Chapter 19
Partition Logics, Finite Automata and Generalized Urn Models

19.1 Modelling Complementarity by Finite Partitions

Complementarity was first encountered in quantum mechanics. It is a phenomenon also understandable in classical terms; and although *"it's not a complicated idea but it's an idea that nobody would ever think of"* in analogy to entanglement [287] one might say what follows we shall present finite deterministic models featuring complementarity. The type of complementarity discussed in this chapter grew out of an attempt to understand quantum complementarity by some finite, deterministic, quasi-classical (automaton) model [373].

We shall do this by sets of partitions L of a given set with more than two elements. Suppose one identifies arbitrary elements $\{x_1, \ldots, x_k\}$ of some partition with the proposition *"The properties* x_1, *or*, ..., or x_k *are true."* Then each partition in L can be associated with a Boolean algebra or, synonymously, with a context, or block. Arbitrary partitions of L can be intertwined or pasted together [249, 263, 300, 376] in their common elements. This pasting construction yields a *partition logic*.

19.2 Generalized Urn and Automata Models

For the sake of getting a better intuition of partition logic and their relation to complementarity, two quasi-classical models will be discussed: (i) generalized urn models [577, 578] or, equivalently [506, 511], (ii) the (initial) state identification problem of finite deterministic automata [104, 184, 373, 446, 499] which are in an unknown initial state.

Both quasi-classic examples mimic complementarity to the extent that even quasi-quantum cryptography can be performed with them [509] as long as one sticks to the rules (limiting measurements to certain types), and as long as value indefiniteness is not a feature of the protocol [38, 519], that is, for instance, the Bennett and Brassard

K. Svozil, *Physical (A)Causality*, Fundamental Theories of Physics 192,
https://doi.org/10.1007/978-3-319-70815-7_19

1984 protocol [56] can be implemented with generalized urn models, whereas the Ekert protocol [198] cannot.

19.2.1 Automaton Models

A (Mealy type) automaton $\mathcal{A} = \langle S, I, O, \delta, \lambda \rangle$ is characterized by the set of states S, by the set of input symbols I, and by the set of output symbols O. $\delta(s, i) = s'$ and $\lambda(s, i) = o$, $s, s' \in S$, $i \in I$ and $o \in O$ represent the transition and the output functions, respectively. The restriction to Mealy automata is for convenience only.

The *(initial) state identification problem* for finite deterministic (Mealy) automata is the following: suppose one is presented with a (blackbox containing a) *single copy* of a finite deterministic automaton whose specifications are completely given with the exception of the state it is initially in: find that initial state by the input/output analysis of experiments with that automaton.

Then, as already pointed out by Moore, *"there exists a [[finite and deterministic]] machine such that any pair of its states are distinguishable, but there is no simple experiment which can determine what state the machine was in at the beginning of the experiment"* [373, Theorem 1, p. 138].

19.2.2 Generalized Urn Models

Wright's generalized urn model can be sketched by considering black balls with symbols in different colours drawn simultaneously on it. Perception of these colours are all "exclusive" or "complementary" by assuming that one looks at the ball with (coloured) glasses which are capable of transmitting only a single colour. Therefore, only the symbol in the respective colour is visible; all the symbols in different colours merge with the black background and are therefore unrecognizable. Suppose there are a lot of balls of many types (with various colours and an equal number of symbols per colour) in an urn. The question or task is this: Suppose one single ball is drawn from that urn; what is this particular type of ball or "ball state?"

Formally, a generalized urn model $\mathcal{U} = \langle U, C, L, \Lambda \rangle$ is characterized as follows. Consider an ensemble of balls with black background colour. Printed on these balls are some colour symbols from a symbolic alphabet L. The colours are elements of a set of colours C. A particular ball type is associated with a unique combination of mono-spectrally (no mixture of wavelength) coloured symbols printed on the black ball background. Let U be the set of ball types. We shall assume that every ball contains just one single symbol per colour. (Not all conceivable types of balls; i.e., not all colour/symbol combinations, may be present in this ensemble, though.)

Let $|U|$ be the number of different types of balls, $|C|$ be the number of different mono-spectral colours, $|L|$ be the number of different output symbols.

Consider the deterministic "output" or "lookup" function $\Lambda(u, c) = v$, $u \in U$, $c \in C$, $v \in L$, which returns one symbol per ball type and colour. One interpretation of this lookup function Λ is as follows. Consider a set of $|C|$ eyeglasses build from filters for the $|C|$ different colours. Let us assume that these mono-spectral filters are "perfect" in that they totally absorb light of all other colours but a particular single one. In that way, every colour can be associated with a particular eyeglass and vice versa.

19.2.3 Logical Equivalence for Concrete Partition Logics

The following considerations (largely based on [506, 511]) apply only to partition logics which have "enough" – that is, a *separating* set of – two-valued states. A logic L has a separating set of two-valued states if for every $a, b \in L$ with $a \neq b$ there is a two-valued state s such that $s(a) \neq s(b)$; that is, different propositions are distinguishable by some state [523].

The connection between those toy models and partition logics can be achieved by "inverting" the set of two-valued states as follows.

1. In the first step, every atom of this lattice is indexed or labelled by some natural number, starting from "1" to "n", where n stands for the number of lattice atoms. The set of atoms is denoted by $A = \{1, 2, \ldots, n\}$.
2. Then, all two-valued states of this lattice are labelled consecutively by natural numbers, starting from "v_1" to "v_r", where r stands for the number of two-valued states. The set of states is denoted by $V = \{v_1, v_2, \ldots, v_r\}$.
3. Now partitions are defined as follows. For every atom, a set is created whose members are the index numbers or "labels" of the two-valued states which are "true" or take on the value "1" on this atom. More precisely, the elements $p_i(a)$ of the partition $\mathcal{P}_j$ corresponding to some atom $a \in A$ are defined by

   $$p_i(a) = \{k \mid v_k(a) = 1,\ v_k \in V\}\,.$$

   The partitions are obtained by taking the unions of all p_i which belong to the same subalgebra $\mathcal{P}_j$. That the corresponding sets are indeed partitions follows from the properties of two-valued states: two-valued states (are "true" or) take on the value "1" on just one atom per subalgebra and ("false" or) take on the value "0" on all other atoms of this subalgebra.
4. Let there be t partitions labelled by "1" through "t". The partition logic is obtained by a pasting of all partitions $\mathcal{P}_j$, $1 \leq j \leq t$.
5. In the following step, a corresponding generalized urn model or automaton model is obtained from the partition logic just constructed.

   a. A generalized urn model is obtained by the following identifications (see also [577, p. 271]).

i. Take as many ball types as there are two-valued states; i.e., r types of balls.
ii. Take as many colours as there are subalgebras or partitions; i.e., t colours.
iii. Take as many symbols as there are elements in the partition(s) with the maximal number of elements; i.e., $\max_{1\le j\le t} |\mathcal{P}_j| \le n$. To make the construction easier, we may just take as many symbols as there are atoms; i.e., n symbols. (In most cases, much less symbols will suffice). Label the symbols by s_l. Finally, take r "generic" balls with black background. Now associate with every measure a different ball type. (There are r two-valued states, so there will be r ball types.)
iv. The ith ball type is painted by coloured symbols as follows: Find the atoms for which the ith two-valued state v_i is 1. Then paint the symbol corresponding to every such lattice atom on the ball, thereby choosing the colour associated with the subalgebra or partition the atom belongs to. If the atom belongs to more than one subalgebra, then paint the same symbol in as many colours as there are partitions or subalgebras the atom belongs to (one symbol per subalgebra).

This completes the construction.

b. A Mealy automaton is obtained by the following identifications (see also [499, pp. 154–155]).
i. Take as many automaton states as there are two-valued states; that is, r automaton states.
ii. Take as many input symbols as there are subalgebras or partitions; i.e., t symbols.
iii. Take as many output symbols as there are elements in the partition(s) with the maximal number of elements (plus one additional auxiliary output symbol "$*$", see below); i.e., $\max_{1\le j\le t} |\mathcal{P}_j| \le n+1$.
iv. The output function is chosen to match the elements of the state partition corresponding to some input symbol. Alternatively, let the lattice atom $a_q \in A$ must be an atom of the subalgebra corresponding to the input i_l. Then one may choose an output function such as

$$\lambda(v_k, i_l) = \begin{cases} a_q & \text{if } v_k(a_q) = 1 \\ * & \text{if } v_k(a_q) = 0 \end{cases}$$

with $1 \le k \le r$ and $1 \le l \le t$. Here, the additional output symbol "$*$" is needed.
v. The transition function is r–to–1 (e.g., by $\delta(s, i) = s_1, s, s_1 \in S, i \in I$), i.e., after one input the information about the initial state is completely lost.

This completes the construction.

19.3 Some Examples

The universe of possible partition logics [184, 446, 499, 511] is huge; and so are the conceivable probability measures [521] on them. In what follows we shall restrict our attention to partition logics containing partitions with equal numbers of elements.

19.3.1 Logics of the "Chinese Lantern Type"

Let us, for the sake of illustration, just mention as an example a set of partitions of the set $\{1, 2, 3\}$:

$$L = \{\{\{1\}, \{2, 3\}\}, \{\{1, 3\}, \{2\}\}, \{\{1, 2\}, \{3\}\}\}\,. \tag{19.1}$$

The term "{1}" corresponds to the proposition "1 is true." Every partition forms a 2-atomic Boolean subalgebra. It results in three Boolean algebras "spanned" by the atoms $\{1\}, \text{not}(\{1\}) = \{2, 3\}, \{2\}, \text{not}(\{2\}) = \{1, 3\}$, and $\{3\}, \text{not}(\{3\}) = \{1, 2\}$, which are not intertwined and thus form a *horizontal sum* of three Boolean subalgebras 2^3. This is equivalent to a quantum logic of, say, spin-$\frac{1}{2}$ particles whose spin is measured along three distinct directions [501].

Complementarity is obtained by realizing that one has to make choices: each choice of a particular partition corresponds to a type of measurement made. The set of (sometimes intertwined) partitions represents the "universe of conceivable measurements."

19.3.2 (Counter-)Examples of Triangular Logics

The propositional structure depicted in Fig. 19.1(i) has no two-valued (admissible [3, 5, 6]) state: The supposition that one element is "1" forces the remaining two to be "0," thus leaving the "adjacent" block without a "1" (there cannot be only zeroes in a context). This means that it has no representation as a quasi-classical partition logic.

The logic depicted in Fig. 19.1(ii) has sufficiently many (indeed four) two-valued measures to be representable by a partition logic [519]. Indeed, a concrete partition logic obtained by the earlier construction based on the inversion of the 4 two-valued states is

$$L = \{\{\{1\}, \{2, 4\}, \{3\}\}, \{\{2\}, \{3, 4\}, \{1\}\}, \{\{3\}, \{1, 4\}, \{2\}\}\}\,. \tag{19.2}$$

The propositional structure depicted in Fig. 19.1(iii) is too tightly interlinked to be representable by a partition logic – it allows only one two-valued state and thus has no separating set of two-valued states.

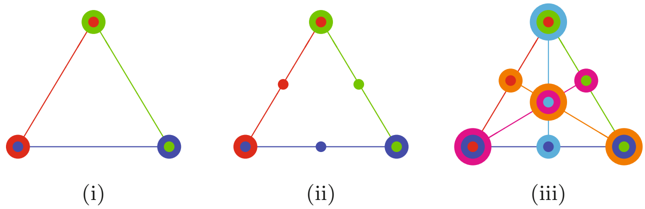

Fig. 19.1 Orthogonality diagrams representing tight triangular pastings of two- and three-atomic contexts

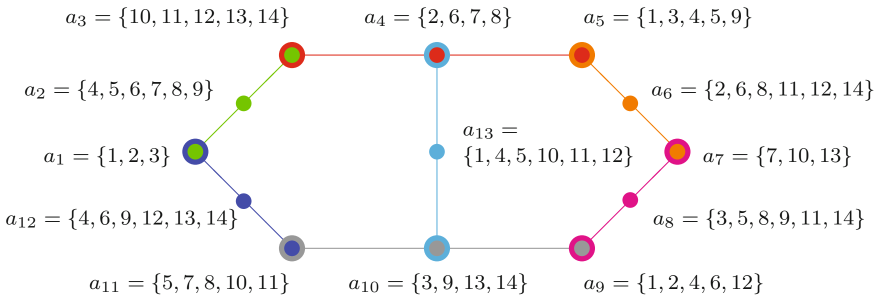

Fig. 19.2 Greechie diagram of automaton partition logic with a nonfull set of dispersion-free measures

19.3.3 Generalized Urn Model of the Kochen–Specker "Bug" Logic

Another example [506, 507, 511] is a logic which is already mentioned by Kochen and Specker [314] (this is a subgraph of their Γ_1 discussed in Sect. 12.9.8.4) whose automaton partition logic is depicted in Fig. 19.2. There are 14 dispersion-free states which are listed in Table 12.4. The associated generalized urn model is listed in Table 19.1.

19.3.4 Kochen–Specker Type Logics

With regards to quantum logic, partition logics share some common features but lack others. For instance, not all partition logics can be represented as sublogics of some quantum logic: as a counterexample take the partition logic depicted in Fig. 19.1(ii), which has no representation in $\mathbb{R}^3$. The central concern here is representability: since atoms in quantum logics can be identified with nonzero vectors or their associated projectors, the partition logic needs to have a geometric interpretation (embedding in vector space) preserving or rather representing the partition logical structure.

Table 19.1 (a) Dispersion-free states of the Kochen–Specker "bug" logic with 14 dispersion-free states and (b) the associated generalized urn model (all blank entries "*" have been omitted)

# v_r and ball type	(a) lattice atoms													(b) colors						
	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	c1	c2	c3	c4	c5	c6	c7
1	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1
2	1	0	0	1	0	1	0	0	1	0	0	0	0	1	2	2	1	1	1	2
3	1	0	0	0	1	0	0	1	0	1	0	0	0	1	1	1	2	2	1	3
4	0	1	0	0	1	0	0	0	1	0	0	1	1	2	1	1	1	1	2	1
5	0	1	0	0	1	0	0	1	0	0	1	0	1	2	1	1	2	3	3	1
6	0	1	0	1	0	1	0	0	1	0	0	1	0	2	2	2	1	1	2	2
7	0	1	0	1	0	0	1	0	0	0	1	0	0	2	2	3	3	3	3	2
8	0	1	0	1	0	1	0	1	0	0	1	0	0	2	2	2	2	3	3	2
9	0	1	0	0	1	0	0	1	0	1	0	1	0	2	1	1	2	2	2	3
10	0	0	1	0	0	0	1	0	0	0	1	0	1	3	3	3	3	3	3	1
11	0	0	1	0	0	1	0	1	0	0	1	0	1	3	3	2	2	3	3	1
12	0	0	1	0	0	1	0	0	1	0	0	1	1	3	3	2	1	1	2	1
13	0	0	1	0	0	0	1	0	0	1	0	1	0	3	3	3	3	2	2	3
14	0	0	1	0	0	1	0	1	0	1	0	1	0	3	3	2	2	2	2	3

Nevertheless, all finite sublogics of quantum logics with a separating set of two-valued states are equivalent to partition logics [139].

On the other hand, the Kochen–Specker theorem (cf. Sect. 12.9.8.7 on p. 97; in particular, the quantum sublattice depicted in Fig. 12.6) asserts that there exist sublogics of quantum logics which have no two-valued state at all. As has already been noted earlier, in a very precise and formal way, this can be identified with either contextuality [517, 518] or with value indefiniteness [286, 401].

This is all "bad news for partition logics" because although these quantum mechanical sublogics can be embedded in some (even low-dimensional) vector space, they have no two-valued state at all – alas, a separating set of two-valued state would be needed for a construction or characterization of any partition logics. Indeed it is even possible to show that, with reasonable side assumptions such as noncontextuality, there exist constructive proofs demonstrating that there is no value definiteness – that is, no two-valued state – beyond *a single proposition* and its negation [3, 5, 6] (cf. Sect. 12.9.8.7 on p. 97). Whether or not partition logics have empirical relevance remains an open question.