M(c) \subseteq X - \{q\}, lb(c) = s\} \leq L^{[D]}(\alpha_{qq}^{X-\{q\}})(s)$$

for each $s \in \Sigma^*$. Then we assert that for all $s \in \Sigma^*$,

$$(d) \bigvee \{[Path_{\mathfrak{R}}(c)] : c \in T(Q, \Sigma), b(c) = e(c) = q, M(c) \subseteq X, lb(c) = s\}$$

$$\leq L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)(s).$$

In fact, for any $c \in T(Q, \Sigma)$, if $b(c) = e(c) = q$, $M(c) \subseteq X$ and $lb(c) = s$, we write c_i for the substring of c beginning with the i th q and ending at the $(i+1)$ th q . If the number of occurrences of q in c is $m+1$, then

$$[Path_{\mathfrak{R}}(c)] = \bigwedge_{i=1}^m [Path_{\mathfrak{R}}(c_i)].$$

Furthermore, by using (c) and noting that $s = lb(c_1)\dots lb(c_m)$ we obtain:

$$\begin{aligned} [Path_{\mathfrak{R}}(c)] &= \bigwedge_{i=1}^m L^{[D]}(\alpha_{qq}^{X-\{q\}})(lb(c_i)) \\ &\leq \bigvee \left\{ \bigwedge_{i=1}^n L^{[D]}(\alpha_{qq}^{X-\{q\}})(s_i) : n \geq 0, s_1, \dots, s_n \in \Sigma^*, s = s_1\dots s_n \right\} \\ &= (L^{[D]}(\alpha_{qq}^{X-\{q\}}))^*(s) \\ &= L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)(s). \end{aligned}$$

Let c range over $\{c \in T(Q, \Sigma) : b(c) = e(c) = q, M(c) \subseteq X, lb(c) = s\}$. Then (d) is proved.

From the induction hypothesis and (d) we have:

$$\begin{aligned} &([L^{[D]}(\alpha_{uq}^{X-\{q\}})L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)]L^{[D]}(\alpha_{qv}^{X-\{q\}}))(s) = \\ &\quad \bigvee \{[L^{[D]}(\alpha_{uq}^{X-\{q\}})L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)](x) \wedge L^{[D]}(\alpha_{qv}^{X-\{q\}})(y) : s = xy\} \\ &= \bigvee \{ \bigvee \{L^{[D]}(\alpha_{uq}^{X-\{q\}})(x_1) \wedge L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)(x_2) : x = x_1x_2\} \wedge L^{[D]}(\alpha_{qv}^{X-\{q\}})(y) : s = xy \} \\ &\geq \bigvee \{L^{[D]}(\alpha_{uq}^{X-\{q\}})(x_1) \wedge L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)(x_2) \wedge L^{[D]}(\alpha_{qv}^{X-\{q\}})(y) : s = x_1x_2y\} \\ &\geq \bigvee \{[Path_{\mathfrak{R}}(c_1)] \wedge [Path_{\mathfrak{R}}(c_2)] \wedge [Path_{\mathfrak{R}}(c_3)] : c_1, c_2, c_3 \in T(Q, \Sigma), \\ &\quad b(c_1) = u, e(c_1) = b(c_2) = e(c_2) = b(c_3) = q, e(c_3) = v, s = lb(c_1)lb(c_2)lb(c_3)\} \\ &= \bigvee \{[Path_{\mathfrak{R}}(c)] : c \in T(Q, \Sigma), b(c) = u, e(c) = v, q \in M(c)\}. \end{aligned}$$

This yields further

$$\begin{aligned} L^{[D]}(\alpha_{uv}^X)(s) &= L^{[D]}(\alpha_{uv}^{X-\{q\}})(s) \vee ([L^{[D]}(\alpha_{uq}^{X-\{q\}})L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)]L^{[D]}(\alpha_{qv}^{X-\{q\}}))(s) \\ &\geq \text{the left - hand side of (a)}. \end{aligned}$$

We now turn to consider (b). The induction hypothesis gives

$$(e) \quad [\gamma(\text{atom}(\mathfrak{R}))] \wedge L^{[D]}(\alpha_{uv}^{X-\{q\}})(s) \leq \bigvee \{[\text{Path}_{\mathfrak{R}}(c)] : c \in T(Q, \Sigma), \\ b(c) = u, e(c) = v, M(c) \subseteq X - \{q\}, lb(c) = s\}.$$

For any $n \geq 0$ and $s_1, \dots, s_n \in \Sigma^*$ with $s = s_1 \dots s_n$, from (e) we can apply Lemmas 2.6 and 2.7 to obtain:

$$\begin{aligned} [\gamma(\text{atom}(\mathfrak{R}))] \wedge \bigwedge_{i=1}^n L^{[D]}(\alpha_{qq}^{X-\{q\}})(s_i) &= [\gamma(\text{atom}(\mathfrak{R}))] \wedge \bigwedge_{i=1}^n [[\gamma(\text{atom}(\mathfrak{R}))] \wedge L^{[D]}(\alpha_{qq}^{X-\{q\}})(s_i)] \\ &\leq [\gamma(\text{atom}(\mathfrak{R}))] \wedge \bigwedge_{i=1}^n \bigvee \{[\text{Path}_{\mathfrak{R}}(c_i)] : c_i \in T(Q, \Sigma), \\ &\quad b(c_i) = e(c_i) = q, M(c_i) \subseteq X - \{q\}, lb(c_i) = s_i\} \\ &\leq \bigvee \{ \bigwedge_{i=1}^n [\text{Path}_{\mathfrak{R}}(c_i)] : c_i \in T(Q, \Sigma), b(c_i) = e(c_i) = q, M(c_i) \subseteq X - \{q\}, \\ &\quad lb(c_i) = s_i \text{ for each } i = 1, 2, \dots, n \} \\ &\leq \bigvee \{[\text{Path}_{\mathfrak{R}}(\overline{c_1 \dots c_n})] : c_i \in T(Q, \Sigma), b(c_i) = e(c_i) = q, M(c_i) \subseteq X - \{q\}, \\ &\quad lb(c_i) = s_i \text{ for each } i = 1, 2, \dots, n \}, \end{aligned}$$

where $\overline{c_1 \dots c_n} = c_1 c'_2 \dots c'_n$, c'_i is the resulting string after removing the first q in c_i for each $i \geq 2$. Note that $lb(\overline{c_1 \dots c_n}) = s_1 \dots s_n = s$ whenever $lb(c_i) = s_i$ ($i = 1, 2, \dots, n$). We write

$$\lambda = \bigvee \{[\text{Path}_{\mathfrak{R}}(c)] : c \in T(Q, \Sigma), b(c) = e(c) = q, M(c) \subseteq X, lb(c) = s\}.$$

Then it holds that

$$[\gamma(\text{atom}(\mathfrak{R}))] \wedge \bigwedge_{i=1}^n L^{[D]}(\alpha_{qq}^{X-\{q\}})(s_i) \leq \lambda.$$

Moreover, note that $[\gamma(\text{atom}(\mathfrak{R}))], L^{[D]}(\alpha_{qq}^{X-\{q\}})(s_i) \in [\text{atom}(\mathfrak{R})]$. It follows that

$$\begin{aligned} [\gamma(\text{atom}(\mathfrak{R}))] \wedge L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)(s) &= [\gamma(\text{atom}(\mathfrak{R}))] \wedge [\gamma(\text{atom}(\mathfrak{R}))] \wedge \\ &\quad \bigvee \{ \bigwedge_{i=1}^n L^{[D]}(\alpha_{qq}^{X-\{q\}})(s_i) : n \geq 0, s = s_1 \dots s_n \} \\ &\leq \bigvee \{[\gamma(\text{atom}(\mathfrak{R}))] \wedge \bigwedge_{i=1}^n L^{[D]}(\alpha_{qq}^{X-\{q\}})(s_i) : n \geq 0, s = s_1 \dots s_n\} \leq \lambda. \end{aligned}$$

This enables us to obtain:

$$\begin{aligned}
& [\gamma(\text{atom}(\mathfrak{R}))] \wedge [L^{[D]}(\alpha_{uq}^{X-\{q\}})L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)](x) \\
&= [\gamma(\text{atom}(\mathfrak{R}))] \wedge [\gamma(\text{atom}(\mathfrak{R}))] \wedge \bigvee \{L^{[D]}(\alpha_{uq}^{X-\{q\}})(x_1) \wedge L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)(x_2) : x = x_1x_2\} \\
&\leq \bigvee \{[\gamma(\text{atom}(\mathfrak{R}))] \wedge L^{[D]}(\alpha_{uq}^{X-\{q\}})(x_1) \wedge L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)(x_2) : x = x_1x_2\} \\
&= \bigvee \{[\gamma(\text{atom}(\mathfrak{R}))] \wedge [[\gamma(\text{atom}(\mathfrak{R}))] \wedge L^{[D]}(\alpha_{uq}^{X-\{q\}})(x_1)] \wedge \\
&\quad [[\gamma(\text{atom}(\mathfrak{R}))] \wedge L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)(x_2)] : x = x_1x_2\} \\
&\leq \bigvee \{[\gamma(\text{atom}(\mathfrak{R}))] \wedge [\bigvee \{[Path_{\mathfrak{R}}(c_1)] : c_1 \in T(Q, \Sigma), b(c_1) = u, e(c_1) = q, \\
&\quad M(c_1) \subseteq X - \{q\}, lb(c_1) = x_1\}] \wedge [\bigvee \{[Path_{\mathfrak{R}}(c_2)] : c_2 \in T(Q, \Sigma), b(c_2) = \\
&\quad e(c_2) = q, M(c_2) \subseteq X, lb(c_2) = x_2\}] : x = x_1x_2\} \\
&\leq \bigvee \{[Path_{\mathfrak{R}}(c_1)] \wedge [Path_{\mathfrak{R}}(c_2)] : c_1, c_2 \in T(Q, \Sigma), b(c_1) = u, e(c_1) = b(c_2) = e(c_2) = q, \\
&\quad M(c_1) \subseteq X - \{q\}, M(c_2) \subseteq X, x = lb(c_1)lb(c_2)\}.
\end{aligned}$$

Furthermore, we can derive in a similar way that

$$\begin{aligned}
& [\gamma(\text{atom}(\mathfrak{R}))] \wedge ([L^{[D]}(\alpha_{uq}^{X-\{q\}})L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)]L^{[D]}(\alpha_{qv}^{X-\{q\}}))(s) \\
&\leq \bigvee \{[Path_{\mathfrak{R}}(c_1)] \wedge [Path_{\mathfrak{R}}(c_2)] \wedge [Path_{\mathfrak{R}}(c_3)] : c_1, c_2, c_3 \in T(Q, \Sigma), b(c_1) = u, \\
&\quad e(c_1) = b(c_2) = e(c_2) = b(c_3) = q, e(c_3) = v, s = lb(c_1)lb(c_2)lb(c_3)\} \\
&= \bigvee \{[Path_{\mathfrak{R}}(c)] : c \in T(Q, \Sigma), b(c) = u, e(c) = v, q \in M(c), s = lb(c)\}.
\end{aligned}$$

Consequently, it holds that

$$\begin{aligned}
& [\gamma(\text{atom}(\mathfrak{R}))] \wedge L^{[D]}(\alpha_{uv}^X)(s) = [\gamma(\text{atom}(\mathfrak{R}))] \wedge \{L^{[D]}(\alpha_{uv}^{X-\{q\}})(s) \vee \\
&\quad ([L^{[D]}(\alpha_{uq}^{X-\{q\}})L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)]L^{[D]}(\alpha_{qv}^{X-\{q\}}))(s)\} \\
&\leq [[\gamma(\text{atom}(\mathfrak{R}))] \wedge L^{[D]}(\alpha_{uv}^{X-\{q\}})(s)] \vee \{[\gamma(\text{atom}(\mathfrak{R}))] \wedge ([L^{[D]}(\alpha_{uq}^{X-\{q\}}) \\
&\quad L^{[D]}((\alpha_{qq}^{X-\{q\}})^*)]L^{[D]}(\alpha_{qv}^{X-\{q\}}))(s)\} \\
&\leq \text{the right - hand side of (b)}.
\end{aligned}$$

After proving (a), we can assert that

$$\begin{aligned}
[s \in L^{[D]}(k(\mathfrak{R}))] &= \bigvee_{u,v \in Q} [I(u) \wedge T(v) \wedge L^{[D]}(\alpha_{uv}^Q)(s)] \\
&\geq \bigvee_{u,v \in Q} [(I(u) \wedge T(v)) \wedge \bigvee \{[Path_{\mathfrak{R}}(c)] : c \in T(Q, \Sigma), b(c) = u, e(c) = v, lb(c) = s\}] \\
&\geq \bigvee_{u,v \in Q} \bigvee \{I(u) \wedge T(v) \wedge [Path_{\mathfrak{R}}(c)] : c \in T(Q, \Sigma), b(c) = u, e(c) = v, lb(c) = s\} \\
&= [rec_{\mathfrak{R}}^{[D]}(s)].
\end{aligned}$$

By using (b) and Lemmas 2.6 and 2.7, we have:

$$\begin{aligned}
& [\gamma(\text{atom}(\mathfrak{R}))] \wedge [s \in L^{[D]}(k(\mathfrak{R}))] = [\gamma(\text{atom}(\mathfrak{R}))] \wedge \bigvee_{u,v \in Q} [I(u) \wedge T(v) \wedge L^{[D]}(\alpha_{uv}^Q)(s)] \\
& \leq \bigvee_{u,v \in Q} [I(u) \wedge T(v) \wedge [\gamma(\text{atom}(\mathfrak{R}))] \wedge L^{[D]}(\alpha_{uv}^Q)(s)] \\
& \leq \bigvee_{u,v \in Q} [(I(u) \wedge T(v)) \wedge [\gamma(\text{atom}(\mathfrak{R}))] \wedge \bigvee\{[\text{Path}_{\mathfrak{R}}(c)] : c \in T(Q, \Sigma), \\
& \quad b(c) = u, e(c) = v, lb(c) = s\}] \\
& \leq \bigvee_{u,v \in Q} \bigvee\{I(u) \wedge T(v) \wedge [\text{Path}_{\mathfrak{R}}(c)] : c \in T(Q, \Sigma), b(c) = u, e(c) = v, lb(c) = s\} \\
& = [\text{rec}_{\mathfrak{R}}^{[D]}(s)].
\end{aligned}$$

Thus, (1) and (2) are proved, and the part that (3.1) implies (3.2) and (3.3) of (3) is a simple corollary of (2). We now turn to prove that (3.2) implies (3.1). For any $a, b, c \in L$, we consider the ℓ -valued automaton $\mathfrak{R} = \langle \{u, v\}, \delta, u_a, \{u, v\} \rangle$, where $\delta(u, \sigma, u) = b$, $\delta(u, \sigma, v) = c$, and δ takes value 0 for other cases (see Figure 9). Then

$$\begin{aligned}
[\text{rec}_{\mathfrak{R}}^{[D]}(\sigma)] &= \bigvee\{I(q_0) \wedge T(q_1) \wedge \delta(q_0, \sigma, q_1) : q_0, q_1 \in Q\} \\
&= (a \wedge b) \vee (a \wedge c).
\end{aligned}$$

On the other hand, we have

$$\begin{cases} \alpha_{uu}^{\emptyset} = \varepsilon + b\sigma, \\ \alpha_{uv}^{\emptyset} = c\sigma, \\ \alpha_{vv}^{\emptyset} = \varepsilon, \\ \alpha_{vu}^{\emptyset} = \phi. \end{cases}$$

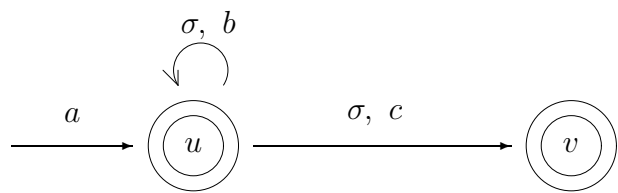
Therefore,

$$\begin{aligned}
\alpha_{uu}^{\{v\}} &= \alpha_{uu}^{\emptyset} + [\alpha_{uv}^{\emptyset}(\alpha_{vv}^{\emptyset})^*] \alpha_{vu}^{\emptyset} \\
&= (\varepsilon + b\sigma) + [c\sigma(\varepsilon)^*] \phi = \varepsilon + b\sigma,
\end{aligned}$$

$$\begin{aligned}
\alpha_{uv}^{\{v\}} &= \alpha_{uv}^{\emptyset} + [\alpha_{uv}^{\emptyset}(\alpha_{vv}^{\emptyset})^*] \alpha_{vv}^{\emptyset} \\
&= c\sigma + [c\sigma(\varepsilon)^*] \varepsilon = c\sigma,
\end{aligned}$$

and

$$\begin{aligned}
\alpha_{uv}^{\{u,v\}} &= \alpha_{uv}^{\{v\}} + [\alpha_{uu}^{\{v\}}(\alpha_u^{\{v\}})^*] \alpha_{uv}^{\{v\}} \\
&= \varepsilon + b\sigma + [(\varepsilon + b\sigma)(\varepsilon + b\sigma)^*] (c\sigma).
\end{aligned}$$



From the assumption (3.2) we know that

$$\begin{aligned}
(a \wedge b) \vee (a \wedge c) &= [\text{rec}_{\mathfrak{R}}^{[D]}(\sigma)] \\
&= L^{[D]}(k(\mathfrak{R}))(\sigma) \\
&= [L^{[D]}(a\alpha_u^{\{u,v\}}) \cup L^{[D]}(a\alpha_{uv}^{\{u,v\}})](\sigma) \\
&\geq L^{[D]}(a\alpha_u^{\{u,v\}})(\sigma) \\
&= a \wedge L^{[D]}(\alpha_u^{\{u,v\}})(\sigma) \\
&= a \wedge L^{[D]}(\varepsilon + b\sigma + [(\varepsilon + b\sigma)(\varepsilon + b\sigma)^*](c\sigma))(\sigma) \\
&\geq a \wedge (b \vee c).
\end{aligned}$$

The implication from (3.3) to (3.1) may be proved in a similar way. This completes the proof. \square

Corollary 3.31. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let $\rightarrow = \rightarrow_3$. Then for any $A \in L^{\Sigma^*}$, we have:

$$\begin{aligned}
\models^\ell CReg_\Sigma^{[D]}(A) &\rightarrow (\exists \text{ regular expression } \alpha)(A \equiv L^{[D]}(\alpha)), \\
\models^\ell CReg_\Sigma^{[W]}(A) &\rightarrow (\exists \text{ regular expression } \alpha)(A \equiv L^{[W]}(\alpha)).
\end{aligned}$$

Proof. It can be derived from Theorem 3.30 in a way similar to the proof of Corollary 3.14. \square

We now turn to consider homomorphisms of ℓ -valued regular expressions. Let Σ and Γ be two alphabet, and let $h : \Sigma \rightarrow \Gamma^*$ be a mapping. Then it can be uniquely extended to a mapping, denoted still by h , from ℓ -valued regular expressions over Σ into ℓ -valued regular expressions over Γ . For any ℓ -valued regular expression α over Σ , $h(\alpha)$ is defined to be the ℓ -valued regular expression over Γ obtained by replacing each letter $\sigma \in \Sigma$ appearing in α with the string $h(\sigma) \in \Gamma^*$. Formally, $h(\alpha)$ is defined by induction on the length of α :

$$\left\{ \begin{array}{l}
h(\varepsilon) = \varepsilon, \\
h(\phi) = \phi, \\
h(\sigma) \text{ is already given for each } \sigma \in \Sigma, \\
h(\lambda\alpha) = \lambda h(\alpha), \\
h(\alpha_1 + \alpha_2) = h(\alpha_1) + h(\alpha_2), \\
h(\alpha_1 \cdot \alpha_2) = h(\alpha_1) \cdot h(\alpha_2), \\
h(\alpha^*) = (h(\alpha))^*.
\end{array} \right.$$

For each ℓ -valued regular expression α over Σ , we write $\Lambda(\alpha)$ for the set of scalar values $\lambda \in L$ occurring in α . Indeed, $\Lambda(\alpha)$ may be formally defined by induction on the length of α as follows:

$$\begin{cases} \Lambda(\varepsilon) = \Lambda(\phi) = \Lambda(\sigma) = \emptyset \text{ for every } \sigma \in \Sigma, \\ \Lambda(\lambda\alpha) = \{\lambda\} \cup \Lambda(\alpha), \\ \Lambda(\alpha_1 + \alpha_2) = \Lambda(\alpha_1 \cdot \alpha_2) = \Lambda(\alpha_1) \cup \Lambda(\alpha_2), \\ \Lambda(\alpha^*) = \Lambda(\alpha). \end{cases}$$

It is easy to see that $\Lambda(\alpha)$ is a finite subset of L . Moreover, we write $\Delta(\alpha) = \{\mathbf{a} : a \in \Lambda(\alpha)\}$ for the set of (constant) propositions in our logical language corresponding to the elements in $\Lambda(\alpha)$.

The following two lemmas are very useful when we are dealing with orthomodular lattice-valued expressions, and they evaluate the range of language generated by an orthomodular lattice-valued regular expression. In particular, it will be shown in Lemma 3.33 that this range is a finite set whenever the lattice ℓ of truth values is a Boolean algebra.

Lemma 3.32. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice. Then for any ℓ -valued regular expression α , we have $\{L^{[D]}(\alpha)(s) : s \in \Sigma^*\} \subseteq [A(\alpha)]$, where $[A]$ denotes the subalgebra of ℓ generated by A for any $A \subseteq L$. The same conclusion holds for $L^{[W]}(\cdot)$.

Proof. We only prove this lemma for $L^{[D]}(\cdot)$, the case of $L^{[W]}(\cdot)$ is similar. We use an induction argument on the length of α . For simplicity, we only consider the following two cases, and the other cases are easy or similar.

(1) From the induction hypothesis we know that

$$L^{[D]}(\lambda.\alpha)(s) = \lambda \wedge L^{[D]}(\alpha)(s) \in [\{\lambda\} \cup \Lambda(\alpha)] = [\Lambda(\lambda.\alpha)]$$

for each $s \in \Sigma^*$.

(2) Let $s \in \Sigma^*$. For any $s_1, \dots, s_n \in \Sigma^*$ with $s_1 \dots s_n = s$, we suppose that $s_{i_1}, \dots, s_{i_m} \neq \varepsilon$ and $s_i = \varepsilon$ for every $i \in \{1, \dots, n\} - \{i_1, \dots, i_m\}$. Then $s_{i_1} \dots s_{i_m} = s$ and

$$L^{[D]}(\alpha)(s_1) \wedge \dots \wedge L^{[D]}(\alpha)(s_n) = \begin{cases} L^{[D]}(\alpha)(s_{i_1}) \wedge \dots \wedge L^{[D]}(\alpha)(s_{i_m}), & \text{if } m = n, \\ L^{[D]}(\alpha)(s_{i_1}) \wedge \dots \wedge L^{[D]}(\alpha)(s_{i_m}) \wedge L^{[D]}(\alpha)(\varepsilon), & \text{if } m < n. \end{cases}$$

Furthermore, we note that

$$\{(s_1, \dots, s_n) : n \geq 0, s_1, \dots, s_n \in \Sigma^* - \{\varepsilon\} \text{ and } s_1 \dots s_n = s\}$$

is finite. Therefore,

$$\{L^{[D]}(\alpha)(s_1) \wedge \dots \wedge L^{[D]}(\alpha)(s_n) : s_1 \dots s_n = s\}$$

is also finite, and with the induction hypothesis we have

$$L^{[D]}(\alpha^*)(s) = \bigvee \{L^{[D]}(\alpha)(s_1) \wedge \dots \wedge L^{[D]}(\alpha)(s_n) : s_1 \dots s_n = s\} \in [\Lambda(\alpha)]. \quad \square$$

Lemma 3.33. If $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ is a Boolean algebra, then for any ℓ -valued regular expression α , $\{L^{[D]}(\alpha)(s) : s \in \Sigma^*\}$ is a finite set.

Proof. From Lemma 3.32 and the distributivity of \wedge over \vee we know that for any $s \in \Sigma^*$, there are $\lambda_{ij_i} \in \Lambda(\alpha)$ ($i = 1, \dots, m; j_i = 1, \dots, n_i$) such that

$$L^{[D]}(\alpha)(s) = \bigvee_{i=1}^m \left(\bigwedge_{j_i=1}^{n_i} \lambda_{ij_i} \right).$$

Since $\Lambda(\alpha)$ is finite, both

$$\Lambda(\alpha)^{(\wedge)} = \{\lambda_1 \wedge \dots \wedge \lambda_n : n \geq 0, \lambda_1, \dots, \lambda_n \in \Lambda(\alpha)\}$$

and

$$\Lambda(\alpha)^{(\wedge)(\vee)} = \{\bigvee M : M \subseteq \Lambda(\alpha)^{(\wedge)}\}$$

are also finite. Therefore,

$$\Lambda(\alpha)^{(\wedge)(\vee)} \supseteq \{L^{[D]}(\alpha)(s) : s \in \Sigma^*\}$$

is a finite set. \square

Note that the above lemma is also true for $L^{[W]}(\cdot)$ because $L^{[W]}(\cdot) = L^{[D]}(\cdot)$ when ℓ is a Boolean algebra.

The following proposition shows that a homomorphism preserves the language generated by an orthomodular lattice-valued regular expression under the condition that all elements in the range of the expression under consideration are commutative.

Proposition 3.34. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice and \rightarrow fulfil the Birkhoff-von Neumann requirement, and let Σ and Γ be two alphabets.

(1) For any mapping $h : \Sigma \rightarrow \Gamma^*$, and for any ℓ -valued regular expression α over Σ ,

$$\models^\ell h(L^{[D]}(\alpha)) \subseteq L^{[D]}(h(\alpha)) \text{ and } \models^\ell h(L^{[W]}(\alpha)) \subseteq L^{[W]}(h(\alpha)).$$

(2) For any mapping $h : \Sigma \rightarrow \Gamma^*$, for any ℓ -valued regular expression α over Σ , and for any $t \in \Gamma^*$,

$$\models^\ell \gamma(\Delta(\alpha)) \wedge t \in L^{[D]}(h(\alpha)) \rightarrow t \in h(L^{[D]}(\alpha)),$$

$$\models^\ell \gamma(\Delta(\alpha)) \wedge t \in L^{[W]}(h(\alpha)) \rightarrow t \in h(L^{[W]}(\alpha)).$$

and if $\rightarrow = \rightarrow_3$ then

$$\models^\ell \gamma(\Delta(\alpha)) \rightarrow L^{[D]}(h(\alpha)) \equiv h(L^{[D]}(\alpha)) \text{ and } \models^\ell \gamma(\Delta(\alpha)) \rightarrow L^{[W]}(h(\alpha)) \equiv h(L^{[W]}(\alpha)).$$

(3) The following three statements are equivalent:

(3.1) ℓ is a Boolean algebra.

(3.2) for any mapping $h : \Sigma \rightarrow \Gamma^*$, and for any ℓ -valued regular expression α over Σ ,

$$\models^\ell h(L^{[D]}(\alpha)) \equiv L^{[D]}(h(\alpha)).$$

(3.3) for any mapping $h : \Sigma \rightarrow \Gamma^*$, and for any ℓ -valued regular expression α over Σ ,

$$\models^\ell h(L^{[W]}(\alpha)) \equiv L^{[W]}(h(\alpha)).$$

Proof. We only consider $L^{[D]}(\cdot)$ and prove (2) and (3). (1) can be observed from the proof of (2). The part that (3.1) implies (3.2) and (3.3) may be derived from (2); and it can also be proved directly by using Lemma 3.33.

Our first aim is to prove that

$$[\gamma(\Delta(\alpha))] \wedge L^{[D]}(h(\alpha))(t) \leq h(L^{[D]}(\alpha))(t)$$

for any $t \in \Gamma^*$ and for any ℓ -valued regular expression α over Σ . We proceed by induction on the length of α .

(a) It is obvious for the case of $\alpha = \varepsilon$ or ϕ , or $\alpha \in \Sigma$.

(b) With the definitions of $h(\cdot)$ and $L^{[D]}(\cdot)$ and the induction hypothesis we derive that

$$\begin{aligned} L^{[D]}(h(\lambda.\alpha))(t) &= L^{[D]}(\lambda.h(\alpha))(t) \\ &= \lambda \wedge L^{[D]}(h(\alpha))(t) \\ &= \lambda \wedge h(L^{[D]}(\alpha))(t) \\ &= \lambda \wedge \bigvee \{L^{[D]}(\alpha)(s) : s \in \Sigma^* \text{ and } h(s) = t\}. \end{aligned}$$

Then from Lemmas 2.6, 2.7 and 3.33, it follows that

$$\begin{aligned} [\gamma(\Delta(\alpha))] \wedge L^{[D]}(h(\lambda.\alpha))(t) &\leq \bigvee \{\lambda \wedge L^{[D]}(\alpha)(s) : s \in \Sigma^* \text{ and } h(s) = t\} \\ &= \bigvee \{L^{[D]}(\lambda.\alpha)(s) : s \in \Sigma^* \text{ and } h(s) = t\} \\ &= h(L^{[D]}(\lambda.\alpha))(t). \end{aligned}$$

(c) It is easy to observe that $h(A \cup B) = h(A) \cup h(B)$ for all $A, B \in L^{\Sigma^*}$. This

together with the induction hypothesis as well as Lemmas 2.6 and 2.7 yields:

$$\begin{aligned}
& \lceil \gamma(\Delta(\alpha_1 + \alpha_2)) \rceil \wedge L^{[D]}(h(\alpha_1 + \alpha_2))(t) = \lceil \gamma(\Delta(\alpha_1 + \alpha_2)) \rceil \wedge L^{[D]}(h(\alpha_1) + h(\alpha_2))(t) \\
& = \lceil \gamma(\Delta(\alpha_1 + \alpha_2)) \rceil \wedge \lceil \gamma(\Delta(\alpha_1 + \alpha_2)) \rceil \wedge [L^{[D]}(h(\alpha_1))(t) \vee L^{[D]}(h(\alpha_2))(t)] \\
& \leq [\lceil \gamma(\Delta(\alpha_1 + \alpha_2)) \rceil \wedge L^{[D]}(h(\alpha_1))(t)] \wedge [\lceil \gamma(\Delta(\alpha_1 + \alpha_2)) \rceil \wedge L^{[D]}(h(\alpha_2))(t)] \\
& \leq [\lceil \gamma(\Delta(\alpha_1)) \rceil \wedge L^{[D]}(h(\alpha_1))(t)] \wedge [\lceil \gamma(\Delta(\alpha_2)) \rceil \wedge L^{[D]}(h(\alpha_2))(t)] \\
& \leq h(L^{[D]}(\alpha_1))(t) \vee h(L^{[D]}(\alpha_2))(t) \\
& = h(L^{[D]}(\alpha_1) \cup L^{[D]}(\alpha_2))(t) \\
& = h(L^{[D]}(\alpha_1 + \alpha_2))(t).
\end{aligned}$$

(d) For any $t \in \Gamma^*$, Lemmas 2.6, 2.7 and 3.33 enable us to assert that

$$\begin{aligned}
& \lceil \gamma(\Delta(\alpha_1 \cdot \alpha_2)) \rceil \wedge L^{[D]}(h(\alpha_1 \cdot \alpha_2))(t) = \lceil \gamma(\Delta(\alpha_1 \cdot \alpha_2)) \rceil \wedge L^{[D]}(h(\alpha_1) \cdot h(\alpha_2))(t) \\
& = \lceil \gamma(\Delta(\alpha_1 \cdot \alpha_2)) \rceil \wedge L^{[D]}(h(\alpha_1))L^{[D]}(h(\alpha_2))(t) \\
& = \lceil \gamma(\Delta(\alpha_1 \cdot \alpha_2)) \rceil \wedge \bigvee \{L^{[D]}(h(\alpha_1))(t_1) \wedge L^{[D]}(h(\alpha_2))(t_2) : t_1 t_2 = t\} \\
& \leq \bigvee \{\lceil \gamma(\Delta(\alpha_1 \cdot \alpha_2)) \rceil \wedge L^{[D]}(h(\alpha_1))(t_1) \wedge L^{[D]}(h(\alpha_2))(t_2) : t_1 t_2 = t\} \\
& = \bigvee \{\lceil \gamma(\Delta(\alpha_1 \cdot \alpha_2)) \rceil \wedge (\lceil \gamma(\Delta(\alpha_1)) \rceil \wedge L^{[D]}(h(\alpha_1))(t_1)) \wedge (\lceil \gamma(\Delta(\alpha_2)) \rceil \\
& \quad \wedge L^{[D]}(h(\alpha_2))(t_2)) : t_1 t_2 = t\} \\
& \leq \bigvee \{\lceil \gamma(\Delta(\alpha_1 \cdot \alpha_2)) \rceil \wedge h(L^{[D]}(\alpha_1))(t_1) \wedge h(L^{[D]}(\alpha_2))(t_2) : t_1 t_2 = t\}.
\end{aligned}$$

Furthermore, by using Lemmas 2.6, 2.7 and 3.33 again we obtain:

$$\begin{aligned}
& \lceil \gamma(\Delta(\alpha_1 \cdot \alpha_2)) \rceil \wedge h(L^{[D]}(\alpha_1))(t_1) \wedge h(L^{[D]}(\alpha_2))(t_2) = \lceil \gamma(\Delta(\alpha_1 \cdot \alpha_2)) \rceil \wedge \\
& \quad (\bigvee \{L^{[D]}(\alpha_1)(s_1) : h(s_1) = t_1\}) \wedge (\bigvee \{L^{[D]}(\alpha_2)(s_2) : h(s_2) = t_2\}) \\
& \leq \bigvee \{L^{[D]}(\alpha_1)(s_1) \wedge L^{[D]}(\alpha_2)(s_2) : h(s_1) = t_1 \text{ and } h(s_2) = t_2\}.
\end{aligned}$$

Therefore, it follows that

$$\begin{aligned}
& \lceil \gamma(\Delta(\alpha_1 \cdot \alpha_2)) \rceil \wedge L^{[D]}(h(\alpha_1 \cdot \alpha_2))(t) \leq \bigvee \{L^{[D]}(\alpha_1)(s_1) \wedge \\
& \quad L^{[D]}(\alpha_2)(s_2) : h(s_1) = t_1, h(s_2) = t_2 \text{ and } t_1 t_2 = t\} \\
& = \bigvee \{L^{[D]}(\alpha_1)(s_1) \wedge L^{[D]}(\alpha_2)(s_2) : h(s_1 s_2) = t\} \\
& = h(L^{[D]}(\alpha_1)L^{[D]}(\alpha_2))(t) = h(L^{[D]}(\alpha_1 \alpha_2))(t).
\end{aligned}$$

(e) For every $t \in \Gamma^*$, Lemmas 2.6, 2.7 and 3.33 guarantee that

$$\begin{aligned}
& [\gamma(\Delta(\alpha^*))] \wedge L^{[D]}(h(\alpha^*))(t) = [\gamma(\Delta(\alpha^*))] \wedge L^{[D]}((h(\alpha))^*)(t) \\
& = [\gamma(\Delta(\alpha^*))] \wedge (L^{[D]}(h(\alpha)))^*(t) \\
& = [\gamma(\Delta(\alpha^*))] \wedge \bigvee \left\{ \bigwedge_{i=1}^n L^{[D]}(h(\alpha))(t_i) : n \geq 0, t_1, \dots, t_n \in \Gamma^*, t_1 \dots t_n = t \right\} \\
& \leq \bigvee \left\{ [\gamma(\Delta(\alpha^*))] \wedge \bigwedge_{i=1}^n L^{[D]}(h(\alpha))(t_i) : n \geq 0, t_1, \dots, t_n \in \Gamma^*, t_1 \dots t_n = t \right\} \\
& = \bigvee \left\{ [\gamma(\Delta(\alpha))] \wedge \bigwedge_{i=1}^n ([\gamma(\Delta(\alpha))] \wedge L^{[D]}(h(\alpha))(t_i)) : n \geq 0, t_1, \dots, t_n \in \Gamma^*, t_1 \dots t_n = t \right\} \\
& \leq \bigvee \left\{ [\gamma(\Delta(\alpha))] \wedge \bigwedge_{i=1}^n h(L^{[D]}(\alpha))(t_i) : n \geq 0, t_1, \dots, t_n \in \Gamma^*, t_1 \dots t_n = t \right\}.
\end{aligned}$$

On the other hand, we have:

$$\begin{aligned}
[\gamma(\Delta(\alpha))] \wedge \bigwedge_{i=1}^n h(L^{[D]}(\alpha))(t_i) & = [\gamma(\Delta(\alpha))] \wedge \bigwedge_{i=1}^n \left(\bigvee \{ L^{[D]}(\alpha)(s_i) : h(s_i) = t_i \} \right) \\
& \leq \bigvee \left\{ \bigwedge_{i=1}^n L^{[D]}(\alpha)(s_i) : h(s_i) = t_i \ (i = 1, \dots, n) \right\}.
\end{aligned}$$

This further yields:

$$\begin{aligned}
[\gamma(\Delta(\alpha^*))] \wedge L^{[D]}(h(\alpha^*))(t) & \leq \bigvee \left\{ \bigwedge_{i=1}^n L^{[D]}(\alpha)(s_i) : n \geq 0, h(s_i) = t_i \ (i = 1, \dots, n) \text{ and } t = t_1 \dots t_n \right\} \\
& = \bigvee \left\{ \bigwedge_{i=1}^n L^{[D]}(\alpha)(s_i) : n \geq 0, h(s_1 \dots s_n) = t \right\} \\
& = \bigvee \left\{ L^{[D]}(\alpha)^*(s) : h(s) = t \right\} \\
& = h((L^{[D]}(\alpha))^*)(t) = h(L^{[D]}(\alpha^*))(t).
\end{aligned}$$

What remains is to prove that (3.2) or (3.3) implies (3.1). We only consider (3.2), and the other case is left to the reader. This needs indeed to show that the distributivity of \wedge over \vee is derivable from the statement (3.2). Suppose that $a, b, c \in L$. We choose an symbol $\sigma \in \Sigma$ and an symbol $\gamma \in \Gamma$, and define $h(\sigma) = \varepsilon$ and $h(\sigma') = \gamma$ for every $\sigma' \in \Sigma - \{\sigma\}$. We further set $\alpha_1 = a.\sigma$ and $\alpha_2 = b.\varepsilon + c.\sigma$. Then

$$L^{[D]}(\alpha_1.\alpha_2)(\sigma) = \begin{cases} a \wedge b, & \text{if } n = 1, \\ a \wedge c, & \text{if } n = 2, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\begin{aligned} h(L^{[D]}(\alpha_1.\alpha_2))(\varepsilon) &= \bigvee_{n=0}^{\infty} L^{[D]}(\alpha_1.\alpha_2)(\sigma^n) \\ &= (a \wedge b) \vee (a \wedge c). \end{aligned}$$

On the other hand, we have:

$$\begin{aligned} L^{[D]}(h(\alpha_1.\alpha_2))(\varepsilon) &= L^{[D]}((a.\varepsilon).(b.\varepsilon + c.\varepsilon))(\varepsilon) \\ &= L^{[D]}(a.\varepsilon)(\varepsilon) \wedge L^{[D]}(b.\varepsilon + c.\varepsilon)(\varepsilon) = a \wedge (b \vee c). \end{aligned}$$

From (3.2) we know that $h(L(\alpha_1.\alpha_2))(\varepsilon) = L(h(\alpha_1.\alpha_2))(\varepsilon)$. This indicates that $(a \wedge b) \vee (a \wedge c) = a \wedge (b \vee c)$. \square

3.6. The Myhill-Nerode Theorem for Orthomodular Lattice-Valued Languages

The Myhill-Nerode theorem in classical automata theory gives another nice characterization of regular languages in terms of binary relations over strings of input symbols. In this section, we are going to establish a generalization of this theorem in the framework of quantum logic.

Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, let Σ be a finite alphabet, and let A be an ℓ -valued language over Σ . For any ℓ -valued binary relation \approx over Σ^* , namely, a mapping \approx from $\Sigma^* \times \Sigma^*$ into L , we first introduce the following derived logical formulas:

$$\begin{aligned} Ref(\approx) &\stackrel{def}{=} (\forall x \in \Sigma^*)(x \approx x); \\ Sym(\approx) &\stackrel{def}{=} (\forall x, y \in \Sigma^*)(x \approx y \rightarrow y \approx x); \\ Tran(\approx) &\stackrel{def}{=} (\forall x, y, z \in \Sigma^*)(x \approx y \wedge y \approx z \rightarrow x \approx z); \\ RCon(\approx) &\stackrel{def}{=} (\forall x, y \in \Sigma^*)(\forall \sigma \in \Sigma)(x \approx y \rightarrow x\sigma \approx y\sigma); \\ Refin(\approx, A) &\stackrel{def}{=} (\forall x, y \in \Sigma^*)[x \approx y \rightarrow (x \in A \leftrightarrow y \in A)]; \\ FInd(\approx) &\stackrel{def}{=} (\exists n \in \omega)(\exists x_1, \dots, x_n \in \Sigma^*)(\forall x \in \Sigma^*)(\exists i \leq n)(x \approx x_i), \end{aligned}$$

where ω is the set of nonnegative integers. Intuitively, $Ref(\approx)$, $Sym(\approx)$ and $Tran(\approx)$ mean that \approx is reflexive, symmetric and transitive, respectively; $RCon(\approx)$ means that \approx is a right congruence; $Refin(\approx, A)$ indicates that \approx refines A ; and $FInd(\approx)$ expresses that \approx is of finite index. It should be noted that the defining formula of predicate $FInd(\cdot)$ is essentially not a formula in the ordinary first-order language but a formula in infinitary logic.

Recall that for any set X we use $p_\ell(X)$ to denote the set of ℓ -valued points in X . Moreover, for every $e \in p_\ell(X)$, $s(e)$ and $h(e)$ stand respectively for the support

and height of e . Then an ℓ -valued DFA over Σ is a quadruple $\mathfrak{R} = \langle Q, \delta, e_0, T \rangle$ where Q is a finite set of states, $e_0 \in p_\ell(Q)$, T is an ℓ -valued subset of Q , and $\delta : Q \times \Sigma \rightarrow p_\ell(Q)$ is a mapping. The transition function δ may be extended to $\delta : p_\ell(Q) \times \Sigma \rightarrow p_\ell(Q)$ in a natural way:

$$\delta(q_\lambda, \sigma) = s(\delta(q, \sigma))_{\lambda \wedge h(\delta(q, \sigma))}$$

for any $q \in Q$, $\lambda \in L - \{0\}$ and $\sigma \in \Sigma$. Furthermore, it can be extended to $\delta : p_\ell(Q) \times \Sigma^* \rightarrow p_\ell(Q)$ by induction on the length of input string s as follows: $\delta(e, s\sigma) = \delta(\delta(e, s), \sigma)$ for any $e \in p_\ell(Q)$, $s \in \Sigma^*$ and $\sigma \in \Sigma$.

We suppose that \mathfrak{R} is an ℓ -valued DFA. Then it induces an ℓ -valued binary relation $\approx_{\mathfrak{R}}$ on Σ^* in the following way:

$$x \approx_{\mathfrak{R}} y \stackrel{def}{=} \delta(e_0, x) \approx \delta(e_0, y)$$

for all $x, y \in \Sigma^*$. It is easy to see that the truth value of statement $x \approx_{\mathfrak{R}} y$ is given by

$$[x \approx_{\mathfrak{R}} y] = \begin{cases} [h(\delta(e_0, x)) \rightarrow 0] \wedge [h(\delta(e_0, y)) \rightarrow 0], & \text{if } s(\delta(e_0, x)) \neq s(\delta(e_0, y)) \\ h(\delta(e_0, x)) \rightarrow h(\delta(e_0, y)), & \text{otherwise.} \end{cases}$$

We now want to present some basic properties of relation $\approx_{\mathfrak{R}}$. To this end, we need to introduce a weakened notion of finiteness for orthomodular lattices.

Definition 3.35. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let $a \in L$. If for any subset K of L , there exists a finite subset H of K such that for every $\lambda \in K$, we have:

$$\bigvee_{\mu \in H} (\lambda \leftrightarrow \mu) \geq a$$

then ℓ is said to be a -finite.

Intuitively, the above inequality means that for each element λ of K we can find an element μ of H such that the nearness degree between them, measured by $\lambda \leftrightarrow \mu$, is greater than or equal to a pre-specified threshold value a . It is clear that ℓ is 1-finite whenever L is finite.

Lemma 3.36. For any $\mathfrak{R} \in \mathbf{DFA}(\Sigma, \ell)$, we have:

- (1) $\models^\ell \text{Ref}(\approx_{\mathfrak{R}})$;
- (2) $\models^\ell \text{Sym}(\approx_{\mathfrak{R}})$; and
- (3) if $a \in L$ and ℓ is a -finite, then $\models^\ell \mathbf{a} \rightarrow \text{FInd}(\approx_{\mathfrak{R}})$, where \mathbf{a} is the nullary predicate associated with a .

In particular, if $\rightarrow = \rightarrow_3$ then we have:

- (4) $\models^\ell \gamma(\text{atom}(\mathfrak{R})) \rightarrow \text{Tran}(\approx_{\mathfrak{R}})$;
(5) $\models^\ell \gamma(\text{atom}(\mathfrak{R})) \rightarrow \text{RCon}(\approx_{\mathfrak{R}})$; and
(6) $\models^\ell \gamma(\text{atom}(\mathfrak{R})) \rightarrow \text{Refin}(\approx_{\mathfrak{R}}, \text{rec}_{\mathfrak{R}})$.

Proof. (1) and (2) are obvious.

(3) We first define (two-valued) binary relation \sim on Σ^* as follows:

$$x \sim y \text{ if and only if } s(\delta(e_0, x)) = s(\delta(e_0, y))$$

for any $x, y \in \Sigma^*$. It is obvious that \sim is an equivalence relation. Since Q is finite, the quotient $\Sigma^*/\sim = \{[x] : x \in \Sigma^*\}$ should be a finite set, where $[x]$ stands for the equivalence class of x with respect to \sim for each $x \in \Sigma^*$. Assume that $\Sigma^*/\sim = \{[x_1], \dots, [x_m]\}$. For every $i \leq m$, we set $K_i = \{h(\delta(e_0, x)) : x \in [x_i]\}$. Since ℓ is a -finite, there must be a finite subset H_i of K_i such that for any $\lambda \in K_i$,

$$\bigvee_{\mu \in H_i} (\lambda \leftrightarrow \mu) \geq a.$$

Now for each $\mu \in H_i$, there exists $x_{i\mu} \in [x_i]$ such that $h(\delta(e_0, x_{i\mu})) = \mu$.

Let $E_i = \{x_{i\mu} : \mu \in H_i\}$ for all $1 \leq i \leq m$, and let $E = \bigcup_{i=1}^m E_i$. Then E is a finite set, and for any $x \in \Sigma^*$, there is $i \leq m$ such that $x \in [x_i]$. This implies that $h(\delta(e_0, x)) \in K_i$, and

$$\begin{aligned} \bigvee_{y \in E} [x \approx_{\mathfrak{R}} y] &\geq \bigvee_{y \in E_i} [x \approx_{\mathfrak{R}} y] \\ &= \bigvee_{y \in E_i} (h(\delta(e_0, x)) \leftrightarrow h(\delta(e_0, y))) \\ &= \bigvee_{y \in H_i} (h(\delta(e_0, x)) \leftrightarrow \mu) \geq a. \end{aligned}$$

Therefore, we have:

$$\begin{aligned} [FInd(\approx_{\mathfrak{R}})] &= \bigvee_{n \in \omega, x_1, \dots, x_n \in \Sigma^*} \bigwedge_{x \in \Sigma^*} \bigvee_{i \leq n} [x \approx_{\mathfrak{R}} x_i] \\ &\geq \bigwedge_{x \in \Sigma^*} \bigvee_{y \in E} [x \approx_{\mathfrak{R}} y] \geq a. \end{aligned}$$

(4) We write $s_u = s(\delta(e_0, u))$ for $u \in \{x, y, z\}$ and consider the following cases:

Case 1. $s_x = s_y$ and $s_y = s_z$. It follows from Lemmas 2.7 and 2.12(3), (4).

Case 2. $s_x \neq s_y$, $s_y \neq s_z$ and $s_x = s_z$. It suffices to note that

$$\begin{aligned} [x \approx_{\mathfrak{R}} y \wedge y \approx_{\mathfrak{R}} z] &= (h_x \rightarrow 0) \wedge (h_y \rightarrow 0) \wedge (h_z \rightarrow 0) \\ &\leq (h_x \rightarrow h_z) \wedge (h_z \rightarrow h_x) = [x \approx_{\mathfrak{R}} z] \end{aligned}$$

where $h_u = h(\delta(e_0, u))$ for $u \in \{x, y, z\}$.

(5) Assume that $\delta(e_0, x) = p_\lambda$, $\delta(e_0, y) = q_\mu$, $\delta(p, \sigma) = p'_\lambda$, and $\delta(q, \sigma) = q'_\mu$. We consider the following three cases:

Case 1. $p = q$. Then $p' = q'$, $\lambda' = \mu'$, and by using Lemma 2.12(1) we obtain:

$$\begin{aligned} [\gamma(\text{atom}(\mathfrak{R}))] \wedge [x \approx_{\mathfrak{R}} y] &= [\gamma(\text{atom}(\mathfrak{R}))] \wedge (\lambda \leftrightarrow \mu) \\ &\leq \lambda \wedge \lambda' \leftrightarrow \mu \wedge \lambda' = [x\sigma \approx_{\mathfrak{R}} y\sigma]. \end{aligned}$$

Case 2. $p \neq q$ and $p' = q'$. Note that $\lambda \rightarrow 0 \leq \lambda \wedge \lambda' \rightarrow 0 \leq \lambda \wedge \lambda' \rightarrow \mu \wedge \mu'$. Likewise, we have $\mu \rightarrow 0 \leq \mu \wedge \mu' \rightarrow \lambda \wedge \lambda'$. Therefore,

$$[x \approx_{\mathfrak{R}} y] = (\lambda \rightarrow 0) \wedge (\mu \rightarrow 0) \lambda \wedge \lambda' \rightarrow \mu \wedge \mu' = [x\sigma \approx_{\mathfrak{R}} y\sigma].$$

Case 3. $p \neq q$ and $p' \neq q'$. Similar to Case 2.

(6) We also assume that $\delta(e_0, x) = p_\lambda$ and $\delta(e_0, y) = q_\lambda$. Then it is easy to see that $\text{rec}_{\mathfrak{R}}(x) = T(p) \wedge \lambda$ and $\text{rec}_{\mathfrak{R}}(y) = T(q) \wedge \lambda$. If $p = q$, then it follows from Lemma 2.12(1) that

$$\begin{aligned} [\gamma(\text{atom}(\mathfrak{R}))] \wedge [x \approx_{\mathfrak{R}} y] &= [\gamma(\text{atom}(\mathfrak{R}))] \wedge (\lambda \leftrightarrow \mu) \\ &\leq T(p) \wedge \lambda \leftrightarrow T(p) \wedge \mu \\ &= [\text{rec}_{\mathfrak{R}}(x) \leftrightarrow \text{rec}_{\mathfrak{R}}(y)]. \end{aligned}$$

If $p \neq q$, then we can derive $[x \approx_{\mathfrak{R}} y] \leq [\text{rec}_{\mathfrak{R}}(x) \leftrightarrow \text{rec}_{\mathfrak{R}}(y)]$ in a way similar to Case 2 in the proof of (4). \square

The above lemma shows that the relation $\approx_{\mathfrak{R}}$ induced by an ℓ -valued DFA satisfies the six logical formulas given at the beginning of this section with respect to the ℓ -valued language $\text{rec}_{\mathfrak{R}}$ accepted by \mathfrak{R} (if some extra conditions are fulfilled). This suggests us to introduce the Myhill-Nerode property by putting these formulas, each of which stands for a certain constraint on binary relations over input strings, together.

Definition 3.37. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let Σ be a finite alphabet.

(1) For any ℓ -valued language A over Σ , ℓ -valued unary predicate $MNR(\cdot, A)$ over ℓ -valued binary relations on Σ^* is defined as follows:

$$\begin{aligned} MNR(\approx, A) &\stackrel{\text{def}}{=} \text{Ref}(\approx) \wedge \text{Sym}(\approx) \wedge \text{Tran}(\approx) \\ &\quad \wedge \text{RCon}(\approx) \wedge \text{Refin}(\approx, A) \wedge \text{FInd}(\approx) \end{aligned}$$

for each ℓ -valued binary relation \approx on Σ^* . The predicate $MNR(\cdot, A)$ is called the Myhill-Nerode property with respect to A , and $MNR(\approx, A)$ expresses that \approx is a Myhill-Nerode relation for A .

(2) For any ℓ -valued binary relation \approx on Σ^* with finite $Range(\approx) = \{\approx(x, y) \in L : x, y \in \Sigma^*\}$, and for any ℓ -valued language A on Σ with finite $Range(A)$,

$$CMNR(\approx, A) \stackrel{def}{=} \gamma(r(\approx) \cup r(A)) \wedge MNR(\approx, A),$$

where $r(\approx) = \{\mathbf{a} : a \in Range(\approx)\}$. The predicate $CMNR(\cdot, A)$ is called the commutative Myhill-Nerode property.

As usual the Myhill-Nerode property in classical automata theory splits into two versions, noncommutative one and commutative one, in quantum logic. With the above definition, Lemma 3.36 may be roughly rephrased as that the relation $\approx_{\mathfrak{R}}$ induced by an ℓ -valued DFA \mathfrak{R} satisfies the Myhill-Nerode property with respect to $rec_{\mathfrak{R}}$.

We now turn to consider a binary relation on input strings induced naturally from a given language. Let A be an ℓ -valued language over Σ . We define ℓ -valued binary relation \approx_A as follows:

$$x \approx_A y \stackrel{def}{=} (\forall z \in \Sigma^*)(xz \in A \leftrightarrow yz \in A).$$

Thus the truth value of statement $x \approx_A y$ is given by

$$\lceil x \approx_A y \rceil = \bigwedge_{z \in \Sigma^*} (A(xz) \leftrightarrow R(yz)).$$

The following lemma presents some basic properties of relation \approx_A .

Lemma 3.38. For any ℓ -valued language A over Σ , it holds that

- (1) $\models^\ell Ref(\approx_A)$;
- (2) $\models^\ell Sym(\approx_A)$;
- (3) $\models^\ell RCon(\approx_A)$; and
- (4) $\models^\ell Refin(\approx_A, A)$.

In particular, if $\rightarrow = \rightarrow_3$ and $Range(A) = \{A(s) : s \in \Sigma^*\}$ is finite then we have:

- (5) $\models^\ell \gamma(r(A)) \rightarrow Tran(\approx_A)$, where $r(A) = \{\mathbf{a} : a \in Range(A)\}$, and \mathbf{a} is the nullary predicate associated with a for every $a \in L$.

Proof. (1), (2), (3) and (4) are obvious. For (5), it suffices to show that $\lceil \gamma(r(A)) \rceil \wedge \lceil x \approx_A y \rceil \wedge \lceil y \approx_A z \rceil \leq \lceil x \approx_A z \rceil$ for any $x, y, z \in \Sigma^*$. Indeed, with Lemma 2.12(3) we have:

$$\begin{aligned} \lceil \gamma(r(A)) \rceil \wedge \lceil x \approx_A y \rceil \wedge \lceil y \approx_A z \rceil &= \lceil \gamma(r(A)) \rceil \wedge \bigwedge_{u \in \Sigma^*} (A(xu) \leftrightarrow A(yu)) \wedge \bigwedge_{v \in \Sigma^*} (A(yv) \leftrightarrow A(zv)) \\ &\leq \bigwedge_{u \in \Sigma^*} (\lceil \gamma(r(A)) \rceil \wedge (A(xu) \leftrightarrow A(yu)) \wedge (A(yu) \leftrightarrow A(zu))) \\ &\leq \bigwedge_{u \in \Sigma^*} (A(xu) \leftrightarrow A(zu)) = \lceil x \approx_A z \rceil. \quad \square \end{aligned}$$

By combining the results obtained above we are able to establish an orthomodular lattice-valued generalization of the Myhill-Nerode theorem. A simpler presentation of this generalization requires us to introduce the following two auxiliary notions.

Definition 3.39. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice and $a \in L$. If for any subset M of L , there exists $b \in M$ such that $a \leq b$ provided $a \leq \bigvee M$, then a is called an atom of ℓ .

Definition 3.40. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let φ be a logical formula and Φ a set of logical formulas. If for any ℓ -valued interpretation, and for any atom a of ℓ , we have $[\varphi] \geq a$ whenever $[\psi] \geq a$ for all $\psi \in \Phi$, then φ is called an atomic consequence of Φ , and we write $\Phi \Vdash^\ell \varphi$.

It is obvious that $\Phi \models^\ell \varphi$ implies $\Phi \Vdash^\ell \varphi$. Now we are ready to give the main result of this section.

Theorem 3.41. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, let $\rightarrow = \rightarrow_3$, and let Σ be a finite alphabet.

(1) If $a \in L$ satisfies aCb for every $b \in L$, and ℓ is a -finite, then for any ℓ -valued language A over Σ with finite $Range(A)$, we have:

$$\models^\ell \mathbf{a} \wedge C DReg_\Sigma(A) \rightarrow (\exists (\ell\text{-valued}) \text{ binar relation } \approx \text{ on } \Sigma^*) MNR(\approx, A).$$

(2) For any ℓ -valued language A over Σ , it holds that

$$\models^\ell (\exists (\ell\text{-valued}) \text{ binary relation } \approx \text{ on } \Sigma^*) CMNR(\approx, A) \rightarrow FInd(\approx_A).$$

(3) For any ℓ -valued language A over Σ , we have:

$$FInd(\approx_A) \wedge \gamma(r(A)) \Vdash^\ell DReg_\Sigma(A).$$

Proof. (1) First, if $\mathfrak{R} \in \mathbf{DFA}(\Sigma, \ell)$, then for any $x, y \in \Sigma^*$, it follows from Lemma 2.12(1) and (4) that

$$\begin{aligned} & [\gamma(\text{atom}(\mathfrak{R}) \cup r(A))] \wedge [A \equiv \text{rec}_\mathfrak{R}] \wedge [\text{Refin}(\approx_\mathfrak{R}, \text{rec}_\mathfrak{R})] \leq [\gamma(\text{atom}(\mathfrak{R}) \cup r(A))] \wedge \\ & [x \approx_\mathfrak{R} y \rightarrow (\text{rec}_\mathfrak{R}(x) \leftrightarrow \text{rec}_\mathfrak{R}(y))] \wedge (A(x) \leftrightarrow \text{rec}_\mathfrak{R}(x)) \wedge (A(y) \leftrightarrow \text{rec}_\mathfrak{R}(y)) \\ & \leq [\gamma(\text{atom}(\mathfrak{R}) \cup r(A))] \wedge [x \approx_\mathfrak{R} y \rightarrow (\text{rec}_\mathfrak{R}(x) \leftrightarrow \text{rec}_\mathfrak{R}(y))] \wedge [x \approx_\mathfrak{R} y \rightarrow (A(x) \leftrightarrow \text{rec}_\mathfrak{R}(x))] \wedge \\ & [x \approx_\mathfrak{R} y \rightarrow (A(y) \leftrightarrow \text{rec}_\mathfrak{R}(y))] \wedge [x \approx_\mathfrak{R} y \rightarrow [\gamma(\text{atom}(\mathfrak{R}) \cup r(A))]] \\ & \leq x \approx_\mathfrak{R} y \rightarrow [\gamma(\text{atom}(\mathfrak{R}) \cup r(A))] \wedge (\text{rec}_\mathfrak{R}(x) \leftrightarrow \text{rec}_\mathfrak{R}(y)) \wedge (A(x) \leftrightarrow \text{rec}_\mathfrak{R}(x)) \wedge (A(y) \leftrightarrow \text{rec}_\mathfrak{R}(y)) \\ & \leq x \approx_\mathfrak{R} y \rightarrow (A(x) \leftrightarrow A(y)). \end{aligned}$$

Therefore, we obtain:

$$\lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil \wedge \lceil A \equiv \text{rec}_{\mathfrak{R}} \rceil \wedge \lceil \text{Refin}(\approx_{\mathfrak{R}}, \text{rec}_{\mathfrak{R}}) \rceil \leq \lceil \text{Refin}(\approx_{\mathfrak{R}}, A) \rceil.$$

Second, for any $\mathfrak{R} \in \mathbf{DFA}(\Sigma, \ell)$, using the above inequality and Lemma 3.36(1)-(6) we obtain:

$$\begin{aligned} \lceil MNR(\approx_{\mathfrak{R}}, A) \rceil &\geq \lceil MNR(\approx_{\mathfrak{R}}, \text{rec}_{\mathfrak{R}}) \rceil \wedge \lceil A \equiv \text{rec}_{\mathfrak{R}} \rceil \wedge \lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil \\ &\geq a \wedge \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil A \equiv \text{rec}_{\mathfrak{R}} \rceil \wedge \lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil \\ &= a \wedge \lceil A \equiv \text{rec}_{\mathfrak{R}} \rceil \wedge \lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil. \end{aligned}$$

Consequently, it follows from Lemma 2.3 that

$$\begin{aligned} \lceil (\exists(\ell - \text{valued}) \text{ binary relation } \approx \text{ on } \Sigma^*) MNR(\approx, A) \rceil &\geq \bigvee_{\mathfrak{R} \in \mathbf{DFA}(\Sigma, \ell)} \lceil MNR(\approx_{\mathfrak{R}}, A) \rceil \\ &\geq \bigvee_{\mathfrak{R} \in \mathbf{DFA}(\Sigma, \ell)} (a \wedge \lceil A \equiv \text{rec}_{\mathfrak{R}} \rceil \wedge \lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil) \\ &= a \wedge \bigvee_{\mathfrak{R} \in \mathbf{DFA}(\Sigma, \ell)} (\lceil A \equiv \text{rec}_{\mathfrak{R}} \rceil \wedge \lceil \gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \rceil) \\ &= a \wedge \lceil CDReg_{\Sigma}(A) \rceil \end{aligned}$$

because aCb for every $b \in L$.

(2) We first show that

$$\lceil x \approx y \rceil \wedge \lceil RCon(\approx) \rceil \wedge \lceil \gamma(r(\approx)) \rceil \leq \lceil xz \approx yz \rceil$$

for any $x, y, z \in \Sigma^*$ by induction on the length $|z|$ of z . It is clear for the case of $z = \varepsilon$. Suppose that the conclusion holds for z . Then for each $\sigma \in \Sigma$, with Lemma 2.12(5) we have:

$$\begin{aligned} \lceil x \approx y \rceil \wedge \lceil RCon(\approx) \rceil \wedge \lceil \gamma(r(\approx)) \rceil &\leq \lceil xz \approx yz \rceil \wedge \lceil RCon(\approx) \rceil \wedge \lceil \gamma(r(\approx)) \rceil \\ &\leq \lceil xz \approx yz \rceil \wedge (\lceil xz \approx yz \rightarrow xz\sigma \approx yz\sigma \rceil \wedge \lceil \gamma(r(\approx)) \rceil) \\ &\leq \lceil xz\sigma \approx yz\sigma \rceil. \end{aligned}$$

Second, it follows from Lemma 2.12(5) that

$$\begin{aligned} \lceil x \approx y \rceil \wedge \lceil RCon(\approx) \rceil \wedge \lceil \text{Refin}(\approx, A) \rceil \wedge \lceil \gamma(r(\approx) \cup r(A)) \rceil &\leq \\ \lceil xz \approx yz \rceil \wedge \lceil \text{Refin}(\approx, A) \rceil \wedge \lceil \gamma(r(\approx) \cup r(A)) \rceil & \\ \leq \lceil xz \approx yz \rceil \wedge (\lceil xz \approx yz \rceil \rightarrow (A(xz) \leftrightarrow A(yz))) \wedge \lceil \gamma(r(\approx) \cup r(A)) \rceil & \\ \leq A(xz) \leftrightarrow A(yz) & \end{aligned}$$

for any $z \in \Sigma^*$. Therefore, it holds that

$$\lceil x \approx y \rceil \wedge \lceil RCon(\approx) \rceil \wedge \lceil \text{Refin}(\approx, A) \rceil \wedge \lceil \gamma(r(\approx) \cup r(A)) \rceil \leq$$

$$\bigwedge_{z \in \Sigma^*} (A(xz) \leftrightarrow A(yz)) = [x \approx_A y]$$

for all $x, y \in \Sigma^*$.

Third, for each ℓ -valued binary relation \approx on Σ^* , using Lemmas 2.6 and 2.7 we obtain:

$$\begin{aligned} [CMNR(\approx, A)] &\leq [RCon(\approx)] \wedge [Refin(\approx, A)] \wedge [FInd(\approx)] \wedge [\gamma(r(\approx) \cup r(A))] \\ &= [RCon(\approx)] \wedge [Refin(\approx, A)] \wedge [\gamma(r(\approx) \cup r(A))] \wedge \bigvee_{n \in \omega, x_1, \dots, x_n \in \Sigma^*} \bigwedge_{x \in \Sigma^*} \bigvee_{i \leq n} [x \approx x_i] \\ &\leq \bigvee_{n \in \omega, x_1, \dots, x_n \in \Sigma^*} \bigwedge_{x \in \Sigma^*} \bigvee_{i \leq n} ([RCon(\approx)] \wedge [Refin(\approx, A)] \wedge [\gamma(r(\approx) \cup r(A))] \wedge [x \approx x_i]) \\ &\leq \bigvee_{n \in \omega, x_1, \dots, x_n \in \Sigma^*} \bigwedge_{x \in \Sigma^*} \bigvee_{i \leq n} [x \approx_A x_i] \\ &= [FInd(\approx_A)]. \end{aligned}$$

This yields:

$$\begin{aligned} [\exists (\ell - \text{valued}) \text{ binary relation } \approx \text{ on } \Sigma^* \text{ } CMNR(\approx, A)] &= \\ \bigvee \{ [CMNR(\approx, A)] : \approx \text{ is an } \ell - \text{valued binary relation on } \Sigma^* \} & \\ \leq [FInd(\approx_A)]. & \end{aligned}$$

(3) For any ℓ -valued interpretation, and for every atom a of ℓ , suppose that $a \leq [FInd(\approx_A) \wedge \gamma(r(A))]$, that is,

$$a \leq [FInd(\approx_A)] = \bigvee_{n \in \omega, x_1, \dots, x_n \in \Sigma^*} \bigwedge_{x \in \Sigma^*} \bigvee_{i \leq n} [x \approx_A x_i]$$

and $a \leq [\gamma(r(A))]$. We define binary relation $\overset{a}{\sim}$ on Σ^* as follows:

$$x \overset{a}{\sim} y \text{ if and only if } [x \approx_A y] \text{ (under this interpretation)}$$

for any $x, y \in \Sigma^*$. It is clear from Lemma 3.38(1) and (2) that $\overset{a}{\sim}$ is reflexive and symmetric. For any $x, y, z \in \Sigma^*$, if $x \overset{a}{\sim} y$ and $y \overset{a}{\sim} z$, then $[x \approx_A y] \geq a$ and $[y \approx_A z] \geq a$. With Lemma 3.38(5) we know that

$$[x \approx_A y] \wedge [y \approx_A z] \wedge [\gamma(r(A))] \leq [x \approx_A z].$$

Noting that $[\gamma(r(A))] \geq a$ we obtain $[x \approx_A z] \geq a$ and $x \overset{a}{\sim} z$. Therefore, $\overset{a}{\sim}$ is transitive.

We now construct the quotient $\Sigma^* / \overset{a}{\sim} = \{[x] : x \in \Sigma^*\}$, where $[x]$ stands for the equivalence class of x with respect to $\overset{a}{\sim}$ for any $x \in \Sigma^*$. From $[FInd(\approx_A)] \geq a$, we know that there are $x_1, \dots, x_n \in \Sigma^*$ ($n \in \omega$) such that for each $x \in \Sigma^*$, $[x \approx_A x_i] \geq a$ for some $i \leq n$ because a is an atom of ℓ . Consequently, for any $x \in \Sigma^*$, it holds that $x \overset{a}{\sim} x_i$ for some $i \leq n$, and $\Sigma^* / \overset{a}{\sim} = \{[x_1], \dots, [x_n]\}$ is a finite set (note

that it is possible to have $[x_i] = [x_j]$ for some $1 \leq i < j \leq n$). This enables us to construct an ℓ -valued deterministic finite automaton with $\Sigma^*/\overset{a}{\sim}$ as its set of states: $\mathfrak{R} = \langle \Sigma^*/\overset{a}{\sim}, \delta, [\varepsilon], T \rangle$, where

- (i) ε is the empty string;
- (ii) $T : \Sigma^*/\overset{a}{\sim} \rightarrow L$ is defined by

$$T([x]) = \bigvee_{y \in [x]} A(y)$$

for every $x \in \Sigma^*$; and

- (iii) $\delta([x], \sigma) = [x\sigma]$ for any $x \in \Sigma^*$ and $\sigma \in \Sigma$.

It is guaranteed by Lemma 3.38(3) that δ is well-defined (note that δ is an ordinary mapping from $\Sigma^*/\overset{a}{\sim} \times \Sigma$ into $\Sigma^*/\overset{a}{\sim}$, and ℓ is indeed not involved in δ). Moreover, it is easy to see that $\delta([x], y) = [xy]$ by induction on the length of y for any $x, y \in \Sigma^*$.

Finally, we assert that

$$\begin{aligned} DReg_{\Sigma}(A) &\geq [A \equiv rec_{\mathfrak{R}}] \\ &= \bigwedge_{x \in \Sigma^*} (A(x) \leftrightarrow T(\delta([\varepsilon], x))) \\ &= \bigwedge_{x \in \Sigma^*} (A(x) \leftrightarrow T([x])) \\ &= \bigwedge_{x \in \Sigma^*} (A(x) \leftrightarrow \bigwedge_{y \in [x]} A(y)) \\ &\geq \bigwedge_{x \in \Sigma^*} [\bigwedge_{y \in [x]} (A(x) \leftrightarrow A(y)) \wedge [\gamma(r(A))]] \geq a \end{aligned}$$

because $[\gamma(r(A))] \geq a$, and for each $x \in \Sigma^*$ and $y \in [x]$, with Lemma 3.38(4) it holds that $A(x) \leftrightarrow A(y) \geq [x \approx_A y] \geq a$. \square

3.7. Pumping Lemma for Orthomodular Lattice-Valued Regular Languages

The pumping lemma in the classical automata theory is a powerful tool to show that certain languages are not regular, and it exposes some limitations of finite automata. The purpose of this section is to establish a generalization of the pumping lemma for orthomodular lattice-valued languages. It is worth noting that the following orthomodular lattice-valued version of pumping lemma is given for the commutative regularity. In general, the pumping lemma is not valid for noncommutative regularity. In addition, in the pumping lemma we take the implication operator to be the Sasaki hook \rightarrow_3 , and such a pumping lemma does not hold if other implications are adopted. From Corollary 3.10 we know that $CReg_{\Sigma}^{[D]}$ and $CReg_{\Sigma}^{[W]}$ are equivalent. Thus, in this section we only consider $CReg_{\Sigma}^{[D]}$.

Theorem 3.42. (The pumping lemma) Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and let $\rightarrow = \rightarrow_3$. For any $A \in L^{\Sigma^*}$, if $\text{Range}(A)$ is finite, then

$$\models^\ell CReg_\Sigma^{[D]}(A) \rightarrow (\exists n \geq 0)(\forall s \in \Sigma^*)[s \in A \wedge |s| \geq n \rightarrow (\exists u, v, w \in \Sigma^*)(s = uvw \wedge |uv| \leq n \wedge |v| \geq 1 \wedge (\forall i \geq 0)(uv^i w \in A))],$$

where for any word $t = \sigma_1 \dots \sigma_k \in \Sigma^*$, $|t|$ stands for the length n of t .

Proof. For simplicity, we use $X(s, n)$ to mean the statement that $u, v, w \in \Sigma^*$, $s = uvw$, $|uv| \leq n$, and $|v| \geq 1$ for each $s \in \Sigma^*$ and $n \geq 0$. Then it suffices to show that

$$[CReg_\Sigma^{[D]}(A)] \leq \bigvee_{n \geq 0} \bigwedge_{s \in \Sigma^*, |s| \geq n} (A(s) \rightarrow \bigvee_{X(s, n)} \bigwedge_{i \geq 0} A(uv^i w)).$$

From Definition 3.9 we know that

$$[CReg_\Sigma^{[D]}(A)] = \bigvee_{\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)} ([\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge [A \equiv \text{rec}_\mathfrak{R}^{[D]}]]).$$

Thus, we only need to prove that for any $\mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell)$,

$$[\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge [A \equiv \text{rec}_\mathfrak{R}^{[D]}]] \leq \bigvee_{n \geq 0} \bigwedge_{s \in \Sigma^*, |s| \geq n} (A(s) \rightarrow \bigvee_{X(s, n)} \bigwedge_{i \geq 0} A(uv^i w)).$$

Let Q be the set of states of \mathfrak{R} . First, it holds that for any $s \in \Sigma^*$ with $|s| \geq |Q|$,

$$(1) \quad \text{rec}_\mathfrak{R}^{[D]}(s) \leq \bigvee_{X(s, n)} \bigwedge_{i \geq 0} \text{rec}_\mathfrak{R}^{[D]}(uv^i w).$$

In fact, suppose that $s = \sigma_1 \dots \sigma_k$. Then

$$(2) \quad \text{rec}_\mathfrak{R}^{[D]}(s) = \bigvee_{q_0, q_1, \dots, q_k} [I(q_0) \wedge T(q_k) \wedge \bigwedge_{i=0}^{k-1} \delta(q_i, \sigma_{i+1}, q_{i+1})].$$

Therefore, it amounts to showing that for any $q_0, q_1, \dots, q_k \in Q$,

$$(3) \quad I(q_0) \wedge T(q_k) \wedge \bigwedge_{i=0}^{k-1} \delta(q_i, \sigma_{i+1}, q_{i+1}) \leq \bigvee_{X(s, n)} \bigwedge_{i \geq 0} \text{rec}_\mathfrak{R}^{[D]}(uv^i w).$$

Since $k = |s| \geq |Q|$, there are two identical states among $q_0, q_1, \dots, q_{|Q|}$; in other words, there are $m \geq 0$ and $n > 0$ such that $m + n \leq |Q|$ and $q_m = q_{m+n}$. We

set $u_0 = \sigma_1 \dots \sigma_m$, $v_0 = \sigma_{m+1} \dots \sigma_{m+n}$, and $w_0 = \sigma_{m+n+1} \dots \sigma_k$. Then $s = u_0 v_0 w_0$, $|u_0 v_0| = m + n \leq |Q|$, $|v| = n \geq 1$, and

$$(4) \quad \bigvee_{X(s,n)} \bigwedge_{i \geq 0} \text{rec}_{\mathfrak{R}}^{[D]}(uv^i w) \geq \bigwedge_{i \geq 0} \text{rec}_{\mathfrak{R}}^{[D]}(u_0 v_0^i w_0).$$

From the definition of $\text{rec}_{\mathfrak{R}}^{[D]}$, it is easy to see that for all $i \geq 0$,

$$(5) \quad \begin{aligned} \text{rec}_{\mathfrak{R}}^{[D]}(u_0 v_0^i w_0) &\geq [\text{Path}_{\mathfrak{R}}(q_0 \sigma_1 q_1 \dots \sigma_m q_m \\ &\quad (\sigma_{m+1} q_{m+1} \dots \sigma_{m+n} q_{m+n})^i \sigma_{m+n+1} q_{m+n+1} \dots \sigma_k q_k)] \\ &= I(q_0) \wedge T(q_k) \wedge \bigwedge_{j=0}^{m+n-1} \delta(q_j, \sigma_{j+1}, q_{j+1}) \wedge \bigwedge_{l=1}^{i-1} [\delta(q_{m+n}, \sigma_{m+1}, q_{m+1}) \wedge \\ &\quad \bigwedge_{j=m+1}^{m+n-1} \delta(q_j, \sigma_{j+1}, q_{j+1})] \wedge \bigwedge_{j=m+n}^{k-1} \delta(q_j, \sigma_{j+1}, q_{j+1}) \\ &= I(q_0) \wedge T(q_k) \wedge \bigwedge_{j=0}^{k-1} \delta(q_j, \sigma_{j+1}, q_{j+1}) \end{aligned}$$

because $q_{m+n} = q_m$ and $\delta(q_{m+n}, \sigma_{m+1}, q_{m+1}) = \delta(q_m, \sigma_{m+1}, q_{m+1})$. Thus, by combining (4) and (5), we obtain (3) which, together with (2), yields (1).

Now we use Lemmas 2.12(1) and (3) and obtain:

$$\begin{aligned} &\bigvee_{X(s,|Q|)} \bigwedge_{i \geq 0} \text{rec}_{\mathfrak{R}}^{[D]}(uv^i w) \rightarrow \bigvee_{X(s,|Q|)} \bigwedge_{i \geq 0} A(uv^i w) \geq [\gamma(\text{atom}(\mathfrak{R}) \cup r(A))] \wedge \\ &\quad \bigwedge_{X(s,|Q|)} (\bigwedge_{i \geq 0} \text{rec}_{\mathfrak{R}}^{[D]}(uv^i w) \rightarrow \bigwedge_{i \geq 0} A(uv^i w)) \\ &\geq [\gamma(\text{atom}(\mathfrak{R}) \cup r(A))] \wedge \bigwedge_{X(s,|Q|)} \bigwedge_{i \geq 0} (\text{rec}_{\mathfrak{R}}^{[D]}(uv^i w) \rightarrow A(uv^i w)) \\ &\geq [\gamma(\text{atom}(\mathfrak{R}) \cup r(A))] \wedge \bigwedge_{t \in \Sigma^*} (\text{rec}_{\mathfrak{R}}^{[D]}(t) \rightarrow A(t)) \\ &= [\gamma(\text{atom}(\mathfrak{R}) \cup r(A))] \wedge [\text{rec}_{\mathfrak{R}}^{[D]} \subseteq A]. \end{aligned}$$

Furthermore, from the above inequality we have:

$$\begin{aligned}
& [\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge [\text{rec}_{\mathfrak{R}}^{[D]} \equiv A] = [\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge [A \subseteq \text{rec}_{\mathfrak{R}}^{[D]}] \wedge [\text{rec}_{\mathfrak{R}}^{[D]} \subseteq A] \\
& = [\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge \bigwedge_{s \in \Sigma^*} (A(s) \rightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s)) \wedge [\text{rec}_{\mathfrak{R}}^{[D]} \subseteq A] \\
& \leq [\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge \bigwedge_{s \in \Sigma^*, |s| \geq |Q|} (A(s) \rightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s)) \wedge [\text{rec}_{\mathfrak{R}}^{[D]} \subseteq A] \\
& = \bigwedge_{s \in \Sigma^*, |s| \geq |Q|} ([\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge (A(s) \rightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s)) \wedge [\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge [\text{rec}_{\mathfrak{R}}^{[D]} \subseteq A]) \\
& \leq \bigwedge_{s \in \Sigma^*, |s| \geq |Q|} ([\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge (A(s) \rightarrow \text{rec}_{\mathfrak{R}}^{[D]}(s)) \\
& \quad \wedge (\bigvee_{X(s, |Q|)} \bigwedge_{i \geq 0} \text{rec}_{\mathfrak{R}}^{[D]}(uv^i w) \rightarrow \bigvee_{X(s, |Q|)} \bigwedge_{i \geq 0} A(uv^i w))].
\end{aligned}$$

Then from (1) it follows that

$$\begin{aligned}
[\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge [\text{rec}_{\mathfrak{R}}^{[D]} \equiv A] & \leq \bigwedge_{s \in \Sigma^*, |s| \geq |Q|} ([\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge \\
& (A(s) \rightarrow \bigvee_{X(s, |Q|)} \bigwedge_{i \geq 0} \text{rec}_{\mathfrak{R}}^{[D]}(uv^i w)) \wedge \\
& (\bigvee_{X(s, |Q|)} \bigwedge_{i \geq 0} \text{rec}_{\mathfrak{R}}^{[D]}(uv^i w) \rightarrow \bigvee_{X(s, |Q|)} \bigwedge_{i \geq 0} A(uv^i w))].
\end{aligned}$$

By using Lemmas 2.12(1) and (3) we know that

$$\begin{aligned}
[\gamma(\text{atom}(\mathfrak{R}) \cup r(A)) \wedge [\text{rec}_{\mathfrak{R}}^{[D]} \equiv A] & \leq \bigwedge_{s \in \Sigma^*, |s| \geq |Q|} (A(s) \rightarrow \bigvee_{X(s, |Q|)} \bigwedge_{i \geq 0} A(uv^i w)) \\
& \leq \bigvee_{n \geq 0} \bigwedge_{s \in \Sigma^*, |s| \geq n} (A(s) \rightarrow \bigvee_{X(s, n)} \bigwedge_{i \geq 0} A(uv^i w)),
\end{aligned}$$

and this completes the proof. \square

4. Orthomodular Lattice-Valued Pushdown Automata

Pushdown automaton is another mathematical model of finite state machines. It is more powerful than finite automaton, and it differs from finite automaton in two ways: (1) it has a stack and can use the top symbol of the stack to figure out what transition to take; and (2) it can manipulate the stack during performing a transition. Pushdown automata and their accepted context-free languages have been widely applied in the specification of programming languages and in the design and implementation of compilers. They have also found successful applications in the study of natural languages.

The purpose of this Section is to re-build the theory of pushdown automata in the framework of quantum logic and to observe the essential difference between it and the classical theory of pushdown automata.

4.1. Orthomodular Lattice-Valued Context-Free Grammars

We recall that a context-free grammar over a given finite alphabet Σ , whose elements are usually called terminals, is a triple $G = \langle N, P, S \rangle$, where

- (i) N is a finite set of nonterminal symbols, and it is required that $N \cap \Sigma = \emptyset$;
- (ii) $S \in N$ is the start symbol; and
- (iii) P is a finite subset of $Prod(G) \stackrel{def}{=} \{A \rightarrow \alpha : A \in N \text{ and } \alpha \in (N \cup \Sigma)^*\}$,

whose elements are called productions.

For a context-free grammar $G = \langle N, P, S \rangle$ over alphabet Σ , the direct derivation relation \Rightarrow_G is defined to be a binary relation on $(N \cup \Sigma)^*$ in such a way: $\alpha \Rightarrow_G \beta$ if and only if there are $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$ and $A \rightarrow \gamma \in P$ such that $\alpha = \alpha_1 A \alpha_2$ and $\beta = \alpha_1 \gamma \alpha_2$. We write $\xRightarrow{*}_G$ for the reflexive and transitive closure of \Rightarrow_G , and it is called the derivation relation in G . Then the language $L(G)$ generated by G is defined by

$$L(G) \stackrel{def}{=} \{w \in \Sigma^* : S \xRightarrow{*}_G w\}.$$

A language $A \subseteq \Sigma^*$ is said to be context-free if $A = L(G)$ for some context-free grammar G .

The notion of orthomodular lattice-valued context-free grammar can be formally defined in a similar way.

Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice and Σ a finite alphabet. Then an ℓ -valued context-free grammar (CFG for short) is a triple $G = \langle N, P, S \rangle$, where N and S are the same as in a classical context-free grammar, P is a finite ℓ -valued subset of $Prod(G)$, that is, a mapping from $Prod(G)$ into L such that $suppP = \{p \in Prod(G) : P(p) > 0\}$ is a finite set, and $Prod(G)$ is defined as in the classical case.

The (proper) class of ℓ -valued CFGs over Σ is denoted by $\mathbf{CFG}(\Sigma, \ell)$.

There are two ways of defining the language generated by an ℓ -valued CFG. Both of them are natural generalizations of the corresponding classical definition. In classical automata theory, distributivity of Boolean logic warrants that these two ways are equivalent. However, they become nonequivalent in the case of ℓ -valued CFGs due to lack of distributivity in quantum logic. The difference between them comes mainly from how we evaluate the truth value of the proposition that a word is generated from the truth values of the propositions concerning the involved transitions.

We first consider the depth-first way. Let $G = \langle N, P, S \rangle$ be an ℓ -valued CFG over alphabet Σ . For any $\alpha, \beta \in (N \cup \Sigma)^*$ and production $p = A \rightarrow \gamma \in \text{Prod}(G)$, if there exist $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$ such that $\alpha = \alpha_1 A \alpha_2$ and $\beta = \alpha_1 \gamma \alpha_2$, then it is said that p is compatible with the direct derivation relation $\alpha \xrightarrow{G} \beta$.

By the term a quasi-derivation (of length $n \geq 0$) we mean an element

$$d = ((\alpha_0, \alpha_1, \dots, \alpha_n), (p_1, \dots, p_n)) \in (N \cup \Sigma)^{n+1} \times \text{Prod}(G)^n$$

such that p_i is compatible with $\alpha_{i-1} \xrightarrow{G} \alpha_i$ for all $i \leq n$. The length of d is $|d| = n$. We write $QDer_G$ for the set of quasi-derivations in G . The ℓ -valued (unary) predicate $Der_G \in L^{QDer_G}$, interpreted as “to be a derivation in G ”, is then defined by

$$Der_G((\alpha_0, \alpha_1, \dots, \alpha_n), (p_1, \dots, p_n)) \stackrel{def}{=} \bigwedge_{i=1}^n (p_i \in P)$$

for any $d = ((\alpha_0, \alpha_1, \dots, \alpha_n), (p_1, \dots, p_n)) \in QDer_G$, $n \geq 0$.

With the above technical notations we are able to present a formal definition of the language generated by an ℓ -valued CFG in the depth-first way.

Definition 4.1. For any ℓ -valued CFG $G = \langle N, P, S \rangle$ over Σ . The language $L^{[D]}(G)$ generated by G in the depth-first way is defined to be an ℓ -valued subset of Σ^* , and it is given by

$$w \in L^{[D]}(G) \stackrel{def}{=} (\exists d \in QDer_G)(Der_G(d) \wedge I(d) = S \wedge L(d) = w)$$

for every $w \in \Sigma^*$, where $I(d)$ and $L(d)$ are respectively the first and the last strings of nonterminals and terminals in d , namely, $I(d) = \alpha_0$ and $L(d) = \alpha_n$ whenever $d = ((\alpha_0, \alpha_1, \dots, \alpha_n), (p_1, \dots, p_n))$.

The context-freeness in automata theory based on quantum logic is defined to be an ℓ -valued predicate over ℓ -valued languages. Remember that regularity of languages in classical automata theory splits into two nonequivalent versions in quantum logic: (noncommutative) regularity and commutative regularity. The latter is obtained by adding a certain commutator into the former (see Definitions 3.4 and 3.9). Obviously, we can also define commutative and noncommutative versions of

context-freeness. But it may be observed that in evaluating the language generated by an ℓ -valued CFG according to the depth-first principle, all meet operations occur at the innermost, and they do not interact with any join operations which are at the outermost. Thus, distributivity of meet over join is not required at all, and we do not need to add any commutator in order to recover local distributivity. This suggests us to consider only noncommutative version of context-freeness when depth-first principle is applied.

Definition 4.2. Let Σ be a finite alphabet. The ℓ -valued (unary) predicate $CFG_{\Sigma}^{[D]}$ on L^{Σ^*} is defined by

$$CFG_{\Sigma}^{[D]}(A) \stackrel{def}{=} (\exists G \in \mathbf{CFG}(\Sigma, \ell))(A \equiv L^{[D]}(G))$$

for each $A \in L^{\Sigma^*}$, and it is interpreted as “to be a (noncommutative) context-free language over Σ according to the depth-first principle.”

The appearance of the proper class $\mathbf{CFG}(\Sigma, \ell)$ in the above definition requires a set-theoretic explanation. Indeed, a simple modification of the foundational exposition after Definition 3.2 can serve for this purpose.

To illustrate the above two definitions, let us consider a simple example.

Example 4.3. We consider the orthomodular lattice $\ell = MO_n$ ($n \geq 2$) (see Figure 10). Note that MO_2 is exactly the lattice $L(x) \oplus L(\bar{x})$ considered in Example 3.8. Let $\Sigma = \{a, b\}$ and $N = \{S, S_1, \dots, S_n\}$. We put

$$P(S \rightarrow a^n S b^n) = 1,$$

$$P(S \rightarrow a S_i b) = \lambda_i \text{ and } P(S \rightarrow a^{i-1} b^{i-1}) = 1 \text{ (} i = 1, \dots, n),$$

and $P(p) = 0$ for other productions p . Then $G = \langle N, P, S \rangle$ is an ℓ -valued CFG over Σ , and

$$L^{(D)}(G)(w) = \begin{cases} \lambda_i, & \text{if } w = a^{kn+i} b^{kn+i} \text{ for some } k \geq 0, \\ 1, & \text{otherwise.} \end{cases}$$

We further suppose that $1 \leq l \leq n$, and ℓ -valued language A_l over Σ is defined by

$$A_l(w) = \begin{cases} \lambda_i, & \text{if } w = a^{kn+i} b^{kn+i} \text{ for some } k \geq 0, \\ \lambda_l^\perp, & \text{if } |w| \equiv l \pmod{n} \text{ and } w \text{ is not of the form } a^m b^m \text{ for any } m \geq 1, \\ 1, & \text{otherwise.} \end{cases}$$

Then if $\rightarrow = \rightarrow_j$ ($j = 0, 1, \dots, 5$) we have:

$$[CF L_{\Sigma}^{[D]}(A_l)] \geq [A \equiv L^{[D]}(G)] = \lambda_l^\perp \leftrightarrow 0 = \lambda_l.$$

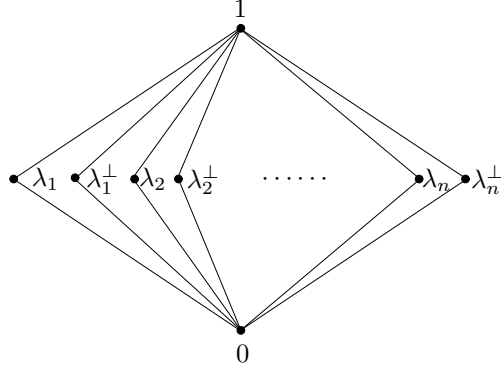


Figure 10: Orthomodular lattice MO_n

It is obvious that there may be some symbols in an ℓ -valued CFG G which are useless in generating its language $L^{[D]}(G)$. As in the case of classical context-free grammars, ℓ -valued CFGs can be simplified by dropping these useless symbols. Let $G\langle N, P, S \rangle$ be an ℓ -valued CFG over Σ and $L^{[D]}(G) \neq \emptyset$. For any symbol X in N or Σ , if there exists a quasi-derivation $((\alpha_0, \alpha_1, \dots, \alpha_n), (p_1, \dots, p_n))$ in G such that $\alpha_0 = S$, $\alpha_n \in \Sigma^*$, X appears in at least one of α_i ($0 \leq i \leq n$), and $\bigwedge_{i=1}^n P(p_i) > 0$, then X is said to be useful. Clearly, S is useful because $L^{[D]}(G) \neq \emptyset$. We put N' and Σ' to be the sets of useful symbols in N and Σ , respectively, and define $P'(p) = P(p)$ if all symbols in p are in N' and Σ' . Then $G' = \langle N', P', S \rangle$ is an ℓ -valued CFG over Σ' , and $L^{[D]}(G) = L^{[D]}(G')$.

As shown by the next lemma, we can further simplify an ℓ -valued CFG by eliminating some trivial productions.

Lemma 4.4. For any ℓ -valued CFG $G = \langle N, \Sigma, P, S \rangle$, there exists an ℓ -valued CFG $G' = \langle N, \Sigma, P', S \rangle$ satisfying the following conditions:

- (1) $P'(p) = 0$ for all $p \in Prod_\varepsilon(G) = \{A \rightarrow \varepsilon : A \in N'\}$, the set of ε -productions in G' ;
- (2) $P'(p) = 0$ for all $p \in Prod_{unit}(G) = \{A \rightarrow B : A, B \in N'\}$, the set of unit productions in G' ; and
- (3) $L^{[D]}(G') = L^{[D]}(G) - \{\varepsilon\}$, where for any ℓ -valued language L over Σ , $L - \{\varepsilon\}$ is defined by

$$(L - \{\varepsilon\})(w) = \begin{cases} L(w), & \text{if } w \neq \varepsilon, \\ 0, & \text{if } w = \varepsilon. \end{cases}$$

Proof. We construct G' from G in two steps. First, we eliminate all ε -productions from G . For any production $p = A \rightarrow X_1X_2\dots X_n \in Prod(G)$, for any subsequence $I = (i_1, i_2, \dots, i_n)$ of $(1, 2, \dots, n)$, and for any $d_j \in QDer_G$ with $I(d_j) = X_{i_j}$ and $L(d_j) = \varepsilon$ ($j = 1, 2, \dots, k$), we introduce new terminals $A_j^{(p, I, \vec{d})}$ ($j = 1, 2, \dots, k$),

where $\vec{d} = (d_1, d_2, \dots, d_k)$. Note that $A_j^{(p,I,\vec{d})}$ are not defined whenever X_{i_j} is a terminal for some $j \leq k$. Moreover, $A_j^{(p,I,\vec{d})}$ are different for different (p, I, \vec{d}) .

Let

$$N'' = N \cup \bigcup_{p,I,\vec{d}} \{A_1^{(p,I,\vec{d})}, \dots, A_k^{(p,I,\vec{d})}\}$$

and $G'' = \langle N'', P'', S \rangle$, where P'' is defined as follows:

$$\begin{aligned} P''(A \rightarrow X_1 \dots X_{i_1-1} A_1^{(p,I,\vec{d})}) &= P(p), \\ P''(A_1^{(p,I,\vec{d})} \rightarrow X_{i_1+1} \dots X_{i_2-1} A_2^{(p,I,\vec{d})}) &= [Der_G(d_1)], \\ &\dots\dots\dots \\ P''(A_{k-1}^{(p,I,\vec{d})} \rightarrow X_{i_{k-1}+1} \dots X_{i_k-1} A_k^{(p,I,\vec{d})}) &= [Der_G(d_{k-1})], \\ P''(A_k^{(p,I,\vec{d})} \rightarrow X_{i_k+1} \dots X_n) &= [Der_G(d_k)], \end{aligned}$$

and $P''(p) = P(p)$ if $p \in Prod(G) - Prod_\varepsilon(G)$, $P''(p) = 0$ for other productions p in $Prod(G'')$. It is clear that G'' satisfies the condition (1). On the other hand, for each $w \in \Sigma^* - \{\varepsilon\}$, and for each $A \in N$, it is routine to prove the following claim by induction on the length $|d|$ of d :

Claim 1. for any $d \in QDer_G$ with $I(d) = A$ and $L(d) = w$, we can find $d'' \in QDer_{G''}$ such that $I(d'') = A$, $L(d'') = w$, and $[Der_{G''}(d'')] = [Der_G(d)]$.

Similarly, we can prove the following claim:

Claim 2. for any $d \in QDer_{G''}$ with $I(d) = A$ and $L(d) = w$, we can find $d'' \in QDer_G$ such that $I(d'') = A$, $L(d'') = w$, and $[Der_G(d'')] = [Der_{G''}(d)]$.

By combining the above two claims we are able to assert that $L^{[D]}(G)(w) = L^{[D]}(G'')(w)$ provided $w \neq \varepsilon$.

Second, we eliminate all unit productions from G'' . For any $A, B \in N''$, and for any $d \in QDer_{G''}$ with $I(d) = A$ and $L(d) = B$, we set

$$P_{A,B,d}(p) = \begin{cases} P(B \rightarrow \alpha) \wedge [Der_{G''}], & \text{if } p = A \rightarrow \alpha, \text{ where } \alpha \in (N'' \cup \Sigma)^*, \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, we define

$$P^* = P \cup \bigcup_{A,B,d} P_{A,B,d}$$

and

$$P'(p) = \begin{cases} 0, & \text{if } p \in Prod_{unit}(G''), \\ P^*(p), & \text{otherwise.} \end{cases}$$

Then we have $G' = \langle N'', P', S \rangle \in \mathbf{CFG}(\ell, \Sigma)$, and it satisfies the conditions (1) and (2). Finally, it is easy to verify that $L^{[D]}(G') = L^{[D]}(G'') = L^{[D]}(G) - \{\varepsilon\}$. \square

Two of the most useful special forms of context-free grammars are Chomsky normal form and Greibach normal form. The orthomodular lattice-valued extensions of them are give by the following definition.

Definition 4.5. Let Σ be an alphabet and N a set of nonterminal symbols. We write

$$CNF(\Sigma, N) \stackrel{def}{=} \{A \rightarrow a : A \in N \text{ and } a \in \Sigma\} \cup \{A \rightarrow BC : A, B, C \in N\}$$

and

$$GNF(\Sigma, N) \stackrel{def}{=} \{A \rightarrow a\alpha : a \in \Sigma \text{ and } \alpha \in N^*\}.$$

Then for any $G = \langle N, P, S \rangle$ in $\mathbf{CFG}(\Sigma, \ell)$, we say G is in Chomsky normal form (CNF for short) if $\text{supp}P \subseteq CNF(\Sigma, N)$, and G is said to be in Greibach normal form (GNF for short) if $\text{supp}P \subseteq GNF(\Sigma, N)$.

The next two theorems establish the generalizations of Chomsky normal form theorem and Greibach normal form theorem in quantum logic.

Theorem 4.6. For any ℓ -valued language $A \in L^{\Sigma^*}$ over Σ , we have:

$$\models^{\ell} CFL_{\Sigma}^{[D]}(A) \rightarrow (\exists \text{CNF } G)(L^{[D]}(G) \equiv A - \{\varepsilon\}).$$

Proof. For any G' in $\mathbf{CFG}(\Sigma, \ell)$, with Lemma 4.4 we are able to find G'' in $\mathbf{CFG}(\Sigma, \ell)$ satisfying the following two conditions:

- (i) $L^{[D]}(G'') = L^{[D]}(G' - \{\varepsilon\})$;
- (ii) $P''(p) = 0$ for all $p \in \text{Prod}_{\varepsilon}(G'') \cup \text{Prod}_{\text{unit}}(G'')$.

Consequently, it holds that

$$\begin{aligned} [CFL_{\Sigma}^{[D]}(A)] &= \bigvee_{G' \in \mathbf{CFG}(\Sigma, \ell)} [A \equiv L^{(D)}(G')] \\ &\leq \bigvee_{G'' \text{ satisfies (ii)}} [A - \{\varepsilon\} \equiv L^{(D)}(G'')]. \end{aligned}$$

Now it suffices to show that for any $G'' = \langle N'', P'', S \rangle \in \mathbf{CFG}(\Sigma, \ell)$, there exists CNF $G \in \mathbf{CFG}(\Sigma, \ell)$ with $L^{[D]}(G) = L^{[D]}(G'')$ provided G'' satisfies (ii). The proof is similar to Theorem 4.5 in [37]. First, for each $a \in \Sigma$, we introduce a new nonterminal symbol C_a . Let $N_1 = N'' \cup \{C_a : a \in \Sigma\}$. We put $G_1 = \langle N_1, P_1, S \rangle \in \mathbf{CFG}(\Sigma, \ell)$, where for each $p \in \text{Rule}(G_1)$, $P_1(p)$ is defined as follows:

$$P_1(p) = \begin{cases} P''(p'), & \text{if } p \text{ is obtained from } p' \in \text{Rule}(G'') \text{ by replacing all terminals} \\ & a \text{ in } p \text{ with } C_a, \\ 1, & \text{if } p = C_a \rightarrow a \text{ for some } a \in \Sigma, \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to see that P_1 is well-defined. Furthermore, by induction on the length of d we may prove the following two claims (here the routine but dull details are omitted), which imply $L^{[D]}(G_1) = L^{[D]}(G'')$:

Claim 1. For any $d \in QDer_{G''}$, there exists $d' \in QDer_{G_1}$ such that $I(d') = I(d)$, $L(d') = L(d)$ and $\lceil Der_{G_1}(d') \rceil = \lceil Der_{G''}(d) \rceil$.

Claim 2. For any $d \in QDer_{G_1}$, there exists $d' \in QDer_{G''}$ such that $I(d') = I(d)$, $L(d') = L(d)$ and $\lceil Der_{G_1}(d) \rceil \leq \lceil Der_{G''}(d') \rceil$.

We now construct $G = \langle N, P, S \rangle$ by modifying G_1 . Obviously, for any $p \in Rule(G_1)$, if $P_1(p) > 0$, then p is of the form $A \rightarrow a$ or $A \rightarrow A_1 \dots A_n$ with $A, A_1, \dots, A_n \in N_1$, $a \in \Sigma$ and $n \geq 2$. For any $\vec{A} = (A, A_1, \dots, A_n) \in N_1^{n+1}$ ($n \geq 2$), we introduce new nonterminals $D_{\vec{A},1}, \dots, D_{\vec{A},|\vec{A}|-2}$, where $|\vec{A}| = n + 1$. Let

$$N = N_1 \cup \bigcup_{\vec{A}} \{D_{\vec{A},1}, \dots, D_{\vec{A},|\vec{A}|-2}\}.$$

We also define

$$P_{\vec{A}}(A \rightarrow A_1 D_{\vec{A},1}) = P(A \rightarrow A_1 \dots A_n),$$

$$P_{\vec{A}}(D_{\vec{A},j} \rightarrow A_j D_{\vec{A},j+1}) = P_{\vec{A}}(D_{\vec{A},n-1} \rightarrow A_{n-1} A_n) = 1 \quad (j = 1, \dots, n-2)$$

and $P_{\vec{A}}(p) = 0$ for other $p \in Rule(G)$. Put

$$P = \bigcup_{\vec{A}} P_{\vec{A}} \cup P_*$$

where

$$P_* = \begin{cases} P_1(p), & \text{if } p \text{ is of the form } A \rightarrow a \text{ with } A \in N_1 \text{ and } a \in \Sigma, \\ 0, & \text{otherwise.} \end{cases}$$

By an argument similar to the above claims we are able to show that $L^{[D]}(G) = L^{[D]}(G_1)$ and thus complete the proof. \square

Theorem 4.7. For any ℓ -valued language A over Σ , it holds that

$$\models^\ell CFL_\Sigma^{[D]}(A) \rightarrow (\exists \text{GNF } G)(L^{[D]}(G) \equiv A - \{\varepsilon\}).$$

Proof. We first observe the following two technical facts:

Fact 1. Let $G = \langle N, P, S \rangle \in \mathbf{CFG}(\Sigma, \ell)$, and let $A, B \in N$ and $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$. We put

$$P_1(p) = \begin{cases} P(p), & \text{if } p \in Rule(G) - \{A \rightarrow \alpha_1 B \alpha_2\}, \\ 0, & \text{if } p = A \rightarrow \alpha_1 B \alpha_2, \end{cases}$$

$$P_2(p) = \begin{cases} P(A \rightarrow \alpha_1 B \alpha_2) \wedge P(B \rightarrow \gamma), & \text{if } p = A \rightarrow \alpha_1 \gamma \alpha_2 \text{ for some } \gamma \in (N \cup \Sigma)^*, \\ 0, & \text{otherwise} \end{cases}$$

and $G' = \langle N, P_1 \cup P_2, S \rangle$. Then $G' \in \mathbf{CFG}(\Sigma, \ell)$ and $L^{[D]}(G') = L^{[D]}(G)$.

Note that the production $A \rightarrow \alpha_1 B \alpha_2$ is deleted and it is no longer in $\text{supp}(B_1 \cup B_2)$.

Fact 2. Let $G = \langle N, P, S \rangle \in \mathbf{CFG}(\Sigma, \ell)$ and $A \in N$. We introduce a new nonterminal $B \notin N$ and set $G' = \langle N \cup \{B\}, P_1 \cup P_2 \cup P_3 \cup P_4, S \rangle$, where for each $p \in \text{Rule}(G')$, $P_i(p)$ ($1 \leq i \leq 4$) are defined as follows:

$$P_1(p) = \begin{cases} 0, & \text{if } p \notin \text{Rule}(G) \text{ or } p = A \rightarrow A\alpha \text{ for some } \alpha \in (N \cup \Sigma)^*, \\ P(p), & \text{otherwise,} \end{cases}$$

$$P_2(p) = \begin{cases} P(A \rightarrow \beta), & \text{if } p = A \rightarrow \beta B \text{ for some } \beta \in (N \cup \Sigma)^* \\ & \text{with the leftmost symbol of } \beta \neq A, \\ 0, & \text{otherwise,} \end{cases}$$

$$P_3(p) = \begin{cases} P(A \rightarrow A\alpha), & \text{if } p = B \rightarrow \alpha \text{ for some } \alpha \in (N \cup \Sigma)^*, \\ 0, & \text{otherwise} \end{cases}$$

and

$$P_4(p) = \begin{cases} P(A \rightarrow A\alpha), & \text{if } p = B \rightarrow \alpha B \text{ for some } \alpha \in (N \cup \Sigma)^*, \\ 0, & \text{otherwise.} \end{cases}$$

Then $G' \in \mathbf{CFG}(\Sigma, \ell)$ and $L^{[D]}(G') = L^{[D]}(G)$. Note that all productions of the form $A \rightarrow A\alpha$ do not appear in $\text{supp}(P_1 \cup P_2 \cup P_3 \cup P_4)$.

The proofs of the above two facts follows the intuition behind Lemmas 4.3 and 4.4 in [37], with the technique used in the proofs of Lemma 4.4 and Theorem 4.6. The details are omitted here. We only point out that commutativity and associativity of \wedge in ℓ are needed, but distributivity of \wedge over \vee is not required in the proofs.

Now we are ready to prove the conclusion of this theorem. It is required to show that for each $G \in \mathbf{CFG}(\Sigma, \ell)$, there exists a GNF G' with $[L^{[D]}(G) \equiv A] \leq [L^{[D]}(G') \equiv A - \{\varepsilon\}]$. Using Theorem 4.6, it suffices to show that for each CNF G we can find a GNF G' with $L^{[D]}(G') = L^{[D]}(G)$. The above two facts allows us to construct G' from G according to the procedure outlined in the proof of Theorem 4.6 in [37]. Since the construction is almost the same as that given in [37], here we are not going to describe such a procedure in detail. \square

It is worth noting that the generalizations of Chomsky normal form theorem and Greibach normal form theorem in quantum logic are straightforward when we comply with the depth-first principle.

We now turn to consider the second way of defining the language generated by an ℓ -valued CFG, namely, the width-first way. For any $G = \langle N, P, S \rangle \in \mathbf{CFG}(\Sigma, \ell)$,

the direct derivation relation \xRightarrow{G} is defined to be an ℓ -valued binary relation on $(N \cup \Sigma)^*$, and for any $\alpha, \beta \in (N \cup \Sigma)^*$,

$$\alpha \xRightarrow{G} \beta \stackrel{def}{=} (\exists A, \alpha_1, \alpha_2, \gamma)(\alpha = \alpha_1 A \alpha_2 \wedge \beta = \alpha_1 \gamma \alpha_2 \wedge A \rightarrow \gamma \in P).$$

Furthermore, we define

$$\begin{aligned} \xRightarrow{G}^0 &= Id_{(N \cup \Sigma)^*}, \\ \xRightarrow{G}^{n+1} &= \xRightarrow{G}^n \circ \xRightarrow{G} \text{ for any } n \geq 0 \end{aligned}$$

and

$$\xRightarrow{G}^* = \bigcup_{n=0}^{\infty} \xRightarrow{G}^n.$$

Definition 4.8. Let G be an ℓ -valued CFG over Σ . Then the ℓ -valued language $L^{[W]}(G) \in L^{\Sigma^*}$ generated by G in the width-first way is defined by

$$w \in L^{[W]}(G) \stackrel{def}{=} S \xRightarrow{G}^* w$$

for each $w \in \Sigma^*$.

The following lemma clarifies the relationship between the languages generated by an orthomodular lattice-valued context-free grammar in the depth-first way and the width-first way.

Lemma 4.9. Let $\Sigma \neq \emptyset$ be a finite alphabet.

(1) For any $G \in \mathbf{CFG}(\Sigma, \ell)$ and $w \in \Sigma^*$, we have:

$$\models^{\ell} w \in L^{[D]}(G) \rightarrow w \in L^{[W]}(G).$$

(2) The following two statements are equivalent:

(2.1) ℓ is a Boolean algebra;

(2.2) for any $G \in \mathbf{CFG}(\Sigma, \ell)$ and $w \in \Sigma^*$,

$$\models^{\ell} w \in L^{[D]}(G) \leftrightarrow w \in L^{[W]}(G).$$

(3) For any $G = \langle N, P, S \rangle \in \mathbf{CFG}(\Sigma, \ell)$ and $w \in \Sigma^*$, we have:

$$\models^{\ell} \gamma(\text{atom}(G)) \wedge w \in L^{[W]}(G) \rightarrow w \in L^{[D]}(G),$$

and in particular if $\rightarrow = \rightarrow_3$ then it holds that

$$\models^{\ell} \gamma(\text{atom}(G)) \rightarrow (w \in L^{[D]}(G) \leftrightarrow w \in L^{[W]}(G))$$

where $atom(G)$ is the set of atomic propositions about G , that is, $atom(G) = \{A \rightarrow \alpha \in P : A \rightarrow \alpha \in \text{supp}P\}$.

Proof. (1) Let $G = \langle N, P, S \rangle$. We first prove the following:

Claim 1. For any $d = ((\alpha_0, \alpha_1, \dots, \alpha_n), (p_1, \dots, p_n)) \in QDer_G(d)$, $[Der_G(d)] \leq [\alpha_0 \xrightarrow[n]{G} \alpha_n]$.

We proceed by induction on n . For the case of $n = 1$, since p_1 is compatible with $\alpha_0 \Rightarrow \alpha_1$, it holds that $\alpha_0 = \beta_1 A \beta_2$, $\alpha_1 = \beta_1 \gamma \beta_2$ and $p_1 = A \rightarrow \gamma$ for some A, β_1, β_2 and γ . This leads to $[Der_G(d)] = P(p_1) \leq [\alpha_0 \xrightarrow[1]{G} \alpha_1]$. In general, we may assume that $\alpha_{n-1} = \beta_1 A \beta_2$, $\alpha_n = \beta_1 \gamma \beta_2$, $p_n = A \rightarrow \gamma$ for some $A, \beta_1, \beta_2, \gamma$ because p_n is compatible with $\alpha_{n-1} \Rightarrow \alpha_n$. This yields $P(p_n) \leq [\alpha_{n-1} \xrightarrow[G]{G} \alpha_n]$. On the other hand, by the induction hypothesis we assert that

$$\bigwedge_{i=1}^{n-1} P(p_i) = [Der_G((\alpha_0, \alpha_1, \dots, \alpha_{n-1}), (p_1, \dots, p_{n-1}))] \leq [\alpha_0 \xrightarrow[n-1]{G} \alpha_{n-1}].$$

Therefore, we have

$$[Der_G(d)] = \bigwedge_{i=1}^n P(p_i) \leq [\alpha_0 \xrightarrow[n-1]{G} \alpha_{n-1}] \wedge [\alpha_{n-1} \xrightarrow[G]{G} \alpha_n] \leq [\alpha_0 \xrightarrow[n]{G} \alpha_n].$$

Now using the above claim we obtain $[Der_G(d)] \leq [S \xrightarrow[G]{*} w] = L^{[W]}(G)(w)$ for any $d \in QDer_G$ with $I(d) = S$ and $L(d) = w$. This implies $L^{[D]}(G)(w) \leq L^{[W]}(G)(w)$ for every $w \in \Sigma^*$.

(2) The implication from (2.1) to (2.2) is a direct corollary of (3). So here we only prove that (2.2) implies (2.1). Assume that (2.2) is valid. We are going to show that $\lambda \wedge (\mu_1 \vee \mu_2) = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2)$ for all $\lambda, \mu_1, \mu_2 \in L$. Suppose that $a \in \Sigma$. Let $G = \langle \{S, A, B\}, P, S \rangle$ where $P(S \rightarrow A) = \lambda$, $P(A \rightarrow B) = \mu_1$, $P(A \rightarrow C) = \mu_2$, $P(B \rightarrow a) = P(C \rightarrow a) = 1$, and $P(p) = 0$ for other productions $p \in Rule(G)$. A simple calculation yields $L^{[D]}(G)(a) = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2)$ and $L^{[W]}(G)(a) = \lambda \wedge (\mu_1 \vee \mu_2)$. Then we know from (2.2) that $L^{[D]}(G)(a) = L^{[W]}(G)(a)$, and it is done.

(3) We first prove the following:

Claim 2. For any $\alpha, \beta \in (N \cup \Sigma)^*$, and for any $n \geq 0$,

$$[\gamma(atom(G))] \wedge [\alpha \xrightarrow[n]{G} \beta] \leq \bigvee \{ [Der_G(d)] : d \in QDer_G, I(d) = \alpha \text{ and } L(d) = \beta \}.$$

We use induction on n . First, for any A, α_1, α_2 and γ with $\alpha = \alpha_1 A \alpha_2$ and $\beta = \alpha_1 \gamma \alpha_2$, we set $p = A \rightarrow \gamma$. It is clear that p is compatible with $\alpha \Rightarrow \beta$. Put $d_{A, \alpha_1, \alpha_2, \gamma} = ((\alpha, \beta), p)$. Then $d_{A, \alpha_1, \alpha_2, \gamma} \in QDer_G$, $I(d_{A, \alpha_1, \alpha_2, \gamma}) = \alpha$, $L(d_{A, \alpha_1, \alpha_2, \gamma}) =$

β , and $\lceil Der_G(d_{A,\alpha_1,\alpha_2,\gamma}) \rceil = P(p) = P(A \rightarrow \gamma)$. Thus,

$$\begin{aligned} \lceil \gamma(\text{atom}(G)) \rceil \wedge \lceil \alpha \xrightarrow[G]{\frac{1}{G}} \beta \rceil &\leq \lceil \alpha \xrightarrow[G]{\frac{1}{G}} \beta \rceil = \bigvee_{\alpha=\alpha_1 A \alpha_2, \beta=\alpha_1 \gamma \alpha_2} P(A \rightarrow \gamma) \\ &= \bigvee_{\alpha=\alpha_1 A \alpha_2, \beta=\alpha_1 \gamma \alpha_2} \lceil Der_G(d_{A,\alpha_1,\alpha_2,\gamma}) \rceil \\ &\leq \bigvee \{ \lceil Der_G(d) \rceil : d \in QDer_G, I(d) = \alpha \text{ and } L(d) = \beta \}. \end{aligned}$$

The conclusion is valid for the case of $n = 1$. In general, it follows from Lemmas 2.6 and 2.7 that

$$\begin{aligned} \lceil \gamma(\text{atom}(G)) \rceil \wedge \lceil \alpha \xrightarrow[G]{\frac{n}{G}} \beta \rceil &= \lceil \gamma(\text{atom}(G)) \rceil \wedge \bigvee_{\delta} (\lceil \alpha \xrightarrow[G]{\frac{n-1}{G}} \delta \rceil \wedge \lceil \delta \xrightarrow[G]{\Rightarrow} \beta \rceil) \\ &= \lceil \gamma(\text{atom}(G)) \rceil \wedge \lceil \gamma(\text{atom}(G)) \rceil \wedge \bigvee_{\delta} (\lceil \alpha \xrightarrow[G]{\frac{n-1}{G}} \delta \rceil \wedge \lceil \delta \xrightarrow[G]{\Rightarrow} \beta \rceil) \\ &\leq \bigvee_{\delta} (\lceil \gamma(\text{atom}(G)) \rceil \wedge \lceil \alpha \xrightarrow[G]{\frac{n-1}{G}} \delta \rceil \wedge \lceil \delta \xrightarrow[G]{\Rightarrow} \beta \rceil). \end{aligned}$$

Therefore, we only need to show that

$$\lceil \gamma(\text{atom}(G)) \rceil \wedge \lceil \alpha \xrightarrow[G]{\frac{n-1}{G}} \delta \rceil \wedge \lceil \delta \xrightarrow[G]{\Rightarrow} \beta \rceil \leq \bigvee \{ \lceil Der_G(d) \rceil : d \in QDer_G, I(d) = \alpha \text{ and } L(d) = \beta \}$$

for each $\delta \in (N \cup \Sigma)^*$. By the induction hypothesis we obtain:

$$\begin{aligned} \lceil \gamma(\text{atom}(G)) \rceil \wedge \lceil \alpha \xrightarrow[G]{\frac{n-1}{G}} \delta \rceil \wedge \lceil \delta \xrightarrow[G]{\Rightarrow} \beta \rceil &\leq \lceil \gamma(\text{atom}(G)) \rceil \wedge \bigvee \{ \lceil Der_G(d') \rceil : d' \in QDer_G, \\ &\quad I(d') = \alpha \text{ and } L(d') = \delta \} \wedge \lceil \delta \xrightarrow[G]{\Rightarrow} \beta \rceil \\ &= \lceil \gamma(\text{atom}(G)) \rceil \wedge \bigvee \{ \lceil Der_G(d') \rceil : d' \in QDer_G, I(d') = \alpha \text{ and } L(d') = \delta \} \\ &\quad \wedge \bigvee \{ P(A \rightarrow \gamma) : \delta = \alpha_1 A \alpha_2 \text{ and } \beta = \alpha_1 \gamma \alpha_2 \} \\ &\leq \bigvee \{ \lceil Der_G(d') \rceil \wedge P(A \rightarrow \gamma) : d' \in QDer_G, I(d') = \alpha, L(d') = \delta = \alpha_1 A \alpha_2 \text{ and } \beta = \alpha_1 \gamma \alpha_2 \}. \end{aligned}$$

Then it suffices to show that for any $d', A, \alpha_1, \alpha_2$ and γ with $d' \in QDer_G, I(d') = \alpha, L(d') = \delta = \alpha_1 A \alpha_2$ and $\beta = \alpha_1 \gamma \alpha_2$,

$$\lceil Der_G(d') \rceil \wedge P(A \rightarrow \gamma) \leq \bigvee \{ \lceil Der_G(d) \rceil : d \in QDer_G, I(d) = \alpha \text{ and } L(d) = \beta \}.$$

To this end, suppose $d' = ((\theta_0, \theta_1, \dots, \theta_k), (q_1, \dots, q_k))$. Let $d_0 = ((\theta_0, \theta_1, \dots, \theta_k = \delta, \beta), (q_1, \dots, q_k, A \rightarrow \gamma))$. Clearly, $d_0 \in QDer_G, I(d_0) = \theta_0 = I(d') = \alpha, L(d_0) = \beta$ and $\lceil Der_G(d_0) \rceil = \lceil Der_G(d') \rceil \wedge P(A \rightarrow \gamma)$. This concludes the proof of the above claim.

Finally, with the above claim we assert that

$$\lceil \gamma(\text{atom}(G)) \rceil \wedge \lceil S \xrightarrow[G]{\frac{n}{G}} w \rceil \leq \bigvee \{ \lceil Der_G(d) \rceil : d \in QDer_G, I(d) = S \text{ and } L(d) = w \} = L^{(D)}(G)(w)$$

for all $w \in \Sigma^*$ and $n \geq 1$. Note that $\lceil S \xrightarrow[G]{n} w \rceil = 0$. This together with Lemmas 2.6 and 2.7 yields:

$$\begin{aligned}
\lceil \gamma(\text{atom}(G)) \rceil \wedge L^{[W]}(G)(w) &= \lceil \gamma(\text{atom}(G)) \rceil \wedge \lceil S \xrightarrow[G]{*} w \rceil \\
&= \lceil \gamma(\text{atom}(G)) \rceil \wedge \bigvee_{n=0}^{\infty} \lceil S \xrightarrow[G]{n} w \rceil \\
&= \lceil \gamma(\text{atom}(G)) \rceil \wedge \lceil \gamma(\text{atom}(G)) \rceil \wedge \bigvee_{n=0}^{\infty} \lceil S \xrightarrow[G]{n} w \rceil \\
&= \bigvee_{n=0}^{\infty} (\lceil \gamma(\text{atom}(G)) \rceil \wedge \lceil S \xrightarrow[G]{n} w \rceil) \leq L^{[D]}(G)(w). \quad \square
\end{aligned}$$

The notion of ℓ -valued language generated in the width-first way allows us to introduce a new definition of ℓ -valued context-freeness of languages. Notice that in evaluating the language generated by an ℓ -valued CFG according to Definition 4.8, a join operation appear at the outermost stratum, some meet operations then appear at the second stratum, and so on. The meet and join operations are entangled heavily. Thus, distributivity is highly anticipated, and a commutative version of context-freeness should be much more convenient when we comply with the width-first principle.

Definition 4.10. Let Σ be a finite alphabet. Then ℓ -valued (unary) predicate $CCFL_{\Sigma}^{[W]}$ on ℓ -valued languages over Σ is defined by

$$CCFL_{\Sigma}^{[W]}(A) \stackrel{def}{=} (\exists G \in \mathbf{CFG}(\Sigma, \ell)) (\gamma(\text{atom}(G) \cup r(A)) \wedge A \equiv L^{[W]}(G))$$

for each $A \in L^{\Sigma^*}$, where $\text{atom}(G)$ is as in Lemma 4.9, $r(A) = \{\mathbf{a} : a = A(s) \text{ for some } s \in \Sigma^*\}$, and \mathbf{a} is the nullary predicate corresponding to element a in L . The intuitive interpretation of $CCFL_{\Sigma}^{[W]}$ is “commutative context-freeness in the width-first way.”

Combining Lemma 4.9 with Theorems 4.6 and 4.7, we are able to establish Chomsky normal form theorem and Greibach normal form theorem in the sense of preserving the language generated by an ℓ -valued CFG in the width-first way.

Theorem 4.11. For any ℓ -valued language A over Σ , we have:

- (1) $\models^{\ell} CCFL_{\Sigma}^{[W]}(A) \rightarrow (\exists \text{CNF } G)(L^{[W]}(G) \equiv A - \{\varepsilon\});$
- (2) $\models^{\ell} CCFL_{\Sigma}^{[W]}(A) \rightarrow (\exists \text{GNF } G)(L^{[W]}(G) \equiv A - \{\varepsilon\}).$

Proof. (1) For any $G' \in \mathbf{CFG}(\Sigma, \ell)$, with Lemma 4.9 we have

$$\lceil \gamma(\text{atom}(G')) \rceil \leq \lceil L^{[W]}(G') \equiv L^{[D]}(G') \rceil.$$

Consequently, we obtain:

$$\begin{aligned} \lceil \gamma(\text{atom}(G') \cup r(A)) \rceil \wedge \lceil A \equiv L^{[W]}(G') \rceil &\leq \lceil \gamma(\text{atom}(G') \cup r(A)) \rceil \wedge \\ &\quad \lceil A \equiv L^{[W]}(G') \rceil \wedge \lceil L^{[W]}(G') \equiv L^{[D]}(G') \rceil \\ &\leq \lceil A \equiv L^{[D]}(G') \rceil \end{aligned}$$

by using Lemma 2.12(3). On the other hand, we know from the proofs of Lemma 4.4 and Theorem 4.6 that there is a CNF G with $L^{[D]}(G) = L^{[D]}(G') - \{\varepsilon\}$. This implies $\lceil L^{[D]}(G) \equiv L^{[D]}(G') - \{\varepsilon\} \rceil = 1$. Furthermore, from the construction of such an ℓ -valued grammar G given in the proofs of Lemma 4.4 and Theorem 4.6 we assert that

$$\lceil \gamma(\text{atom}(G') \cup r(A)) \rceil \leq \lceil \gamma(\text{atom}(G') \cup \text{atom}(G) \cup r(A)) \rceil$$

by Lemma 2.7. By combining the above conclusions and using Lemma 2.12(3) we obtain:

$$\begin{aligned} \lceil \gamma(\text{atom}(G') \cup r(A)) \rceil \wedge \lceil A \equiv L^{[W]}(G') \rceil &\leq \lceil \gamma(\text{atom}(G') \cup \text{atom}(G) \cup r(A)) \rceil \wedge \\ &\quad \lceil A \equiv L^{[D]}(G') \rceil \wedge \lceil L^{[D]}(G) \equiv L^{[D]}(G') - \{\varepsilon\} \rceil \\ &\leq \lceil \gamma(\text{atom}(G') \cup \text{atom}(G) \cup r(A)) \rceil \wedge \lceil A - \{\varepsilon\} \\ &\quad \equiv L^{[D]}(G') - \{\varepsilon\} \rceil \wedge \lceil L^{[D]}(G') - \{\varepsilon\} \equiv L^{[D]}(G) \rceil \\ &\leq \lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \lceil L^{[D]}(G) \equiv A - \{\varepsilon\} \rceil. \end{aligned}$$

Finally, using Lemma 4.9 once again we have:

$$\lceil \gamma(\text{atom}(G)) \rceil \leq \lceil L^{[W]}(G) \equiv L^{[D]}(G) \rceil.$$

Then it follows from Lemma 2.12(3) that

$$\begin{aligned} \lceil \gamma(\text{atom}(G') \cup r(A)) \rceil \wedge \lceil A \equiv L^{[W]}(G') \rceil &\leq \lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \lceil L^{[D]}(G) \equiv A - \{\varepsilon\} \rceil \wedge \\ &\quad \lceil L^{[W]}(G) \equiv L^{[D]}(G) \rceil \\ &\leq \lceil L^{[W]}(G) \equiv A - \{\varepsilon\} \rceil. \end{aligned}$$

(2) is similar to (1). \square

4.2. Basic Definitions of Orthomodular Lattice-Valued Pushdown Automata

The construction of pushdown automata is similar to that of finite automata, with a control of both input tape and a stack at the top of which symbols may be entered or removed. Here for convenience of the reader we briefly recall the formal definition of pushdown automata. For more detailed exposition we refer to [37, 41].

Let Σ be a finite input alphabet. A (nondeterministic) pushdown automaton over Σ is a 6-tuple $\mathfrak{R} = \langle Q, \Gamma, \delta, q_0, Z_0, F \rangle$, where

- (i) Q is a finite set of states;
- (ii) Γ is a finite set of stack symbols, called the stack alphabet;
- (iii) $q_0 \in Q$ is the initial state;
- (iv) $Z_0 \in \Gamma$ is the start stack symbol;
- (v) $F \subseteq Q$ is the set of final states; and

(vi) δ is a finite subset of $[Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma] \times (Q \times \Gamma^*)$, called the transition relation. Intuitively, if $p, q \in Q$, $a \in \Sigma$, $Z \in \Gamma$ and $\gamma \in \Gamma^*$, then $((p, a, Z), (q, \gamma)) \in \delta$ means that whenever the automaton is in state p , reading input symbol a on the input tape and Z on the top of the stack, it can enter state q , replace stack symbol Z by string γ of stack symbols, and advance its input head one symbol. For the case of $a = \varepsilon$, the same happens except that the input head is not advanced.

For a given pushdown automaton $\mathfrak{R} = \langle Q, \Gamma, \delta, q_0, Z_0, F \rangle$ over input alphabet Σ , each element $(q, w, \gamma) \in Q \times \Sigma^* \times \Gamma^*$ is called a configuration of \mathfrak{R} , where q , w and γ are used to record the current state, the part of the input yet unread, and the current stack content. The next configuration relation $\vdash_{\mathfrak{R}}$ between configurations is defined as follows: for any $p, q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $w \in \Sigma^*$, $Z \in \Gamma$ and $\alpha, \beta \in \Gamma^*$, if $((q, a, Z), (p, \beta)) \in \delta$ then we have

$$(q, aw, Z\alpha) \vdash_{\mathfrak{R}} (p, w, \beta\alpha).$$

The reflexive and transitive closure of $\vdash_{\mathfrak{R}}$ is denoted by $\vdash_{\mathfrak{R}}^*$. Then the language accepted by \mathfrak{R} with final states is defined to be

$$L_{FS}(\mathfrak{R}) = \{w \in \Sigma^* : (q_0, w, Z_0) \vdash_{\mathfrak{R}}^* (p, \varepsilon, \gamma) \text{ for some } p \in F \text{ and } \gamma \in \Gamma^*\},$$

and the language accepted by \mathfrak{R} with empty stack is defined to be

$$L_{ES}(\mathfrak{R}) = \{w \in \Sigma^* : (q_0, w, Z_0) \vdash_{\mathfrak{R}}^* (p, \varepsilon, \varepsilon) \text{ for some } p \in Q\}.$$

We now can introduce the orthomodular lattice-valued generalization of pushdown automaton. Let $\ell = \langle L, \leq, \wedge, \vee, \perp, 0, 1 \rangle$ be an orthomodular lattice, and Σ a finite input alphabet. An ℓ -valued pushdown automaton (PDA for short) over Σ is defined to be a 6-tuple $\mathfrak{R} = \langle Q, \Gamma, \delta, q_0, Z_0, F \rangle$, where

- (i) Q , Γ , q_0 and Z_0 are the same as in an ordinary pushdown automaton;
- (ii) F is an ℓ -valued subset of Q , and intuitively for each $q \in Q$, $T(q)$ is the truth value of the proposition that q is a final state; and
- (iii) δ is an ℓ -valued subset of

$$Rule(\mathfrak{R}) \stackrel{def}{=} [Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma] \times (Q \times \Gamma^*),$$

whose elements are called rules of \mathfrak{R} , such that $supp\delta = \{r \in Rule(\mathfrak{R}) : \delta(r) > 0\}$ is a finite set. If $r = ((q, a, Z), (p, \gamma)) \in Rule(\mathfrak{R})$, then intuitively $\delta(r)$ is the truth

value of the proposition that whenever the PDA is in state q , reading a on the input tape and Z on the top of the stack, it can enter state p , replace Z by string γ of stack symbols, and advance one symbol if $a \neq \varepsilon$.

We write $\mathbf{PDA}(\Sigma, \ell)$ for the (proper) class of ℓ -valued PDAs over Σ .

There are also two nonequivalent ways of defining the language accepted by an ℓ -valued PDA, the depth-first one and the width-first one, due to the fact that ℓ does not enjoy distributivity of \wedge over \vee . We first propose the definition in the depth-first way. Suppose that $\mathfrak{R} = \langle Q, \Gamma, \delta, q_0, Z_0, F \rangle$ is an ℓ -valued PDA over Σ .

We write $Con(\mathfrak{R})$ for the set of configurations of \mathfrak{R} , that is, $Con(\mathfrak{R}) \stackrel{def}{=} Q \times \Sigma^* \times \Gamma^*$. For any $r = ((q, a, Z), (p, \gamma)) \in Rule(\mathfrak{R})$, and $C_1 = (q', w, \gamma_1), C_2 = (p', u, \gamma_2) \in Con(\mathfrak{R})$, we say that r is compatible with the next configuration relation $C_1 \vdash C_2$ if $q' = q, p' = p, \gamma_1 = Z\gamma'$ and $\gamma_2 = \gamma\gamma'$ for some $\gamma' \in \Gamma^*$, and

- (a) $a = \varepsilon$ and $u = w$; or
- (b) $a \in \Sigma$ and $w = au$.

A quasi-path (of length $n \geq 0$) is an element

$$c = ((C_0, C_1, \dots, C_n), (r_1, \dots, r_n)) \in Con(\mathfrak{R})^{n+1} \times Rule(\mathfrak{R})^n$$

such that r_i is compatible with $C_{i-1} \vdash C_i$ for each $i \leq n$. The set of quasi-paths in \mathfrak{R} is denoted by $QPath_{\mathfrak{R}}$. Then the ℓ -valued (unary) predicate “to be a computational path in \mathfrak{R} ” is defined to be $Path_{\mathfrak{R}} \in L^{QPath_{\mathfrak{R}}}$ with

$$Path_{\mathfrak{R}}((C_0, C_1, \dots, C_n), (r_1, \dots, r_n)) \stackrel{def}{=} \bigwedge_{i=1}^n (r_i \in \delta)$$

for any $c = ((C_0, C_1, \dots, C_n), (r_1, \dots, r_n)) \in QPath_{\mathfrak{R}}$, and $n \geq 0$.

Definition 4.12. Let $\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)$. Then the recognizability $rec_{\mathfrak{R}}^{[D,FS]}$ of \mathfrak{R} by final stats in the depth-first way is defined to be an ℓ -valued (unary) predicate on Σ^* , that is, $rec_{\mathfrak{R}}^{[D,FS]} \in L^{\Sigma^*}$:

$$rec_{\mathfrak{R}}^{[D,FS]}(w) \stackrel{def}{=} (\exists c \in QPath_{\mathfrak{R}})(Path_{\mathfrak{R}}(c) \wedge I(c) = (q_0, w, Z_0) \wedge Li(c) = \varepsilon \wedge Lq(c) \in F)$$

for every $w \in \Sigma^*$, where $I(c)$ stands for the first configuration in c , and $Li(c)$ and $Lq(c)$ are respectively the unread content on the input tape and the state of the last configuration in c . Similarly, the recognizability $rec_{\mathfrak{R}}^{[D,ES]} \in L^{\Sigma^*}$ of \mathfrak{R} by empty stack in the depth-first way is defined by

$$rec_{\mathfrak{R}}^{[D,ES]}(w) \stackrel{def}{=} (\exists c \in QPath_{\mathfrak{R}})(Path_{\mathfrak{R}} \wedge I(c) = (q_0, w, Z_0) \wedge Li(c) = \varepsilon \wedge Ls(c) = \varepsilon)$$

for every $w \in \Sigma^*$, where $Ls(c)$ is the stack content of the last configuration in c .

The following theorem shows that the in the depth-first way the two notions of recognizability by final states and empty stack are indeed equivalent.

Theorem 4.13. For any $\mathfrak{R}_1 \in \mathbf{PDA}(\Sigma, \ell)$, there is $\mathfrak{R}_2 \in \mathbf{PDA}(\Sigma, \ell)$ such that for any $w \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}_1}^{[D,FS]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}_2}^{[D,ES]}(w);$$

and conversely for any $\mathfrak{R}_1 \in \mathbf{PDA}(\Sigma, \ell)$, there is $\mathfrak{R}_2 \in \mathbf{PDA}(\Sigma, \ell)$ such that for any $w \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}_1}^{[D,ES]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}_2}^{[D,FS]}(w).$$

Proof. We only prove the first part, and the second part is similar. The idea of the proof is the same as in the case of ordinary pushdown automata (see [37], pages 114-115). Suppose that $\mathfrak{R}_1 = \langle Q, \Gamma, \delta, q_0, Z_0, F \rangle$. Then we set $\mathfrak{R}_2 = \langle Q \cup \{q'_0, q_e\}, \Gamma \cup \{Z'_0\}, \delta', q'_0, Z'_0, \emptyset \rangle$, where

$$\delta'(r) = \begin{cases} \delta(r), & \text{if } r \in \text{Rule}(\mathfrak{R}_1), \\ 1, & \text{if } r = ((q'_0, \varepsilon, Z'_0), (q_0, Z_0 Z'_0)) \text{ or } r = ((q_e, \varepsilon, Z), (q_e, \varepsilon)) \text{ for some } Z \in \Gamma \cup \{\varepsilon\}, \\ F(q), & \text{if } r = ((q, \varepsilon, Z), (q_e, Z)) \text{ for some } q \in Q \text{ and } Z \in \Gamma \cup \{\varepsilon\}, \\ 0, & \text{otherwise} \end{cases}$$

for any $r \in \text{Rule}(\mathfrak{R}_2)$.

Now for each $c = ((C_0, C_1, \dots, C_n), (r_1, \dots, r_n)) \in Q\text{Path}_{\mathfrak{R}_1}$ with $I(c) = C_0 = (q_0, w, Z_0)$ and $Li(c) = \varepsilon$, we assume that $C_i = (q_i, w_i, \gamma_i)$ for every $1 \leq i \leq n$, and write $C'_{-1} = (q'_0, w, Z'_0)$, $C'_0 = (q_0, w, Z_0 Z'_0)$, $C'_i = (q_i, w_i, \gamma_i Z'_0)$ for any $1 \leq i \leq n$, $C'_{n+k} = (q_e, \varepsilon, \gamma'_k)$ for any $1 \leq k \leq |\gamma_n| + 1$, where γ'_k is the string consisting of the last $|\gamma_n| - k + 1$ symbols of γ_n . Furthermore, we put

$$\text{ext}(c) = ((C'_{-1}, C'_0, C'_1, \dots, C'_n, C'_{n+1}, \dots, C'_{n+|\gamma_n|+1}), (r_0, r_1, \dots, r_n, r_{n+1}, \dots, r_{n+|\gamma_n|+1})),$$

where $r_0 = ((q'_0, \varepsilon, Z'_0), (q_0, Z_0 Z'_0))$, $r_{n+1} = ((q_n, \varepsilon, Z_n), (q_e, Z_n))$, Z_n is the first symbol of γ_n , $r_{n+k} = ((q_e, \varepsilon, Z'_k), (q_e, \varepsilon))$, and Z'_k is the first symbol of γ'_{k-1} for all $2 \leq k \leq |\gamma_n| + 1$. Then it is easy to see that $\text{ext}(c) \in Q\text{Path}_{\mathfrak{R}_2}$ and

$$[\text{Path}_{\mathfrak{R}_2}(\text{ext}(c))] = [\text{Path}_{\mathfrak{R}_1}(c)] \wedge F(Lq(c)).$$

Therefore, it follows that

$$\begin{aligned} [\text{rec}_{\mathfrak{R}_1}^{[D,FS]}(w)] &= \bigvee \{ [\text{Path}_{\mathfrak{R}_1}(c)] \wedge F(Lq(c)) : c \in Q\text{Path}_{\mathfrak{R}_1}, I(c) = (q_0, w, Z_0) \text{ and } Li(c) = \varepsilon \} \\ &= \bigvee \{ [\text{Path}_{\mathfrak{R}_2}(\text{ext}(c))] : c \in Q\text{Path}_{\mathfrak{R}_1}, I(c) = (q_0, w, Z_0) \text{ and } Li(c) = \varepsilon \} \\ &\leq \bigvee \{ [\text{Path}_{\mathfrak{R}_2}(c')] : c' \in Q\text{Path}_{\mathfrak{R}_2}, I(c') = (q'_0, w, Z'_0) \text{ and } Li(c') = Ls(c') = \varepsilon \} \\ &= [\text{rec}_{\mathfrak{R}_2}^{[D,ES]}(w)]. \end{aligned}$$

Likewise, we are able to show that $[\text{rec}_{\mathfrak{R}_2}^{[D,ES]}(w)] \leq [\text{rec}_{\mathfrak{R}_1}^{[D,FS]}(w)]$. \square

We now turn to present the definition of language accepted by an ℓ -valued PDA according to the width-first principle and compare it with that given according to the depth-first principle. To this end, we first introduce an auxiliary notation. For any $\alpha, \beta \in \Gamma^*$, if there exists $\gamma \in \Gamma^*$ such that $\alpha = \gamma\beta$, then β is called a tail of α and we write $\beta \leq \alpha$. In this case, γ is unique and we define $\gamma = \alpha - \beta$. Then the next configuration relation $\vdash_{\mathfrak{R}}$ is defined to be an ℓ -valued binary relation between configurations:

$$(p, v, \alpha) \vdash_{\mathfrak{R}} (q, w, \beta) \stackrel{def}{=} [w = t(v) \wedge t(\alpha) \leq \beta \wedge ((p, h(v), h(\alpha)), (q, \beta - t(\alpha))) \in \delta] \\ \vee [w = v \wedge t(\alpha) \leq \beta \wedge ((p, \varepsilon, h(\alpha)), (q, \beta - t(\alpha))) \in \delta]$$

for all $p, q \in Q$, $v, w \in \Sigma^*$, and $\alpha, \beta \in \Gamma^*$, where for each string x of symbols, $h(x)$ and $t(x)$ stand for the head and tail of x , respectively, that is, $h(x) = X_1$ and $t(x) = X_2 \dots X_n$ if $x = X_1 X_2 \dots X_n$. Furthermore, let $\vdash_{\mathfrak{R}}^*$ be the reflexive and transitive closure of $\vdash_{\mathfrak{R}}$, that is,

$$\vdash_{\mathfrak{R}}^* = \bigcup_{n=0}^{\infty} \vdash_{\mathfrak{R}}^n,$$

where

$$\vdash_{\mathfrak{R}}^0 = Id_{Con(\mathfrak{R})}$$

and

$$\vdash_{\mathfrak{R}}^{n+1} = \vdash_{\mathfrak{R}} \circ \vdash_{\mathfrak{R}}^n \text{ for any } n \geq 0.$$

With this notation we can present the definition of language accepted by an ℓ -valued PDA according to the width-first principle in a way similar to the classical case.

Definition 4.14. Let $\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)$. Then the recognizability of \mathfrak{R} by final states in the width-first way is an ℓ -valued (unary) predicate $rec_{\mathfrak{R}}^{[W,FS]} \in L^{\Sigma^*}$, and it is defined by

$$rec_{\mathfrak{R}}^{[W,FS]}(w) \stackrel{def}{=} (\exists q \in Q, \gamma \in \Gamma^*) ((q_0, w, Z_0) \vdash_{\mathfrak{R}}^* (q, \varepsilon, \gamma) \wedge q \in F)$$

for each $w \in \Sigma^*$, and the recognizability $rec_{\mathfrak{R}}^{[W,ES]} \in L^{\Sigma^*}$ of \mathfrak{R} by empty stack in the width-first way is defined by

$$rec_{\mathfrak{R}}^{[W,FS]}(w) \stackrel{def}{=} (\exists q \in Q) (q_0, w, Z_0) \vdash_{\mathfrak{R}}^* (q, \varepsilon, \varepsilon)$$

for all $w \in \Sigma^*$.

The following lemma carefully compare recognizability of an orthomodular lattice-valued pushdown automaton in the depth-first way with that in the width-first way.

Lemma 4.15. Let Σ be a nonempty finite alphabet.

(1) For any $\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)$ and for any $w \in \Sigma^*$, we have:

$$\models^\ell \text{rec}_{\mathfrak{R}}^{[D,FS]}(w) \rightarrow \text{rec}_{\mathfrak{R}}^{[W,FS]}(w) \text{ and } \models^\ell \text{rec}_{\mathfrak{R}}^{[D,ES]}(w) \rightarrow \text{rec}_{\mathfrak{R}}^{[W,ES]}(w).$$

(2) The following three statements are equivalent:

(2.1) ℓ is a Boolean algebra;

(2.2) for any $\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)$ and $w \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}}^{[D,FS]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[W,FS]}(w);$$

(2.3) for any $\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)$ and $w \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R}}^{[D,ES]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[W,ES]}(w).$$

(3) For any $\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)$ and for any $w \in \Sigma^*$, we have:

$$\models^\ell \gamma(\text{atom}(\mathfrak{R})) \wedge \text{rec}_{\mathfrak{R}}^{[W,FS]}(w) \rightarrow \text{rec}_{\mathfrak{R}}^{[D,FS]}(w),$$

$$\models^\ell \gamma(\text{atom}(\mathfrak{R})) \wedge \text{rec}_{\mathfrak{R}}^{[W,ES]}(w) \rightarrow \text{rec}_{\mathfrak{R}}^{[D,ES]}(w).$$

In particular, if $\rightarrow = \rightarrow_3$ then it holds that

$$\models^\ell \gamma(\text{atom}(\mathfrak{R})) \rightarrow (\text{rec}_{\mathfrak{R}}^{[D,FS]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[W,FS]}(w)),$$

$$\models^\ell \gamma(\text{atom}(\mathfrak{R})) \rightarrow (\text{rec}_{\mathfrak{R}}^{[D,ES]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[W,ES]}(w)),$$

where $\text{atom}(\mathfrak{R})$ is the set of atomic propositions concerning pushdown automaton \mathfrak{R} , that is, $\text{atom}(\mathfrak{R}) = \{q \in F : q \in Q\} \cup \{\text{rule } r \in \delta : r \in \text{supp}\delta\}$.

Proof. This is similar to the proof of Lemma 4.9. We only give an outline instead of the details.

(1) The key step is to prove that $\lceil \text{Path}_{\mathfrak{R}}(c) \rceil \leq \lceil I(c) \vdash_{\mathfrak{R}}^* L(c) \rceil$ for every $c \in Q\text{Path}_{\mathfrak{R}}$. This can be done by induction on the length $|c|$ of c .

(2) It is immediate from (3) that (2.1) implies both (2.2) and (2.3). Conversely, for any $\lambda, \mu_1, \mu_2 \in L$, let $Q = \{q_0, q_1, q_2, q_3, q_4\}$, $a \in \Sigma \neq \emptyset$, $\Gamma = \{Z_0, Z_1, Z_2, Z_4\}$ and $\mathfrak{R} = \langle Q, \Gamma, \delta, q_0, Z_0, \{q_4\} \rangle$, where $\delta((q_0, a, Z_0), (q_1, Z_1)) = \lambda$, $\delta((q_1, a, Z_1), (q_2, Z_2)) = \mu_2$, $\delta((q_1, a, Z_1), (q_3, Z_3)) = \mu_2$, $\delta((q_2, a, Z_2), (q_4, \varepsilon)) = \delta((q_3, a, Z_3), (q_4, \varepsilon)) = 1$, and $\delta(r) = 0$ for all other rules $r \in \text{Rule}(\mathfrak{R})$. Then we have:

$$\lceil \text{rec}_{\mathfrak{R}}^{[D,FS]}(a^3) \rceil = \lceil \text{rec}_{\mathfrak{R}}^{[D,ES]}(a^3) \rceil = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2)$$

and

$$\lceil \text{rec}_{\mathfrak{R}}^{[W,FS]}(a^3) \rceil = \lceil \text{rec}_{\mathfrak{R}}^{[W,ES]}(a^3) \rceil = \lambda \wedge (\mu_1 \vee \mu_2),$$

and $\lambda \wedge (\mu_1 \vee \mu_2) = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2)$ follows from either (2.2) or (2.3).

(3) We can prove that

$$\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil C_1 \vdash_{\mathfrak{R}}^n C_2 \rceil \leq \bigvee \{ \lceil \text{Path}_{\mathfrak{R}}(c) \rceil : c \in Q\text{Path}_{\mathfrak{R}}, I(c) = C_1 \text{ and } L(c) = C_2 \}$$

for any $C_1, C_2 \in \text{Con}(\mathfrak{R})$ by induction on n . Then

$$\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil \text{rec}_{\mathfrak{R}}^{[W,FS]}(w) \rceil \leq \lceil \text{rec}_{\mathfrak{R}}^{[D,FS]}(w) \rceil$$

and

$$\lceil \gamma(\text{atom}(\mathfrak{R})) \rceil \wedge \lceil \text{rec}_{\mathfrak{R}}^{[W,ES]}(w) \rceil \leq \lceil \text{rec}_{\mathfrak{R}}^{[D,ES]}(w) \rceil$$

follow from repeated applications of Lemmas 2.6 and 2.7. \square

By applying Lemma 4.15 and Theorem 4.13 we are able to show that recognizability by final states is also equivalent to recognizability by empty stack in the width-first way provided a certain commutativity on the pushdown automata under consideration is imposed.

Corollary 4.16. For any $\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)$, if $\rightarrow = \rightarrow_3$ then we have:

$$\models^{\ell} \gamma(\text{atom}(\mathfrak{R})) \rightarrow (\exists \mathfrak{R}' \in \mathbf{PDA}(\Sigma, \ell)) (\forall w \in \Sigma^*) (\text{rec}_{\mathfrak{R}}^{[W,FS]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}'}^{[W,ES]}(w)).$$

Proof. We may observe from the proof of Theorem 4.13 that there exists an ℓ -valued PDA \mathfrak{R}' over Σ satisfying the following two conditions:

- (i) $\text{atom}(\mathfrak{R}) = \text{atom}(\mathfrak{R}')$, and
- (ii) $\lceil \text{rec}_{\mathfrak{R}}^{[D,FS]}(w) \rceil = \lceil \text{rec}_{\mathfrak{R}'}^{[D,FS]}(w) \rceil$ for any $w \in \Sigma^*$.

Now for each $w \in \Sigma^*$, using Lemmas 2.7, 2.12(3) and 4.15(3) we obtain:

$$\begin{aligned} \lceil \text{rec}_{\mathfrak{R}}^{[W,FS]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}'}^{[W,ES]}(w) \rceil &\geq \lceil \gamma(\text{atom}(\mathfrak{R}) \cup \text{atom}(\mathfrak{R}')) \rceil \wedge \lceil \text{rec}_{\mathfrak{R}}^{[W,FS]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[D,FS]}(w) \rceil \\ &\quad \wedge \lceil \text{rec}_{\mathfrak{R}}^{[D,FS]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}'}^{[D,ES]}(w) \rceil \\ &\quad \wedge \lceil \text{rec}_{\mathfrak{R}'}^{[D,ES]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}'}^{[W,ES]}(w) \rceil \\ &\geq \lceil \gamma(\text{atom}(\mathfrak{R}) \cup \text{atom}(\mathfrak{R}')) \rceil = \lceil \gamma(\text{atom}(\mathfrak{R})) \rceil. \quad \square \end{aligned}$$

4.3. Equivalence of Orthomodular Lattice-Valued Context Free Grammars and Pushdown Automata

We first establish equivalence of ℓ -valued CFGs and ℓ -valued PDAs in the depth-first way: each ℓ -valued CFG can be simulated by an ℓ -valued PDA, and vice versa, whenever the depth-first principle is applied. It should be noticed that

distributivity of \wedge over \vee is not required in this case. Thus, no commutators occur in the following two theorem.

Theorem 4.17. For any ℓ -valued language A over Σ , it holds that

$$\models^\ell CFL_\Sigma^{[D]}(A) \rightarrow (\exists \mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell))(A \equiv rec_{\mathfrak{R}}^{[D,ES]}),$$

and consequently

$$\models^\ell CFL_\Sigma^{[D]}(A) \rightarrow (\exists \mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell))(A \equiv rec_{\mathfrak{R}}^{[D,FS]}).$$

Proof. The second part is a simple corollary of the first one and Theorem 4.13. For the first part, using Theorem 4.7 we only need to show that for each GNF $G = \langle N, P, S \rangle$ in $\mathbf{CFG}(\Sigma, \ell)$, there is $\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)$ with $rec_{\mathfrak{R}}^{[D,ES]}(w) = L^{[D]}(G)(w)$ for any $w \in \Sigma^*$. Let $\mathfrak{R} = \langle \{q\}, N, \delta, S, \emptyset \rangle$, where $\delta((q, a, B), (q, \gamma)) = P(B \rightarrow a\gamma)$ for all $a \in \Sigma$, $B \in N$ and $\gamma \in N^*$, and $\delta(r) = 0$ for other rules r in $Rule(\mathfrak{R})$. This is a simple ℓ -valued modification of the construction given in the proof of Theorem 5.3 in [37]. Then by induction on the length $|c|$ of c we can prove the following:

Claim 1. For any $c \in QPath_{\mathfrak{R}}$ with $I(c) = (q, x, S)$ and $L(c) = (q, \varepsilon, \alpha)$, where $x \in \Sigma^*$, $\alpha \in N^*$, and $L(c)$ stands for the last configuration of c , there exists $d \in QDer_G$ such that $I(d) = S$, $L(d) = x\alpha$ and $\lceil Path_{\mathfrak{R}}(c) \rceil \leq \lceil Der_G(d) \rceil$.

Also, by induction on the length $|d|$ of d we are able to prove the following:

Claim 2. For any $d \in QDer_G$ with $I(d) = S$ and $L(d) = x\alpha$, where $x \in \Sigma^*$, and $\alpha \in N^*$, there exists $c \in QPath_{\mathfrak{R}}$ such that $I(c) = (q, x, S)$, $L(c) = (q, \varepsilon, \alpha)$ is the last configuration of c , and $\lceil Der_G(d) \rceil \leq \lceil path_{\mathfrak{R}}(c) \rceil$.

This indeed completes the proof. \square

Theorem 4.18. For any $\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)$, there exists $G \in \mathbf{CFG}(\Sigma, \ell)$ such that

$$\models^\ell w \in L^{[D]}(G) \leftrightarrow rec_{\mathfrak{R}}^{[D,ES]}(w)$$

for every $w \in \Sigma^*$. Consequently, for any $\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)$, there exists $G \in \mathbf{CFG}(\Sigma, \ell)$ such that

$$\models^\ell w \in L^{[D]}(G) \leftrightarrow rec_{\mathfrak{R}}^{[D,FS]}(w)$$

for every $w \in \Sigma^*$.

Proof. We only need to prove the first part. The second part follows from the first one and Theorem 4.13.

Suppose that $\mathfrak{R} = \langle Q, \Gamma, \delta, q_0, Z_0, F \rangle$. The grammar G is again an ℓ -valued modification of the corresponding classical construction presented in the proof of Theorem 5.4 in [37]. Let $G = \langle N, P, S \rangle$, where

$$N = \{S\} \cup \{[q, A, q'] : q, q' \in Q \text{ and } A \in \Gamma\},$$

$$P([q, A, q_{m+1}] \rightarrow a[q_1, B_1, q_2][q_2, B_2, q_3] \dots [q_m, B_m, q_{m+1}]) = \delta((q, a, A), (q_1, B_1 B_2 \dots B_m))$$

for all $q, q_1, q_2, \dots, q_m, q_{m+1} \in Q$, $a \in \Sigma \cup \{\varepsilon\}$ and $A, B_1, B_2, \dots, B_m \in \Sigma$,

$$P(S \rightarrow [q_0, Z_0, q]) = 1$$

for all $q \in Q$, and $P(p) = 0$ for other productions $p \in \text{Prod}(G)$. Then for any $q, q' \in Q$, $A \in \Gamma$ and $w \in \Sigma^*$, we have:

Claim 1. For any $d \in Q\text{Der}_G$ with $I(d) = [q', A, q]$ and $L(d) = w$, there exists $c \in Q\text{Path}_{\mathfrak{R}}$ such that $I(c) = (q', w, A)$, $L(c) = (q, \varepsilon, \varepsilon)$ and $[\text{Der}_G(d)] \leq [\text{Path}_{\mathfrak{R}}(c)]$.

Claim 2. For any $c \in Q\text{Path}_{\mathfrak{R}}$ with $I(c) = (q', w, A)$ and $L(c) = (q, \varepsilon, \varepsilon)$, there exists $d \in Q\text{Der}_G$ such that $I(d) = [q', A, q]$, $L(d) = w$ and $[\text{Path}_{\mathfrak{R}}(c)] \leq [\text{Der}_G(d)]$.

The first claim can be proved by induction on the length $|d|$ of d , and the second by induction on the length $|c|$ of c . The details are omitted here. Putting $q' = q_0$ and $A = Z_0$ in these claims, we obtain $L^{[D]}(G)(w) \leq [\text{rec}_{\mathfrak{R}}^{[D,ES]}(w)]$ and $[\text{rec}_{\mathfrak{R}}^{[D,ES]}(w)] \leq L^{[D]}(G)(w)$, respectively, and thus complete the proof. \square

A combination of the above two theorems and Lemmas 4.9 and 4.15 enables us to establish equivalence of ℓ -valued CFGs and ℓ -valued PDAs in the width-first way. Note that some commutators are needed in the following two corollaries.

Corollary 4.19. If $\rightarrow = \rightarrow_3$ then for any ℓ -valued language A over Σ , we have:

$$\models^{\ell} \text{CCFL}_{\Sigma}^{[W]}(A) \rightarrow (\exists \mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell))(A \equiv \text{rec}_{\mathfrak{R}}^{[W,ES]})$$

and the same for recognizability with final states.

Proof. From the proof of Theorem 4.17 we notice that for each ℓ -valued CFG G over Σ , we can find an ℓ -valued PDA \mathfrak{R} over Σ satisfying the following conditions: (i) $\text{atom}(\mathfrak{R}) \subseteq \text{atom}(G)$; and (ii) $L^{(D)}(G) = \text{rec}_{\mathfrak{R}}^{(D,ES)}$. Then using Lemmas 2.7, 2.12, 4.9(3) and 4.15(3) we obtain:

$$\begin{aligned} [A \equiv \text{rec}_{\mathfrak{R}}^{[W,ES]}] &\geq [A \equiv L^{[W]}(G)] \wedge [L^{[W]}(G) \equiv L^{[D]}(G)] \wedge [L^{[D]}(G) \equiv \text{rec}_{\mathfrak{R}}^{[D,ES]}] \\ &\quad \wedge [\text{rec}_{\mathfrak{R}}^{[D,ES]} \equiv \text{rec}_{\mathfrak{R}}^{[W,ES]}] \wedge [\gamma(\text{atom}(G) \cup r(A))] \\ &\geq [A \equiv L^{[W]}(G)] \wedge [\gamma(\text{atom}(G) \cup r(A))] \end{aligned}$$

and this completes the proof. \square

Corollary 4.20. Suppose that $\rightarrow = \rightarrow_3$. For any ℓ -valued PDA \mathfrak{R} over Σ , there is ℓ -valued CFG G over Σ such that

$$\models^{\ell} \gamma(\text{atom}(\mathfrak{R})) \rightarrow (w \in L^{[W]}(G) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[W,ES]}(w))$$

for each $w \in \Sigma^*$. The same holds for recognizability with final states.

Proof. Similar to Corollary 4.19. \square

4.4. Closure properties of orthomodular lattice-valued context-freeness

We first consider the union of orthomodular lattice-valued context-free grammars. Let $G_1 = \langle N_1, P_1, S_1 \rangle$, $G_2 = \langle N_2, P_2, S_2 \rangle \in \mathbf{CFG}(\Sigma, \ell)$. Then their union is defined to be

$$G_1 \cup G_2 = \langle N_1 \cup N_2 \cup \{S\}, P_1 \cup P_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}, S \rangle,$$

where it is assumed that $N_1 \cap N_2 = \emptyset$ and $S \notin N_1 \cup N_2$.

The following proposition gives a representation of the language generated by the union of two ℓ -valued CFGs in terms of their respective languages.

Proposition 4.21. For any ℓ -valued CFGs over Σ ,

- (i) $L^{[D]}(G_1 \cup G_2) = L^{[D]}(G_1) \cup L^{[D]}(G_2)$;
- (ii) $L^{[W]}(G_1 \cup G_2) = L^{[W]}(G_1) \cup L^{[W]}(G_2)$.

Proof. (i) is immediate from Definition 4.1. For (ii), it is obvious that $L^{[W]}(G_1)(w)$, $L^{[W]}(G_2)(w) \leq L^{[W]}(G_1 \cup G_2)(w)$ for any $w \in \Sigma^*$. Conversely, note that $N_1 \cap N_2 = \emptyset$. We can show that for any $n \geq 0$ and $w \in \Sigma^*$, $\lceil S_1 \xrightarrow[n]{G_1 \cup G_2} w \rceil = \lceil S_1 \xrightarrow[n]{G_1} w \rceil$ and $\lceil S_2 \xrightarrow[n]{G_1 \cup G_2} w \rceil = \lceil S_2 \xrightarrow[n]{G_2} w \rceil$ by induction on n . Then for each $n \geq 1$,

$$\begin{aligned} \lceil S \xrightarrow[n]{G_1 \cup G_2} w \rceil &= \bigvee_{\alpha} (\lceil S \xrightarrow[n]{G_1 \cup G_2} \alpha \rceil \wedge \lceil \alpha \xrightarrow[n-1]{G_1 \cup G_2} w \rceil) \\ &= \lceil S_1 \xrightarrow[n-1]{G_1 \cup G_2} w \rceil \vee \lceil S_2 \xrightarrow[n-1]{G_1 \cup G_2} w \rceil \\ &= \lceil S_1 \xrightarrow[n-1]{G_1} w \rceil \vee \lceil S_2 \xrightarrow[n-1]{G_2} w \rceil \\ &\leq L^{[W]}(G_1)(w) \vee L^{[W]}(G_2)(w) \end{aligned}$$

because $\lceil S \xrightarrow[n]{G_1 \cup G_2} \alpha \rceil = 0$ if $\alpha \neq S_1, S_2$. On the other hand, $\lceil S \xrightarrow[0]{G_1 \cup G_2} w \rceil = 0$. Consequently, we have:

$$\begin{aligned} L^{[W]}(G_1 \cup G_2)(w) &= \lceil S \xrightarrow[*]{G_1 \cup G_2} w \rceil \\ &= \bigvee_{n=0}^{\infty} \lceil S \xrightarrow[n]{G_1 \cup G_2} w \rceil \\ &\leq L^{[W]}(G_1)(w) \vee L^{[W]}(G_2)(w). \quad \square \end{aligned}$$

A direct corollary of the above proposition is that context-freeness is preserved by the union operation of orthomodular lattice-valued languages. To present such a preservation property we have to propose the notion of conformal and commutative context-freeness.

Definition 4.22. Let Σ be a nonempty finite alphabet. The conformal and commutative context-freeness of depth-first is defined to be a binary ℓ -valued predicate on L^{Σ^*} , $ConCCFL_{\Sigma}^{[D]} \in L^{L^{\Sigma^*} \times L^{\Sigma^*}}$, and for any $A_1, A_2 \in L^{\Sigma^*}$,

$$ConCCFL_{\Sigma}^{[D]} \stackrel{def}{=} (\exists G_1, G_2 \in \mathbf{CFG}(\Sigma, \ell)) (\gamma(atom(G_1) \cup atom(G_2) \cup r(A_1) \cup r(A_2)) \wedge (A_1 \equiv L^{[D]}(G_1)) \wedge (A_2 \equiv L^{[D]}(G_2))).$$

The conformal and commutative context-freeness of width-first, $ConCCFL_{\Sigma}^{[W]} \in L^{L^{\Sigma^*} \times L^{\Sigma^*}}$, may be defined by replacing the superscript $[D]$ with $[W]$ in the above equation.

Corollary 4.23. For any ℓ -valued languages A_1 and A_2 over Σ ,

$$\begin{aligned} \models^{\ell} ConCCFL_{\Sigma}^{[D]}(A_1, A_2) &\rightarrow C CFL^{[D]}(A_1 \cup A_2), \\ \models^{\ell} ConCCFL_{\Sigma}^{[W]}(A_1, A_2) &\rightarrow C CFL^{[W]}(A_1 \cup A_2). \end{aligned}$$

Proof. Similar to Corollary 3.21. \square

Second, we introduce the concatenation operation of two orthomodular lattice-valued context-free grammars. The concatenation of $G_1 = \langle N_1, P_1, S_1 \rangle$, $G_2 = \langle N_2, P_2, S_2 \rangle \in \mathbf{CFG}(\Sigma, \ell)$ is defined to be

$$G_1 \cdot G_2 = \langle N_1 \cup N_2 \cup \{S\}, P_1 \cup P_2 \cup \{S \rightarrow S_1 S_2\}, S \rangle,$$

where it is assumed that $N_1 \cap N_2 = \emptyset$, and S is chosen so that $S \notin N_1 \cup N_2$.

The relation between the language generated by the concatenation of two ℓ -valued CFGs G_1 and G_2 and the languages generated by G_1 and G_2 is quite complicated, and it is clearly exposed by the following proposition and Example 4.25 below.

Proposition 4.24. Let Σ be a nonempty finite alphabet.

(1) For any ℓ -valued CFGs G_1 and G_2 over Σ , and for any $w \in \Sigma^*$,

$$\models^{\ell} w \in L^{[D]}(G_1 \cdot G_2) \rightarrow w \in L^{[D]}(G_1) \cdot L^{[D]}(G_2).$$

(2) The following two statements are equivalent:

(2.1) ℓ is a Boolean algebra;

(2.2) for any ℓ -valued CFGs G_1 and G_2 over Σ , and for any $w \in \Sigma^*$,

$$\models^\ell w \in L^{[D]}(G_1 \cdot G_2) \leftrightarrow w \in L^{[D]}(G_1) \cdot L^{[D]}(G_2).$$

(3) For any ℓ -valued CFGs G_1 and G_2 over Σ , and for any $w \in \Sigma^*$, we have:

$$\models^\ell \gamma(\text{atom}(G_1) \cup \text{atom}(G_2)) \wedge w \in L^{[D]}(G_1) \cdot L^{[D]}(G_2) \rightarrow w \in L^{[D]}(G_1 \cdot G_2),$$

$$\models^\ell \gamma(\text{atom}(G_1) \cup \text{atom}(G_2)) \wedge w \in L^{[W]}(G_1) \cdot L^{[W]}(G_2) \rightarrow w \in L^{[W]}(G_1 \cdot G_2),$$

$$\models^\ell \gamma(\text{atom}(G_1) \cup \text{atom}(G_2)) \wedge w \in L^{[W]}(G_1) \cdot L^{[W]}(G_2) \rightarrow w \in L^{[W]}(G_1 \cdot G_2).$$

In particular, if $\rightarrow = \rightarrow_3$ then

$$\models^\ell \gamma(\text{atom}(G_1) \cup \text{atom}(G_2)) \rightarrow (w \in L^{[D]}(G_1 \cdot G_2) \leftrightarrow w \in L^{[D]}(G_1) \cdot L^{[D]}(G_2)),$$

$$\models^\ell \gamma(\text{atom}(G_1) \cup \text{atom}(G_2)) \rightarrow (w \in L^{[W]}(G_1 \cdot G_2) \leftrightarrow w \in L^{[W]}(G_1) \cdot L^{[W]}(G_2)).$$

Proof. We only prove that (2.2) implies (2.1) because the other conclusions may be proved by employing the technique frequently used in the previous proofs (say, the proof of Lemma 4.9). To do this, for any $\lambda, \mu_1, \mu_2 \in L$, let $G = \langle \{S_1\}, P_1, S_1 \rangle$ and $G_2 = \langle \{S_2, A_1, A_2\}, P_2, S_2 \rangle$, where $P_1(S_1 \rightarrow a) = \lambda$, $P_2(S_2 \rightarrow A_1) = \mu_1$, $P_2(S_2 \rightarrow A_2) = \mu_2$, $P_2(A_1 \rightarrow a^2) = P_2(A_2 \rightarrow a^2) = 1$, P_1 and P_2 take value 0 for all other productions, and $a \in \Sigma \neq \emptyset$. Then $L^{[D]}(G_1)(a) = \lambda$, $L^{[D]}(G_2)(a^2) = \mu_1 \vee \mu_2$, $(L^{[D]}(G_1) \cdot L^{[D]}(G_2))(a^3) = \lambda \wedge (\mu_1 \vee \mu_2)$, and $L^{[D]}(G_1 \cdot G_2)(a^3) = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2)$. Therefore, $(L^{[D]}(G_1) \cdot L^{[D]}(G_2))(a^3) = L^{[D]}(G_1 \cdot G_2)(a^3)$ implies $\lambda \wedge (\mu_1 \vee \mu_2) = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2)$. \square

The following example illustrates that both

$$\models^\ell w \in L^{[W]}(G_1 \cdot G_2) \rightarrow w \in L^{[W]}(G_1) \cdot L^{[W]}(G_2)$$

does not hold generally. We also have examples showing that

$$\models^\ell w \in L^{(W)}(G_1) \cdot L^{(W)}(G_2) \rightarrow w \in L^{(W)}(G_1 \cdot G_2)$$

is not valid.

Example 4.25. Consider a complete orthomodular lattice ℓ in which there are two infinite sequences $\{\lambda_n\}_{n=0}^\infty$ and $\{\mu_n\}_{n=0}^\infty$ of elements such that

$$\bigvee_{n=0}^\infty \lambda_n = \bigvee_{n=0}^\infty \mu_n = \theta < 1,$$

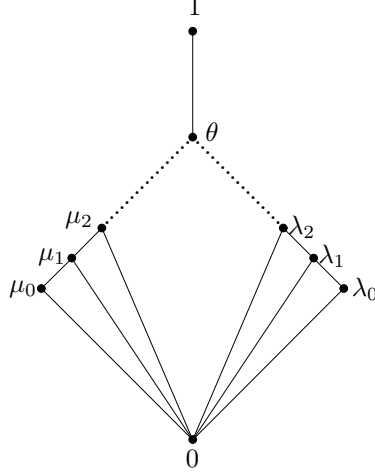


Figure 11: Two sequences

and their ordering is visualized by Figure 11. Let $G_1 = \langle \{S_1, A_1, A_2, \dots\}, P_1, S_1 \rangle$ and $G_2 = \langle \{S_2, B_1, B_2, \dots\}, P_2, S_2 \rangle$, where $P_1(S_1 \rightarrow a) = \lambda_0$, $P_1(S_1 \rightarrow A_1) = P_1(A_i \rightarrow A_{i+1}) = 1$ and $P_1(A_i \rightarrow a_i) = \lambda_i$ for each $i \geq 1$, $P_2(S_2 \rightarrow a^2) = \mu_0$, and $P_2(S_2 \rightarrow B_1) = P_2(B_i \rightarrow B_{i+1}) = 1$ and $P_2(B_i \rightarrow a^2) = \mu_i$ for each $i \geq 1$ (see Figures 12 and 13). Then it is clear that

$$L^{[W]}(G_1)(a) = \bigvee_{i=0}^{\infty} \lambda_i = \theta,$$

$$L^{[W]}(G_2)(a^2) = \bigvee_{i=0}^{\infty} \mu_i = \theta$$

and

$$(L^{[W]}(G_1) \cdot L^{[W]}(G_2))(a^3) = \theta.$$

On the other hand, by induction on n it is easy to prove that $\lceil S_1 S_2 \xrightarrow[n]{G_1 \cdot G_2} a^3 \rceil \leq \lambda_n \wedge \mu_n = 0$. This yields

$$L^{[W]}(G_1 \cdot G_2)(a^3) = \bigvee_{n=0}^{\infty} \lceil S_1 S_2 \xrightarrow[n]{G_1 \cdot G_2} a^3 \rceil = 0.$$

Therefore, $L^{[W]}(G_1 \cdot G_2)(a^3) < (L^{[W]}(G_1) \cdot L^{[W]}(G_2))(a^3)$.

A direct corollary of Proposition 4.25 indicates that context-freeness is preserved by concatenation of ℓ -valued languages whether the depth-first principle or the width-first principle is applied.

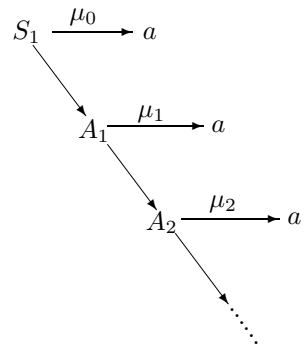


Figure 12: Grammar G_1

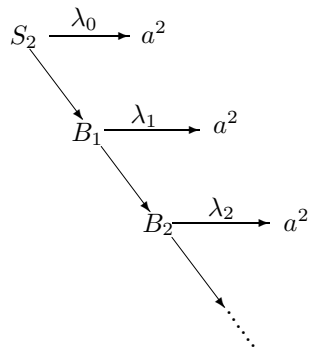


Figure 13: Grammar G_2

Corollary 4.26. For any ℓ -valued languages A_1 and A_2 over Σ ,

$$\begin{aligned} &\models \text{ConCCFL}_{\Sigma}^{[D]}(A_1, A_2) \rightarrow \text{CCFL}_{\Sigma}^{[D]}(A_1 \cdot A_2), \\ &\models \text{ConCCFL}_{\Sigma}^{[W]}(A_1, A_2) \rightarrow \text{CCFL}_{\Sigma}^{[W]}(A_1 \cdot A_2). \end{aligned}$$

Proof. Similar to Corollary 3.21. \square

For any ℓ -valued CFG $G = \langle N, P, S \rangle$ over alphabet Σ , its Kleene closure is defined to be $G^* = \langle N \cup \{S^*\}, P^*, S^* \rangle$, where $P^* = P \cup \{S^* \rightarrow \varepsilon, S^* \rightarrow SS^*\}$.

The language generated by the Kleene closure of an orthomodular lattice-valued context-free grammar is related in the following proposition to the Kleene closure of the language generated by this grammar.

Proposition 4.27. Let Σ be a nonempty finite alphabet.

(1) For any $G \in \mathbf{CFG}(\Sigma, \ell)$, and for any $w \in \Sigma^*$, we have:

$$\models^{\ell} w \in L^{[D]}(G^*) \rightarrow w \in (L^{[D]}(G))^{*[D]}.$$

(2) The following two statements are equivalent:

(2.1) ℓ is a Boolean algebra;

(2.2) for any $G \in \mathbf{CFG}(\Sigma, \ell)$, and for any $w \in \Sigma^*$, we have:

$$\models^{\ell} w \in L^{[D]}(G^*) \leftrightarrow w \in (L^{[D]}(G))^{*[D]}.$$

(3) For any $G \in \mathbf{CFG}(\Sigma, \ell)$, and for any $w \in \Sigma^*$, it holds that

$$\begin{aligned} &\models^{\ell} \gamma(\text{atom}(G)) \wedge w \in (L^{[D]}(G))^{*[D]} \rightarrow w \in L^{[D]}(G^*), \\ &\models^{\ell} \gamma(\text{atom}(G)) \wedge w \in L^{[W]}(G^*) \rightarrow w \in (L^{[W]}(G))^{*[W]}, \\ &\models^{\ell} \gamma(\text{atom}(G)) \wedge w \in (L^{[W]}(G))^{*[W]} \rightarrow w \in L^{[W]}(G^*). \end{aligned}$$

Especially, we have:

$$\begin{aligned} &\models^{\ell} \gamma(\text{atom}(G)) \rightarrow (w \in L^{[D]}(G^*) \rightarrow w \in (L^{[D]}(G))^{*[D]}), \\ &\models^{\ell} \gamma(\text{atom}(G)) \rightarrow (w \in L^{[W]}(G^*) \rightarrow w \in (L^{[W]}(G))^{*[W]}) \end{aligned}$$

whenever $\rightarrow = \rightarrow_3$.

Proof. (1) and (3) can be proved by the technique used before, and the implication from (2.1) to (2.2) follows immediately from (3). For the implication from (2.2) to (2.1), let $\lambda, \mu_1, \mu_2 \in L$. We set $G = \langle \{S, A, B\}, P, S \rangle$, where $P(S \rightarrow a) = \lambda$, $P(S \rightarrow A) = \mu_1$, $P(S \rightarrow B) = \mu_2$, $P(A \rightarrow a^2) = P(B \rightarrow a^2) = 1$, and P

takes value 0 for other productions. Then $L^{[D]}(G^*)(a^3) = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2)$ and $(L^{[D]}(G))^*{}^{[D]}(a^3) = \lambda \wedge (\mu_1 \vee \mu_2)$. So, (2.2) yields $\lambda \wedge (\mu_1 \vee \mu_2) = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2)$. This completes the proof. \square

As shown by the following corollary, context-freeness of orthomodular lattice-valued languages is also preserved by the Kleene closure operation.

Corollary 4.28. For each ℓ -valued language A over Σ , we have:

$$\begin{aligned} & \models^\ell CCFL_\Sigma^{[D]}(A) \rightarrow CCFL_\Sigma^{[D]}(A^{*[D]}), \\ & \models^\ell CCFL_\Sigma^{[W]}(A) \rightarrow CCFL_\Sigma^{[W]}(A^{*[W]}). \end{aligned}$$

Proof. We prove this corollary in the width-first way, and similarly it can be proved in the depth-first way. For any ℓ -valued CFG G over Σ , Lemma 2.12(1) enables us to assert that

$$\lceil \gamma(\text{atom}(G) \cup r(A)) \wedge A \equiv L^{[W]}(G) \rceil \leq \lceil A^{*[W]} \equiv (L^{[W]}(G))^*{}^{[W]} \rceil.$$

On the other hand, we obtain

$$\lceil \gamma(\text{atom}(G)) \rceil \leq \lceil (L^{[W]}(G))^*{}^{[W]} \equiv L^{[W]}(G^*) \rceil$$

from Proposition 4.27(3). Therefore, it follows that

$$\begin{aligned} \lceil \gamma(\text{atom}(G) \cup r(A)) \wedge A \equiv L^{[W]}(G) \rceil & \leq \lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \lceil A^{*[W]} \equiv (L^{[W]}(G))^*{}^{[W]} \rceil \\ & \quad \wedge \lceil (L^{[W]}(G))^*{}^{[W]} \equiv L^{[W]}(G^*) \rceil \\ & \leq \lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \lceil A^{*[W]} \equiv L^{[W]}(G^*) \rceil \end{aligned}$$

by using Lemma 2.12(3). Note that $\text{atom}(G^*) = \text{atom}(G) \cup \{\mathbf{1}\}$. With Lemma 2.7 we have

$$\lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \leq \lceil \gamma(\text{atom}(G^*) \cup r(A^{*[W]})) \rceil.$$

Thus,

$$\begin{aligned} \lceil CCFL_\Sigma^{[W]}(A) \rceil & = \bigvee_{G \in \mathbf{CFG}(\Sigma, \ell)} \lceil \gamma(\text{atom}(G) \cup r(A)) \wedge A \equiv L^{[W]}(G) \rceil \\ & \leq \bigvee_{G \in \mathbf{CFG}(\Sigma, \ell)} \lceil \gamma(\text{atom}(G^*) \cup r(A^{*[W]})) \wedge A^{*[W]} \equiv L^{[W]}(G^*) \rceil \\ & \leq \lceil CCFL_\Sigma^{[W]}(A^{*[W]}) \rceil. \quad \square \end{aligned}$$

Let Σ and Γ be two finite alphabets, and $h : \Sigma \rightarrow \Gamma^*$ a mapping. Suppose N is a finite set of nonterminal symbols. Then for any $\alpha = X_1 \dots X_n \in (N \cup \Sigma)^*$, we put

$h(\alpha) = h(X_1) \dots h(X_n)$, where $h(X_i) = X_i$ whenever $X_i \in N$. Furthermore, h can be extended to a mapping $h : \text{Prod}(G) \rightarrow \{A \rightarrow \beta : A \in N \text{ and } \beta \in (N \cup \Gamma)^*\}$ with $h(A \rightarrow \alpha) = A \rightarrow h(\alpha)$ for all $A \in N$ and $\alpha \in (N \cup \Sigma)^*$. Now for any ℓ -valued CFG $G = \langle N, P, S \rangle$ over Σ , we define the image of G under h to be the ℓ -valued CFG $h(G) = \langle N, h(P), S \rangle$ over Γ , where $h(P)$ is a finite ℓ -valued subset of $\text{Rule}(h(G))$, and it is the image of P under h , that is,

$$h(P)(A \rightarrow \beta) = \bigvee \{h(A \rightarrow \alpha) : \alpha \in (N \cup \Sigma)^* \text{ and } h(\alpha) = \beta\}$$

for all $A \in N$ and $\beta \in (N \cup \Gamma)^*$.

The fact that context-freeness of orthomodular lattice-valued languages is preserved by homomorphism is then presented by the following proposition.

Proposition 4.29. Let Σ and Γ be two nonempty finite alphabets, and $h : \Sigma \rightarrow \Gamma^*$ a mapping.

(1) For any ℓ -valued CFG G over Σ , and for any $w \in \Gamma^*$,

$$\models^\ell w \in h(L^{[D]}(G)) \rightarrow w \in L^{[D]}(h(G)).$$

(2) If $|\Sigma| \geq 2$, then the following two statements are equivalent:

(2.1) ℓ is a Boolean algebra;

(2.2) for any ℓ -valued CFG G over Σ , and for any $w \in \Gamma^*$,

$$\models^\ell w \in L^{[D]}(h(G)) \leftrightarrow w \in h(L^{[D]}(G)).$$

(3) For any ℓ -valued CFG G over Σ , and for any $w \in \Gamma^*$, we have:

$$\models^\ell \gamma(\text{atom}(G)) \wedge w \in L^{[D]}(h(G)) \rightarrow w \in h(L^{[D]}(G)).$$

In particular, if $\rightarrow = \rightarrow_3$ then

$$\models^\ell \gamma(\text{atom}(G)) \rightarrow (w \in L^{[D]}(h(G)) \leftrightarrow w \in h(L^{[D]}(G))).$$

The same conclusion is valid for $L^{[W]}(\cdot)$.

Proof. We only prove that (2.2) implies (2.1), and the proof of the other items is left for the reader. Suppose that $a, b \in \Sigma$, $a \neq b$, and $c \in \Gamma$, and let $f(a) = f(b) = c$. For any $\lambda, \mu_1, \mu_2 \in L$, we set $G = \langle \{S, A\}, P, S \rangle$, where $P(A \rightarrow A) = \lambda$, $P(A \rightarrow a) = \mu_1$, $P(A \rightarrow b) = \mu_2$, and P takes value 0 for the other productions. Then $L^{[D]}(G)(a) = \lambda \wedge \mu_1$, $L^{[D]}(G)(b) = \lambda \wedge \mu_2$, and $h(L^{[D]}(G))(c) = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2)$. On the other hand, $h(P)(S \rightarrow A) = \lambda$, $h(P)(A \rightarrow c) = \mu_1 \vee \mu_2$, and $L^{[D]}(h(G))(c) = \lambda \wedge (\mu_1 \vee \mu_2)$. Therefore, we obtain $\lambda \wedge (\mu_1 \vee \mu_2) = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2)$ from (2.2). \square

To show that orthomodular lattice-valued context-freeness is preserved by the pre-image of a homomorphism, we need to introduce the notion of pre-image of an orthomodular lattice-valued pushdown automaton under a homomorphism. If Σ is an alphabet, then for each $x \in \Sigma^*$, we write $\text{suff}(x)$ for the set of suffixes of x , that is, $\text{suff}(x) = \{y \in \Sigma^* : x = zy \text{ for some } z \in \Sigma^*\}$. Let $\mathfrak{R} = \langle Q, \Gamma, \delta, q_0, Z_0, F \rangle$ be an ℓ -valued PDA over alphabet Σ , and let Δ be a finite alphabet and $h : \Delta \rightarrow \Sigma^*$ be a mapping. Then the pre-image of \mathfrak{R} under h is defined to be the following ℓ -valued PDA over Δ :

$$h^{-1}(\mathfrak{R}) = \langle Q \times \bigcup_{a \in \Delta} \text{suff}(h(a)), \Gamma, h^{-1}(\delta), (q_0, \varepsilon), Z_0, F \times \{\varepsilon\} \rangle$$

where $h^{-1}(\delta)((q, x), a, X), ((p, y), \gamma)$, for $p, q \in Q, a \in \Delta \cup \{\varepsilon\}, x, y \in \bigcup_{a \in \Delta} \text{suff}(h(a)), X \in \Gamma$ and $\gamma \in \Gamma^*$, is given as follows:

$$h^{-1}(\delta)((q, \varepsilon), a, X), ((p, y), \gamma) = \begin{cases} 1, & \text{if } p = q, y = h(a) \text{ and } \gamma = x, \\ 0, & \text{otherwise,} \end{cases}$$

$$h^{-1}(\delta)((q, x), \varepsilon, X), ((p, y), \gamma) = \begin{cases} \delta((q, \text{head}(x), X), (p, \gamma)), & \text{if } y = \text{tail}(x), \\ 0, & \text{otherwise,} \end{cases}$$

and $h^{-1}(\delta)$ takes value 0 for all the other arguments.

The next proposition shows that the language accepted by the pre-image of an orthomodular lattice-valued pushdown automaton under a homomorphism is exactly the pre-image of the language accepted by the automaton under the same homomorphism.

Proposition 4.30. For any $\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)$, for any mapping $h : \Delta \rightarrow \Sigma^*$, and for any $w \in \Delta^*$,

$$\begin{aligned} \models^\ell \text{rec}_{h^{-1}(\mathfrak{R})}^{[D,FS]}(w) &\leftrightarrow \text{rec}_{\mathfrak{R}}^{[D,FS]}(h(w)), \\ \models^\ell \text{rec}_{h^{-1}(\mathfrak{R})}^{[W,FS]}(w) &\leftrightarrow \text{rec}_{\mathfrak{R}}^{[W,FS]}(h(w)). \end{aligned}$$

Proof. By a routine calculation and an easy induction. The details are omitted here. \square

Corollary 4.31. Let $h : \Delta \rightarrow \Sigma^*$ be a mapping. Then for any ℓ -valued language A over Σ ,

$$\models^\ell \text{CFL}_\Sigma^{[D]}(A) \rightarrow \text{CFL}_\Delta^{[D]}(h^{-1}(A));$$

and if $\rightarrow = \rightarrow_3$ then we have:

$$\models^\ell \text{CCFL}_\Sigma^{[W]}(A) \rightarrow \text{CCFL}_\Delta^{[W]}(h^{-1}(A)).$$

Proof. We only prove the second part. The first part is similar and much easier. First, using Corollary 4.19 and observing the construction in the proof of Theorem 4.17 we obtain:

$$\lceil CCFL_{\Sigma}^{[W]}(A) \rceil \leq \bigvee_{\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)} (\lceil A \equiv rec_{\mathfrak{R}}^{[W, FS]} \rceil \wedge \lceil \gamma(atom(\mathfrak{R}) \cup r(A)) \rceil).$$

Second, it is easy to see that $\lceil A \equiv rec_{\mathfrak{R}}^{[W, FS]} \rceil \leq \lceil h^{-1}(A) \equiv rec_{h^{-1}(\mathfrak{R})}^{[W, FS]} \rceil$ from Proposition 4.30. Consequently,

$$\begin{aligned} \lceil CCFL_{\Sigma}^{[W]}(A) \rceil &\leq \bigvee_{\mathfrak{R} \in \mathbf{PDA}(\Sigma, \ell)} (\lceil h^{-1}(A) \equiv rec_{h^{-1}(\mathfrak{R})}^{[W, FS]} \rceil \wedge \lceil \gamma(atom(h^{-1}(\mathfrak{R})) \cup r(A)) \rceil) \\ &\leq \bigvee_{\wp \in \mathbf{PDA}(\Delta, \ell)} (\lceil h^{-1}(A) \equiv rec_{\wp}^{[W, FS]} \rceil \wedge \lceil \gamma(atom(\wp) \cup r(A)) \rceil) \end{aligned}$$

because $atom(h^{-1}(\mathfrak{R})) \subseteq atom(\mathfrak{R})$.

Now for each $\wp \in \mathbf{PDA}(\Delta, \ell)$, from Corollary 4.20 we know that there must be $H \in \mathbf{CFG}(\Delta, \ell)$ with

$$\lceil \gamma(atom(\wp)) \rceil \leq \lceil rec_{\wp}^{[W, FS]} \equiv L^{[W]}(H) \rceil.$$

Then we have:

$$\begin{aligned} \lceil h^{-1}(A) \equiv L^{[W]}(H) \rceil &\geq \lceil h^{-1}(A) \equiv rec_{\wp}^{[W, FS]} \rceil \wedge \lceil rec_{\wp}^{[W, FS]} \equiv L^{[W]}(H) \rceil \\ &\quad \wedge \lceil \gamma(atom(\wp) \cup atom(H) \cup r(A)) \rceil \\ &\geq \lceil h^{-1}(A) \equiv rec_{\wp}^{[W, FS]} \rceil \wedge \lceil \gamma(atom(\wp) \cup atom(H) \cup r(A)) \rceil \end{aligned}$$

and

$$\begin{aligned} \lceil h^{-1}(A) \equiv rec_{\wp}^{[W, FS]} \rceil \wedge \lceil \gamma(atom(\wp) \cup r(A)) \rceil &\leq \lceil h^{-1}(A) \equiv L^{[W]}(H) \rceil \wedge \lceil \gamma(atom(\wp) \cup r(A)) \rceil \\ &\leq \lceil h^{-1}(A) \equiv L^{[W]}(H) \rceil \wedge \lceil \gamma(atom(H) \cup r(A)) \rceil \end{aligned}$$

since we can see $atom(H) \subseteq atom(\wp)$ from the proof of Theorem 4.18. Therefore, it follows that

$$\begin{aligned} \lceil CCFL_{\Sigma}^{[W]}(A) \rceil &\leq \bigvee_{H \in \mathbf{CFG}(\Delta, \ell)} (\lceil h^{-1}(A) \equiv L^{[W]}(H) \rceil \wedge \lceil \gamma(atom(H) \cup r(A)) \rceil) \\ &= \lceil CCFL_{\Delta}^{[W]}(h^{-1}(A)) \rceil. \quad \square \end{aligned}$$

Finally, we give a quantum logical generalization of the fact that the intersection of a regular language and a context-free language is context-free. To this end, we first introduce the product of orthomodular lattice-valued finite automaton and pushdown automaton. Let $\mathfrak{R} = \langle Q_A, \delta_A, p_0, F_A \rangle$ be an ℓ -valued (nondeterministic) finite automaton over alphabet Σ , and let $\wp = \langle Q_B, \Gamma, \delta_B, q_0, Z_0, F_B \rangle$ be an ℓ -valued

PDA over the same alphabet. Then their product is defined to be the following ℓ -valued PDA over Σ :

$$\mathfrak{R} \times \wp = \langle Q_A \times Q_B, \Gamma, \delta, (p_0, q_0), Z_0, F_A \times F_B \rangle,$$

where δ is a finite ℓ -valued subset of $[(Q_A \times Q_B) \times (\Sigma \cup \{\varepsilon\}) \times \Gamma] \times [(Q_A \times Q_B) \times \Gamma^*]$, and it is given as follows: for any $p, p' \in Q_A$, $q, q' \in Q_B$, $a \in \Sigma$, $Z \in \Gamma$, and $\gamma \in \Gamma^*$,

$$\begin{aligned} \delta(((p, q), a, Z), ((p', q'), \gamma)) &= \delta_A(p, a, p') \wedge \delta_B((q, a, Z), (q', \gamma)), \\ \delta(((p, q), \varepsilon, Z), ((p', q'), \gamma)) &= \begin{cases} \delta_B((q, \varepsilon, Z), (q', \gamma)), & \text{if } p' = p, \\ 0, & \text{otherwise,} \end{cases} \end{aligned}$$

Proposition 4.32. Let Σ be a nonempty alphabet.

(1) For any ℓ -valued finite automaton \mathfrak{R} and ℓ -valued PDA \wp over Σ , and for any $w \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R} \times \wp}^{[D,FS]}(w) \rightarrow \text{rec}_{\mathfrak{R}}^{[D,FS]}(w) \wedge \text{rec}_{\wp}^{[D,FS]}(w).$$

(2) The following two statements are equivalent:

(2.1) ℓ is a Boolean algebra;

(2.2) for any ℓ -valued finite automaton \mathfrak{R} and ℓ -valued PDA \wp over Σ , and for any $w \in \Sigma^*$,

$$\models^\ell \text{rec}_{\mathfrak{R} \times \wp}^{[D,FS]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[D,FS]}(w) \wedge \text{rec}_{\wp}^{[D,FS]}(w);$$

(3) For any ℓ -valued finite automaton \mathfrak{R} and ℓ -valued PDA \wp over Σ , and for any $w \in \Sigma^*$,

$$\models^\ell \gamma(\text{atom}(\mathfrak{R}) \cup \text{atom}(\wp)) \wedge \text{rec}_{\mathfrak{R}}^{[D,FS]}(w) \wedge \text{rec}_{\wp}^{[D,FS]}(w) \rightarrow \text{rec}_{\mathfrak{R} \times \wp}^{[D,FS]}(w).$$

In particular, if $\rightarrow = \rightarrow_3$ then we have:

$$\models^\ell \gamma(\text{atom}(\mathfrak{R}) \cup \text{atom}(\wp)) \rightarrow (\text{rec}_{\mathfrak{R} \times \wp}^{[D,FS]}(w) \leftrightarrow \text{rec}_{\mathfrak{R}}^{[D,FS]}(w) \wedge \text{rec}_{\wp}^{[D,FS]}(w)).$$

The same conclusion holds for recognizability in the width-first way.

Proof. We first show that (2.2) implies (2.1). For any $\lambda, \mu_1, \mu_2 \in L$, assume that $a \in \Sigma$, and put $\mathfrak{R} = \langle \{p_0, p_1\}, \delta_A, p_0, \{p_1\} \rangle$ and $\wp = \langle \{q_0, q_1, q_2\}, \{Z_0\}, \delta_B, q_0, Z_0, \{q_1, q_2\} \rangle$, where $\delta_A(p_0, a, p_1) = \lambda$, $\delta_B((q_0, a, Z_0), (q_1, \varepsilon)) = \mu_1$, $\delta_B((q_0, a, Z_0), (q_2, \varepsilon)) = \mu_2$, and they take value 0 for the other cases. Then

$$\lambda \wedge (\mu_1 \vee \mu_2) = \lceil \text{rec}_{\mathfrak{R}}^{[D]}(a) \rceil \wedge \lceil \text{rec}_{\wp}^{[D,FS]}(a) \rceil = \lceil \text{rec}_{\mathfrak{R} \times \wp}^{[D,FS]}(a) \rceil = (\lambda \wedge \mu_1) \vee (\lambda \wedge \mu_2).$$

By a similar construction of ℓ -valued automata, we can also show that (2.3) implies (2.1). The other conclusions may be proved by the technique that we used before. \square

Definition 4.33. Let Σ be a finite alphabet. Then ℓ -valued binary predicate $ConCRegCCFL_{\Sigma}^{(W)} \in L^{L^{\Sigma^*} \times L^{\Sigma^*}}$ on ℓ -valued languages over Σ is defined by

$$ConCRegCCFL_{\Sigma}^{(W)}(A, B) \stackrel{def}{=} (\exists \mathfrak{R} \in \mathbf{NFA}(\Sigma, \ell), G \in \mathbf{CFG}(\Sigma, \ell)) (\gamma(atom(\mathfrak{R}) \cup atom(G) \cup r(A) \cup r(B)) \wedge A \equiv rec_{\mathfrak{R}}^{[W]} \wedge B \equiv L^{[W]}(G))$$

for all $A, B \in L^{\Sigma^*}$. Intuitively, $ConCRegCCFL_{\Sigma}^{[W]}(A, B)$ is interpreted as the proposition that in a conformal way, A is commutatively regular and B is commutatively context-free according to the width-first principle.

Corollary 4.34. For any ℓ -valued languages A and B over Σ ,

$$\models^{\ell} ConCRegCCFL_{\Sigma}^{[W]}(A, B) \rightarrow CCFL_{\Sigma}^{[W]}(A \cap B).$$

Proof. Similar to Corollary 3.21. \square

4.5. The Pumping Lemma for Orthomodular Lattice-Valued Context Free Languages

The aim of this subsection is to establish a quantum logical generalization of the pumping lemma for context-free languages. The idea and proof technique used here are similar to that in Subsection 3.7.

Theorem 4.35. Let Σ be a finite alphabet, and let $\rightarrow = \rightarrow_3$. Then for each ℓ -valued language A over Σ , we have:

$$\models^{\ell} CCFL_{\Sigma}^{[W]}(A) \rightarrow (\exists n \geq 0) (\forall z \in \Sigma^* (z \in A \wedge |z| \geq n \rightarrow (\exists u, v, w, x, y \in \Sigma^*) (z = uvwxy \wedge |vx| \geq 1 \wedge |vwx| \leq n \wedge (\forall i \geq 0) (uv^iwx^iy \in A)))).$$

Proof. For simplicity, we write $X(z, u, v, w, x, y, n)$ for the condition that $u, v, w, x, y \in \Sigma^*$, $z = uvwxy$, $|vx| \geq 1$ and $|vwx| \leq n$. Since

$$\lceil CCFL_{\Sigma}^{[W]}(A) \rceil = \bigvee_{G \in \mathbf{CFG}(\Sigma, \ell)} (\lceil \gamma(atom(G) \cup r(A)) \rceil \wedge \lceil A \equiv L^{[W]}(G) \rceil),$$

we only need to show that for any $G = \langle N, P, S \rangle \in \mathbf{CFG}(\Sigma, \ell)$,

$$\lceil \gamma(atom(G) \cup r(A)) \rceil \wedge \lceil A \equiv L^{[W]}(G) \rceil \leq \bigvee_{n \geq 0} \bigwedge_{z \in \Sigma^* \text{ s.t. } |z| \geq n} (A(z) \rightarrow \bigvee_{X(z, u, v, w, x, y, n)} \bigwedge_{i \geq 0} A(uv^iwx^iy)).$$

From Lemma 4.10(3) we know that $\lceil \gamma(\text{atom}(G)) \rceil \leq \lceil L^{(W)}(G) \equiv L^{(D)}(G) \rceil$. This together with Lemma 2.11(3) leads to

$$\begin{aligned} \lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \lceil A \equiv L^{(W)}(G) \rceil &\leq \lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \lceil A \equiv L^{(W)}(G) \rceil \\ &\quad \wedge \lceil L^{(W)}(G) \equiv L^{(D)}(G) \rceil \\ &\leq \lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \lceil A \equiv L^{(D)}(G) \rceil. \end{aligned}$$

We write $k = |N|$ and $l = \max\{|\alpha| : \alpha \in (N \cup \Sigma)^* \text{ and } P(A \rightarrow \alpha) > 0 \text{ for some } A \in N\}$. Let $n_0 = l^k$. Then it suffices to show that for all $z \in \Sigma^*$ with $|z| \geq n_0$,

$$\lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \lceil A \equiv L^{(D)}(G) \rceil \leq A(z) \rightarrow \bigvee_{X(z,u,v,w,x,y,n)} \bigwedge_{i \geq 0} A(uv^iwx^i y).$$

Note that

$$\begin{aligned} &\lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \lceil A \equiv L^{(D)}(G) \rceil \wedge (A(z) \rightarrow \bigvee_{X(z,u,v,w,x,y,n)} \bigwedge_{i \geq 0} L^{(D)}(G)(uv^iwx^i y)) \\ &\leq \lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \left(\bigvee_{X(z,u,v,w,x,y,n)} \bigwedge_{i \geq 0} L^{(D)}(G)(uv^iwx^i y) \rightarrow \bigvee_{X(z,u,v,w,x,y,n)} \bigwedge_{i \geq 0} A(uv^iwx^i y) \right) \\ &\quad \wedge (A(z) \rightarrow \bigvee_{X(z,u,v,w,x,y,n)} \bigwedge_{i \geq 0} L^{(D)}(G)(uv^iwx^i y)) \\ &\leq A(z) \rightarrow \bigvee_{X(z,u,v,w,x,y,n)} \bigwedge_{i \geq 0} A(uv^iwx^i y) \end{aligned}$$

follows from Lemmas 2.6 and 2.11(1) and (3). We now only need to prove that

$$\lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \lceil A \equiv L^{(D)}(G) \rceil \leq (A(z) \rightarrow \bigvee_{X(z,u,v,w,x,y,n)} \bigwedge_{i \geq 0} L^{(D)}(G)(uv^iwx^i y)),$$

which can be derived from

$$\lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \lceil A \equiv L^{(D)}(G) \rceil \wedge A(z) \leq \bigvee_{X(z,u,v,w,x,y,n)} \bigwedge_{i \geq 0} L^{(D)}(G)(uv^iwx^i y)$$

by using Lemma 2.10 and observing $\lceil \gamma(\text{atom}(G) \cup r(A)) \rceil CA(z)$. On the other hand, we see that

$$\lceil \gamma(\text{atom}(G) \cup r(A)) \rceil \wedge \lceil A \equiv L^{(D)}(G) \rceil \wedge A(z) \leq L^{(D)}(G)(z)$$

by Lemma 2.11(5). Therefore, it suffices to show that

$$L^{(D)}(G)(z) \leq \bigvee_{X(z,u,v,w,x,y,n)} \bigwedge_{i \geq 0} L^{(D)}(G)(uv^iwx^i y),$$

that is,

$$\lceil \text{Der}_G(d) \rceil \leq \bigvee_{X(z,u,v,w,x,y,n)} \bigwedge_{i \geq 0} L^{(D)}(G)(uv^iwx^i y)$$

for each $d \in QDer_G$ with $I(d) = S$ and $L(d) = z$. With Lemma 4.4 it may be assumed that $P(p) = 0$ for all ε -productions or unit productions p . The quasi-derivation d can be represented as a derivation tree in a way similar to that we used for classical context-free grammars. Each of the interior vertex of its derivation tree is labelled with a symbol from N together with an element of L , and each leaf vertex is labelled with a symbol from Σ . If an interior vertex is labelled with $A \in N$ and $\lambda \in L$, and its sons are labelled with X_1, X_2, \dots, X_k from the left, then $P(A \rightarrow X_1 X_2 \dots X_k) = \lambda > 0$. Finally, we can prove the last inequality and thus complete the proof by following the procedure of the proof of Lemma 6.1 in [37] (note that there l is taken to be 2 because a Chomsky normal form is applied). \square

5. Conclusion

It is argued that a theory of computation based on quantum logic has to be established as a logical foundation of quantum computation. Finite automata and pushdown automata are among the simplest abstract mathematical models of computing machines, and automata theory is an essential part of computation theory. In recent years, the author and his colleagues have tried to establish automata theory based on quantum logic. This Chapter is a systematic exposition of such a new theory. In this theory, quantum logic is treated as an orthomodular lattice-valued logic. The approach employed in developing this theory is the semantical analysis suggested by the author in his previous work on topology based on nonclassical logics [80, 81, 82]. The notions of orthomodular lattice-valued finite automaton and pushdown automata and their various variants are introduced. The classes of languages accepted by them are defined, namely orthomodular lattice-valued regular languages and context-free languages. Various properties of automata are reexamined in the framework of quantum logic, including the closure properties of regular languages and context-free languages under various operations, the Kleene theorem concerning equivalence between finite automata and regular expressions, equivalence between pushdown automata and context-free grammars, and the pumping lemma both for regular languages and for context-free languages.

To complete the picture of computing theory based on quantum logic, we shall develop a theory of orthomodular lattice-valued Turing machines in a forthcoming paper.

In the development of automata theory based on quantum logic, some essential differences between the computation theory established by using the classical Boolean logic as the underlying logical tool and that whose meta-logic is quantum logic have been observed. The most interesting thing is, in the author's opinion, the discovery that the universal validity of many fundamental properties (for example, the Kleene theorem) of automata depend heavily upon the distributivity of the underlying logic. It is shown that the universal validity of these properties is equivalent to the requirement that the set of truth values of the meta-logic underlying our theory of automata is a Boolean algebra. This implies that these properties does not universally hold in the realm of quantum logic, and it is in fact a negative conclusion in our theory of automata based on quantum logic. These differences have some significant implications, among which the following are two of the most direct:

- (1) In 1959, M. O. Rabin and D. S. Scott [56] introduced the idea of nondeterministic machines and showed that each nondeterministic finite automaton can be simulated by a deterministic one by the subset construction. Now the notion of nondeterminism has proved to be extremely valuable in computing theory, and it has been a continuous source of inspiration for subsequent researches. Theorem 3.11 in this Chapter shows that Rabin and Scott's equivalence of nondeterministic and deterministic finite automata may be directly generalized into quantum logic if we

comply with the width-first principle when treating interactions between conjunction and disjunction. However, if the depth-first principle is adopted then Theorem 3.10 indicates that each nondeterministic finite automaton is equivalent to its subset construction if and only if the underlying logic is distributive. This exposes a close link between logical distributivity and nondeterminism in computation. Clearly, such an observation could not be made if we only worked within a logical system that enjoys distributivity.

(2) In this Chapter we merely see that the proofs of some even very basic properties of finite and pushdown automata appeal an essential application of the distributivity for the lattice of truth values of the underlying logic. But we believe that there are also many fundamental properties of Turing machines whose universal validity requires the distributivity of meta-logic, and so they hold only in Boolean logic but not in quantum logic. This will provides us with some new negative results in the theory of computation based on quantum logic. Since quantum logic is a logical mechanism that governs the behaviors of quantum systems, these negative results might hint some limitations of quantum computation. More explicitly, some techniques based on certain properties of classical automata maybe have been successfully used in the implementation of classical computing systems, but they do not apply to quantum computers, or at least they are only conditionally effective for quantum computers.

These negative conclusions may have influence even in a wider area. Although at this moment we have merely found several negative results in the computing theory based on quantum logic, it seems that some negative results of similar nature exist in other mathematical theories based on nonclassical logics without distributivity. This stimulates us to consider the problem of a further logical revisit to mathematics. Various classical mathematical results have been established based upon classical logic, and sometimes, their universal validity can only be established by exploiting the full power of classical logic. Mathematicians usually use logic implicitly in their reasoning, and they do not seriously care which logical laws they have employed. But from a logician's point of view, it is very interesting to determine how strong a logic we need to validate a given mathematical theorem, and which logic guarantees this theorem and which does not among the large population of nonclassical logics. To be more explicit and also for a comparison, let us present a short excerpt once again from A. Heyting [32] (see page 3):

“It may happen that for the proof of a theorem we do not need all the axioms, but only some of them. Such a theorem is true not only for models of the whole system, but also for those of the smaller system which contains only the axioms used in the proof. Thus it is important in an axiomatic theory to prove every theorem from the least possible set of axioms.”

We now are in a similar situation. The difference between our case and A. Heyting's one is that we are concerned with the limitation or redundance of power of the logic underlying an axiomatic theory, whereas he considered that of axioms themselves. It seems that the semantical analysis approach provides a nice framework for

solving this kind of problems, much more suitable than a proof-theoretical approach.

Observing that some important properties of automata cannot be built within quantum logic, one may naturally ask the question whether they may be partially recast without appealing to distributivity of the underlying logic. Fortunately, we are able to show that a local validity of these properties of automata can be recovered by imposing a certain commutativity to the truth values of the (atomic) statements about the automata under consideration. Very surprisingly, almost all results in classical automata theory that are not valid in a non-distributive logic can be revived by a certain commutativity in quantum logic. A typical example is the equivalence of deterministic and nondeterministic finite automata with respect to acceptance defined in the depth-first way. Another interesting example is the pumping lemma for regular languages which is not valid for the notion of noncommutative regularity but survives for commutative regularity. This is in fact a partial reason for introducing two different versions of regularity in Section 3.

The successful applications of commutativity in the development of our automata theory based on quantum logic further lead us to a new question: why commutativity plays such a key role for quantum automata, and is there any physical interpretation for it? To answer this question, let us first note that all truth values in quantum logic are taken from an orthomodular lattice. The prototype of orthomodular lattice is the set of linear (closed) subspaces of a Hilbert space with the set inclusion as its ordering relation. Suppose that X and Y are two subspaces of a Hilbert space H . Moreover, we use P_X and P_Y to denote the projections on X and Y respectively. Then P_X and P_Y are Hermitian operators on H , and they may be seen as two (physical) observables A and B in a quantum system whose state space is H , according to the Hilbert space formalism of quantum mechanics. If we write $\Delta(A)$ and $\Delta(B)$ for the respective standard deviations of measurement on A and B , then the Heisenberg uncertainty principle gives the following inequality (see [50], page 89):

$$\Delta(A) \cdot \Delta(B) \geq \frac{1}{2} |\langle \psi | [A, B] | \psi \rangle|$$

for all quantum state $|\psi\rangle$ in H , where $[A, B] = AB - BA$ is the commutator between A and B . We now turn back to the orthomodular lattice of the linear subspaces of H . The commutativity of X and Y is defined by the following condition: $X = (X \wedge Y) \vee (X \wedge Y^\perp)$, where \wedge , \vee and \perp are respectively the meet, union and orthocomplement. It may be seen that the commutativity between X and Y is equivalent to exactly the fact that A and B commute, i.e., $AB = BA$. In this case, $|\langle \psi | [A, B] | \psi \rangle| = 0$, and $\Delta(A) \cdot \Delta(B)$ may vanish; or in other words, $\Delta(A)$ and $\Delta(B)$ can simultaneously become arbitrarily small. Remember that in our theory of automata based on quantum logic the commutativity is attached to the basic statements describing the considered automata. On the other hand, the basic statements are indeed corresponding to some actions in these automata. Therefore, a potential physical interpretation for the need of commutativity is that some nice properties of automata require the standard deviations of the observables concerning the basic actions in these automata being able to reach simultaneously very small values.

The results gained in our approach may offer some new insights on the theory of computation. As an example, let us consider the Church-Turing thesis, one of the most fundamental hypotheses in the whole field of computer science. The realization that the intuitive notion of “effective computation” can be identified with the mathematical concept of “computation by the Turing machine” is based on the fact that the Turing machine is computationally equivalent to some vastly dissimilar formalisms for the same purpose, such as Post systems, μ -recursive functions, λ -calculus and combinatory logic. As pointed out by J. E. Hopcroft and J. D. Ullman [37], another reason for the acceptance of the Turing machine as a general model of computation is that the Turing machine is equivalent to its many modified versions that would seem off-hand to have increased computing power. We should note that the equivalence between the Turing machine and its various generalizations as well as other formalisms of computation has been reached in classical Boolean logic. In addition, quantum logic is known to be strictly weaker than Boolean logic. Thus, it is reasonable to doubt that the same equivalence can be achieved when our underlying meta-logic is replaced by quantum logic, and the Church-Turing thesis needs to be reexamined in the realm of quantum logic. Indeed, in a forthcoming paper we are going to establish a theory of Turing machines based on quantum logic. The details of such a theory is still to be exploited, but the conclusion concerning the equivalence between deterministic and nondeterministic finite automata presented in this Chapter suggests us to believe that the equivalence between deterministic and nondeterministic Turing machines also depends upon the distributivity of the underlying logic, and a certainty commutativity for the basic actions in Turing machines will guarantee such an equivalence. Keeping this belief in mind, we may assert that a certain commutativity of the observables for some basic actions in the Turing machine is a physical support of the Church-Turing thesis in the framework of quantum logic. Furthermore, with the above physical interpretation for commutativity, this hints that there might be a deep connection between the Heisenberg uncertainty principle and the Church-Turing thesis. It is notable that such a connection could be observed via an argument in a nonclassical logic, but it is impossible to be found if we always work within the classical logic in which distributivity is automatically valid. As early as in 1985, it was argued by D. Deutsch [?] that underlying the Church-Turing thesis there is an implicit physical assertion. There is certainly no doubt about the existence of such a physical assertion. The true problem here is: what is it? The answer given by D. Deutsch himself is the following physical principle: “every finitely realizable physical system can be perfectly simulated by a universal model computing machine operating by finite means”. Our above analysis on the role of commutativity in computation theory based on quantum logic perhaps indicates that in order to be simulated by a universal computing machine some observables of the physical system are required to possess a certain commutativity. So, it is fair to say that the observation on commutativity presented above provides a complement to D. Deutsch’s argument from a logical point of view. Furthermore, if such a physical interpretation to the role of commutativity based on the Heisenberg uncertainty principle is reasonable, and the conjecture that a certain connection may

reside between the Heisenberg uncertainty principle and the Church-Turing thesis, two of the greatest scientific discoveries in the twentieth century, is true, then it will give once again an evidence to the unity of the whole science and to the fact that science is not only a simple union of various subjects.

As to the further development of computation theory based on quantum logic, we are of course concerned with the behavior of other models of computation, such Turing machines, in the framework of quantum logic. As mentioned above, the theory of orthomodular lattice-valued Turing machines will be developed in a forthcoming paper by the author.

One of the most interesting things, according to the author's opinion, is to exploit its connection to other mathematical models of quantum computation. Roughly speaking, quantum automata (including quantum finite automata, quantum push-down automata and quantum Turing machines) in the previous literature may be seen as quantum counterparts of probabilistic automata. In a probabilistic automaton, each transition is equipped with a number in the unit interval to indicate the probability of the occurrence of the transition; by contrast in a quantum automaton we associate with each transition a vector in a Hilbert space, which is interpreted as the probability amplitude of the transition. Furthermore, according to the basic postulates of quantum mechanism, the evolution of quantum automata is described in terms of unitary operators [4, 11, 17, 18, 29]. Thus, a potential way of establishing link between these quantum automata and our orthomodular lattice-valued automata is to use Takeuti, Titani and Kozawa's representation of observables and unitary operators by real and complex numbers in the universe $V^{P(H)}$ of quantum set theory, where $P(H)$ is the orthomodular lattice of closed linear subspaces of a Hilbert space H [71, 74]. It is also possible to find a link between orthomodular lattice-valued automata and probabilistic automata via the Gleason theorem (see [21] for a detailed exposition), which completely characterizes probability measures, called states, on the orthomodular lattice $P(H)$ for a separable real or complex Hilbert space H .

In this paper, as we have seen, quantum logic is considered as orthomodular lattice-valued logic, and our theory of computation based on quantum logic is developed with the algebraic semantics. Another interesting problem for further studies would be to establish a theory of computation with the Kripke semantics of quantum logic [13] and to compare it with the theory of the current paper. Moreover, some quantum logics essentially different the orthomodular lattice-valued quantum logic have been proposed in the literature: one recent example is P. Mateus and A. Sernadas's exogenous quantum logic [47], and another is G. Domenech and H. Freytes's contextual quantum logic [20]. It is also interesting to develop computing theory based on these new quantum logics.

6. Bibliographical Notes

The basic idea of establishing a theory of computation based on quantum logic was first proposed by the author in [83], where among other things, equivalence between distributivity of the underlying logic and some simple properties of finite automata was already observed. A systematic development of the theory of finite automata based on quantum logic was presented in [84]. Indeed, the first part of this Chapter is a slightly revised version of [84]. The main change is that regularity in the width-first way is introduced (and thus regularity presented in [84] is renamed as regularity in the depth-first way), and an orthomodular lattice-valued generalization of the Myhill-Nerode theorem is given. The new material of the Chapter is the second part in which the theory of pushdown automata and context-free grammars is thoroughly developed.

After [83], several other authors have also contributed to automata theory based on quantum logic. In [44], R. Q. Lu and H. Zheng introduced a new recognizability for lattice-valued finite automata. It is different from both that given in [83, 84], recognizability in the depth-first way, and recognizability in the width-first way (see Definition 3.2), and indeed it is stronger than the former but weaker than the latter. They also carefully examined recognizability of various lattice-theoretic operations of lattice-valued finite automata. In [45], R. Q. Lu and H. Zheng further proved some interesting variants of pumping lemma for lattice-valued finite automata. A straightforward generalization of pumping lemma to lattice-valued automata was also obtained by D. W. Qiu in [54]. Note that the pumping lemmas given in [45, 54] are concerned directly with lattice-valued automata, whereas the pumping lemmas presented in [83, 84] is stated for orthomodular lattice-valued regular languages. From [54] we can see that distributivity of the underlying logic is not necessary in establishing a pumping lemma directly for finite automata. However, it may be noticed from [84] that certain distributivity of the underlying logic (represented by the commutator) is required when a pumping lemma is given for regular languages. This indicates that logical distributivity is used to fill in the gap between finite automata and regular languages. In this Chapter, the version of pumping lemma in [84] is adopted because we believe that adding commutator into a pumping lemma grasps something essential in quantum logic. Some further observations on equivalence between logical distributivity and automata-theoretic properties, including some topological characterizations of finite automata, were made by D. W. Qiu [55]. In [9], W. Cheng and J. Wang proposed the notion of lattice-valued regular grammar and proved the equivalence of lattice-valued regular grammars and finite automata.

The methodology that we used to establish our automata theory based on quantum logic is semantical analysis developed in [80, 81]. Indeed, a similar approach have been widely applied by G. Takeuti's school in Boolean-valued and Heyting-valued analysis and algebra; see for example [71, 73, 51, 52]. We should note that the mathematical theories developed there are still based on Boolean logic or intuitionistic logic. However, the idea and proof techniques were extended by the same

group of authors to build some mathematical theories based on quantum logic. For example, as already stated in the introduction, G. Takeuti [72] proposed orthomodular lattice-valued set theory. Indeed, the commutator introduced in [72] has played an important role in our automata theory based on quantum logic presented in this Chapter. Takeuti's quantum set theory was further developed by S. Titani and H. Kozawa [74] with a strong implication corresponding to the order in the lattice of truth values. In [75], K. Tokuo presented quantum number theory based a typed version of quantum logic.

The aim of this Chapter is to establish automata theory in the framework of quantum logic. Another approach of connecting automata theory to quantum logic was proposed by K. Svozil et al. In a series of paper [62, 63, 64, 22], they found some interesting connections of Moore and Mealy-type automata with some orthoalgebraic structures. The link between the theory presented in this Chapter and the approach of K. Svozil et al. is still to be exploited.

There are quite a few logical approaches to quantum computation in the literature, essentially different from that presented in this Chapter. J. P. Rawling and S. A. Selesnick [57] proposed a quantization scheme of gates in classical circuits using Kripke models of quantum logic, and then gave a logical interpretation of the notion of quantum parallelism. S. A. Selesnick [66] tried to provide a logical foundation for quantum computing. He introduced a Gentzen sequent calculus for handling quantum resources which can be interpreted in the category of finite-dimensional Hilbert spaces with the aid of Grassmannian quantum set theory. Then various quantum phenomena such as qubits, quantum storage, quantum copying and quantum entanglement may be specified in this calculus. M. L. Dalla Chiara, R. Giuntini and R. Leporini [15] proposed so-called quantum computational logic whose semantics is given in terms of quantum gates and circuits (see also [16], Chapter 17). This logic is then generalized by S. Gudder [30] in order to deal with mixed states. O. Brunet and P. Jorrand [7] presented a dynamic quantum logic in which both unitary operators and quantum measurements may be handled. Thus, it can be used to describe quantum programs.

References

- [1] J. Barwise, Model-theoretic logics: background and aims, in: J. Barwise and S. Feferman (eds.), *Model Theoretic Logics*, Springer-Verlag, Berlin, 1985, pp. 3-27.
- [2] P. A. Benioff, The computer as a physical system: a microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines, *Journal of Statistical Physics* **22**(1980)563-591.
- [3] C. H. Bennet, Logical reversibility of computation, *IBM Journal of Research and Development* **17**(1973)525-532.
- [4] E. Bernstein and U. Vazirani, Quantum complexity theory, *SIAM Journal on Computing* **26**(1997)1411-1473.
- [5] S. Betteli, T. Calarco and L. Serafini, Toward an architecture for quantum programming, arXiv:cs.PL/0103009 v2, 2001.
- [6] G. Birkhoff and J. von Neumann, The logic of quantum mechanics *Annals of Mathematics* **37**(1936)823-843.
- [7] O. Brunet and P. Jorrand, Dynamic quantum logic for quantum programs, *International Journal of Quantum Information* **2**(2004)45-54.
- [8] G. Bruns and J. Harding, Algebraic aspects of orthomodular lattices, in: B. Coecke, D. Moore and A. Wilce (eds.), *Current Research in Operational Quantum Logic: Algebras, Categories, Languages*, Kluwer, Dordrecht, 2000, pp. 37-65.
- [9] W. Cheng and J. Wang, Grammar theory based on quantum logic, *International Journal of Theoretical Physics* **42**(2003)1677-1691.
- [10] J. I. Cirac and P. Zoller, Quantum computation with cold trapped ions, *Physical Review Letters* **74**(1995)4091-4094.
- [11] J. P. Crutchfield and C. Moore, Quantum automata and quantum grammar, *Theoretical Computer Science* **237**(2000)275-306.
- [12] M. L. Dalla Chiara, Some meta-logical pathologies of quantum logic, in: E. Beltrametti and B. C. van Fraassen (eds.), *Current Issues in Quantum Logics*, Plenum, New York, 1981, pp. 147-159.
- [13] M. L. Dalla Chiara, Quantum logic, in: D. Gabbay and E. Guentner (eds.), *Handbook of Philosophical Logic, volume III: Alternatives to Classical Logic*, D. Reidel Publishing Company, Dordrecht, 1986, pp. 427-469.

- [14] M. L. Dalla Chiara and R. Giuntini, Quantum logic, in: D. Gabbay and E. Guenther (eds.), *Handbook of Philosophical Logic, volume 6*, Kluwer, Dordrecht, 2002, pp. 129-228.
- [15] M. L. Dalla Chiara, R. Giuntini and R. Leporini, Logics for quantum computation, *International Journal of Quantum Information* **3**(2005)293-337.
- [16] M. L. Dalla Chiara, R. Giuntini and R. Greechie, *Reasoning in Quantum Theory: Sharp and Unsharp Quantum Logics*, Kluwer, Dordrecht, 2004.
- [17] D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer, *Proc. Roy. Soc. Lond.* **A400**(1985)97-117.
- [18] D. Deutsch, Quantum computational networks, *Proceedings of The Royal Society of London.* **A425**(1989)73-90.
- [19] J. Dieudonne, The current trend of pure mathematics, *Advances in Mathematics* **27**(1978)235-255.
- [20] G. Domenech and H. Freytes, Contextual logic for quantum systems, *Journal of Mathematical Physics* **46**(2005) 012102.
- [21] A. Dvurevcenskij, *Gleason's Theorem and Its Applications*, Kluwer, Dordrecht, 1993.
- [22] A. Dvurevcenskij, S. Pulmannova and K. Svozil, Partition logics, orthoalgebras and automata, *Helvetica Physica Acta* **68**(1995)407-428.
- [23] S. Eilenberg, *Automata, Languages, and Machines, volume A*, Academic Press, New York, 1974.
- [24] R. P. Feynman, Simulating physics with computers, *International Journal of Theoretical Physics* **21**(1982)467-488.
- [25] R. P. Feynman, Quantum mechanical computers, *Foundations of Physics* **16**(1986)507-531.
- [26] P. D. Finch, Quantum logic as an implication algebra, *Bulletin of Australian Mathematical Society* **2**(1970)101-106.
- [27] S. J. Gay and R. Nagarajan, Communicating quantum processes, in: P. Selinger (ed.), *Proceedings of the 2nd International Workshop on Quantum Programming Languages, July 12-13, 2004, Turku, Finland*, pp. 91-107.
- [28] L. K. Grover, A fast quantum mechanical algorithm for database search, in: *Proceedings of the 28th ACM STOC*, 1996, pp. 212-219.
- [29] S. Gudder, Basic properties of quantum automata, *Foundations of Physics* **30**(2000)301-319.

- [30] S. Gudder, Quantum computational logic, *International Journal of Theoretical Physics* **42**(2003)39-47.
- [31] W. Hatcher, *The Logical Foundations of Mathematics*, Pergamon, Oxford, 1982.
- [32] A. Heyting, *Axiomatic Projective Geometry*, North-Holland, Amsterdam, 1963.
- [33] H. Hodes, Three-valued logics - an introduction, a comparison of various logical lexica, and some philosophical remarks, *Annals of Pure and Applied Logic* **43**(1989)99-145.
- [34] K. Husimi, Studies on the foundations of quantum mechanics I, *Proceedings of the Physicomathematical Society of Japan* **19**(1937)766-789.
- [35] L. Herman, E. Marsden and R. Piziak, Implication connectives in orthomodular lattices, *Notre Dame Journal of Formal Logic* **16**(1975)305-328.
- [36] T. Hoare and R. Milner (eds.), *Grand Challenges in Computing Research*, <http://www.ukerc.org.uk/gcresearch.pdf>.
- [37] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, 1979.
- [38] G. Kalmbach, Orthomodular logic, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **20**(1974)395-406.
- [39] G. Kalmbach, *Orthomodular Lattices*, Academic Press, London, 1983.
- [40] A. Kondacs and J. Watrous, On the power of quantum finite state automata, in: *Proc. of the 38th Annual Symposium on Foundations of Computer Science*, 1997, pp. 65-75.
- [41] D. C. Kozen, *Automata and Computability*, Springer-Verlag, New York, 1997.
- [42] M. Lalire and P. Jorrand, A process algebraic approach to concurrent and distributed quantum computation: operational semantics, in: P. Selinger (ed.), *Proceedings of the 2nd International Workshop on Quantum Programming Languages, July 12-13, 2004, Turku, Finland*, pp. 109-126.
- [43] S. Lloyd, A potentially realizable quantum computer, *Science* **261**(1993)1569-1571.
- [44] R. Q. Lu and H. Zheng, Lattices of quantum automata, *International Journal of Theoretical Physics* **42**(2003)1425-1449.
- [45] R. Q. Lu and H. Zheng, Pumping lemma for quantum automata, *International Journal of Theoretical Physics* **43**(2004)1191-1217.

- [46] J. Malinowski, The deduction theorem for quantum logic - some negative results, *Journal of Symbolic Logic* **55**(1990)615-625.
- [47] P. Mateus and A. Sernadas, Exogenous quantum logic. In W. A. Carnielli, F. M. Dionisio, and P. Mateus (eds.), *Proceedings of CombLog'04, Workshop on Combination of Logics: Theory and Applications, Lisboa, Portugal, 2004*, pp. 141–149.
- [48] A. Mostowski, *Thirty Years of Foundational Studies: Lectures on the Development of Mathematical Logic and the Study of the Foundations of Mathematics in 1930-1964*, Acta Philosophica Fennica 17, Helsinki, 1965.
- [49] J. von Neumann, Quantum logics (strict- and probability-logics), summarized in: J. von Neumann, *Collected Works, vol. IV*, Macmillan, New York, 1962.
- [50] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, 2000.
- [51] H. Nishimura, Some connections between Heyting valued set theory and algebraic geometry—prolegomena to intuitionistic algebraic geometry, *Publications of Research Institute for Mathematical Sciences* **23**(1987)501-515.
- [52] H. Nishimura, Heyting valued considerations on some fundamental existence theorems in modern analysis, *Publications of Research Institute for Mathematical Sciences* **24**(1988)621–637.
- [53] B. Ömer, *A Procedural Formalism for Quantum Computing*, Master's thesis, Department of Theoretical Physics, Technical University of Vienna, 1998.
- [54] D. W. Qiu, Automata and grammars theory based on quantum logic, *Journal of Software* **14**(2003)23-27.
- [55] D. W. Qiu, Automata theory based on quantum logic: some characterizations, *Information and Computation* **190**(2004)179-195.
- [56] M. O. Rabin and D. S. Scott, Finite automata and their decision problems, *IBM Journal of Research and Development* **3**(1959)115-125.
- [57] J. P. Rawling and S. A. Selesnick, Orthologic and quantum logic: models and computational elements, *Journal of the ACM* **47**(2000)721-751.
- [58] L. Román and B. Rumbos, Quantum logic revisited, *Foundations of Physics* **21**(1991)727-734.
- [59] L. Román and R. E. Zuazua, Quantum implication, *International Journal of Theoretical Physics* **38**(1999)793-797.
- [60] J. B. Rosser and A. R. Turquette, *Many-Valued Logics*, North-Holland, Amsterdam, 1952.

- [61] J. W. Sanders and P. Zuliani, Quantum programming, in: *Mathematical of Program Construction*, LNCS 1837, 2000, pp. 80-99.
- [62] M. Schaller and K. Svozil, Partition logics of automata, *Nuovo Cimento B* **109**(1994)167–176.
- [63] M. Schaller and K. Svozil, Automaton partition logic versus quantum logic, *International Journal of Theoretical Physics* **34**(1995)1741–1749.
- [64] M. Schaller and K. Svozil, Automaton logic, *International Journal of Theoretical Physics* **35**(1996)911–940.
- [65] K. -G. Schlesinger, Toward quantum mathematics. I. From quantum set theory to universal quantum mechanics, *Journal of Mathematics Physics* **40**(1999)1344-1358.
- [66] S. A. Selesnick, Foundation for quantum computing, *International Journal of Theoretical Physics* **42**(2003)383-426.
- [67] P. Selinger, Towards a quantum programming language, *Mathematical Structures in Computer Science* **14**(2004)527-586.
- [68] P. W. Shor, Polynomial-time algorithm for prime factorization and discrete logarithms on quantum computer, in: *Proc. 35th Annual Symp. on Foundations of Computer Science, Santa Fe*, IEEE Computer Society Press, 1994.
- [69] P. W. Shor, Why haven't more quantum algorithms been found? *Journal of the ACM* **50**(2003)87-90.
- [70] K. Svozil, *Quantum Logic*, Springer-Verlag, Berlin, 1998.
- [71] G. Takeuti, *Two Applications of Logic to Mathematics*, Princeton University Press, New Jersey, 1978.
- [72] G. Takeuti, Quantum set theory, in: E. Beltrametti and B. C. van Fraassen (eds.), *Current Issues in Quantum Logics*, Plenum, New York, 1981, pp. 303-322.
- [73] G. Takeuti, Boolean simple groups and Boolean simple rings, *Journal of Symbolic Logic* **53**(1988)160-173.
- [74] S. Titani and H. Kozawa, Quantum set theory, *International Journal of Theoretical Physics* **42**(2003)2575-2602.
- [75] K. Tokuo, Quantum number theory, *International Journal of Theoretical Physics* **43**(2004)2461-2481.
- [76] A. V. Tonder, A lambda calculus for quantum computation, *SIAM Journal on Computing* **33**(2004)1109-1135.

- [77] A. S. Troelstra and D. van Dalen, *Constructivism in Mathematics: An Introduction*, volume I, II, North-Holland, Amsterdam, 1988.
- [78] V. Vedral and M. B. Plenio, Basics of quantum computation, *Progress in Quantum Electronics* **22**:1(1998).
- [79] A. C. Yao, Quantum circuit complexity, *Proc. of the 34th Ann. IEEE Symp. on Foundations of Computer Science*, 1993, pp. 352-361.
- [80] M. S. Ying, A new approach for fuzzy topology (I), (II), (III), *Fuzzy Sets and Systems*, **39**(1991)303-321; **47** (1992)221-232; **55**(1993)193-207.
- [81] M. S. Ying, Fuzzifying topology based on complete residuated lattice-valued logic (I), *Fuzzy Sets and Systems* **56**(1993)337-373.
- [82] M. S. Ying, Fuzzy tology based on residuated lattice-valued logic, *Acta Mathematica Sinica (English Series)* **17**(2001)89-102.
- [83] M. S. Ying, Automata theory based on quantum logic (I), (II), *International Journal of Theoretical Physics* **39**(2000)985-995; 2545-2557.
- [84] M. S. Ying, A theory of computation based on quantum logic (I), *Theoretical Computer Science* **344**(2005)134-207 or see <http://arxiv.org/abs/cs.LO/0403041>.