International Journal of Theoretical Physics, Vol. 44, No. 7, July 2005 (© 2005)
DOI: 10.1007/s10773-005-7052-0

Logical Equivalence Between Generalized Urn Models and Finite Automata

Karl Svozil[1]

Received May 23, 2004; accepted May 28, 2004

To every generalized urn model there exists a finite (Mealy) automaton with identical propositional calculus. The converse is true as well.

KEY WORDS: computational complementarity; automation logic; generalized urn models.

1. INTRODUCTION OF CONCEPTS

In what follows we shall explicitly and constructively demonstrate the equivalence of the empirical logics (i.e., the propositional calculi) associated with the generalized urn models (GUM) suggested by Wright (1978, 1990), and automaton partition logics (APL) (Svozil, 1993, 1998; Schaller and Svozil, 1996; Dvurečenskij *et al*., 1995; Calude *et al*., 1997). (The result has been mentioned already in (Svozil, 1998, p. 145), but no proof has been given). The logical equivalence of automaton models (AM) with generalized urn models suggests that these logics are more general and "robust" with respect to changes of the particular model than could have been expected from the particular instances of their first appearance.

1.1. Generalized Urn Models

A generalized urn model $\mathcal{U} = \langle U, C, L, \Lambda \rangle$ is characterized as follows. Consider an ensemble of balls with black background color. Printed on these balls are some color symbols from a symbolic alphabet L. The colors are elements of a set of colors C. A particular ball type is associated with a unique combination of mono-spectrally (no mixture of wavelength) colored symbols printed on the black ball background. Let U be the set of ball types. We shall assume that every

[1] Institut für Theoretische Physik, University of Technology Vienna, Wiedner Hauptstraße 8-10/136, A-1040 Vienna, Austria; e-mail: svozil@tuwien.ac.at.

ball contains just one single symbol per color. (Not all types of balls; i.e., not all color/symbol combinations, may be present in the ensemble, though).

Let $|U|$ be the number of different types of balls, $|C|$ be the number of different mono-spectral colors, $|L|$ 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 color. One interpretation of this lookup function Λ is as follows. Consider a set of $|C|$ eyeglasses build from filters for the $|C|$ different colors. Let us assume that these mono-spectral filters are "perfect" in that they totally absorb light of all other colors but a particular single one. In that way, every color can be associated with a particular eyeglass and vice versa.

When a spectator looks at a particular ball through such an eyeglass, the only operationally recognizable symbol will be the one in the particular color which is transmitted through the eyeglass. All other colors are absorbed, and the symbols printed in them will appear black and therefore cannot be differentiated from the black background. Hence the ball appears to carry a different "message" or symbol, depending on the color at which it is viewed.

An empirical logic can be constructed as follows. Consider the set of all ball types. With respect to a particular colored eyeglass, this set disjointly "decays" or gets partitioned into those ball types which can be separated by the particular color of the eyeglass. Every such partition of ball types can then be identified with a Boolean algebra whose atoms are the elements of the partition. A pasting of all of these Boolean algebras yields the empirical logic associated with the particular urn model.

1.2. 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.

A typical automaton experiment aims at an operational determination of an unknown initial state by the input of some symbolic sequence and the observation of the resulting output symbols. Every such input/output experiment results in a state partition in the following way. Consider a particular automaton. Every experiment on such an automaton which tries to solve the initial state problem is characterized by a set of input/output symbols as a result of the possible input/output sequences for this experiment. Every such distinct set of input/output symbols is associated with a set of initial automaton states which would reproduce that sequence. This state set may contain one or more states, depending on the ability of the experiment to separate different initial automaton states. A partitioning

of the automaton states is obtained if one considers a single input sequence and the variety of all possible output sequences (given a particular automaton). Stated differently: given a set of inputs, the set of automaton states decays into disjoint subsets associated with the possible output sequences. (All elements of a subset yield the same output on the same input).

This partition can then be identified with a Boolean algebra, with the elements of the partition interpreted as atoms. By pasting the Boolean algebras of the "finest" partitions together one obtains an empirical partition logic associated with the particular automaton. (The converse construction is also possible, but not unique; see below).

For the sake of simplicity, we shall assume that every experiment just deals with a single input/output combination. That is, the finest partitions are reached already after the first symbol. This does not impose any restriction on the partition logic, since given any particular automaton, it is always possible to construct another automaton with exactly the same partition logic as the first one with the above property.

More explicitly, given any partition logic, it is always possible to construct a corresponding automaton with the following specification: associate with every element of the set of partitions a single input symbol. Then take the partition with the highest number of elements and associate a single output symbol with any element of this partition. (There are then sufficient output symbols available for the other partitions as well). Different partitions require different input symbols; one input symbol per partition. The output function can then be defined by associating a single output symbol per element of the partition (associated with a particular input symbol). Finally, choose a transition function which completely looses the state information after only one transition; i.e., a transition function which maps all automaton state into a single one.

2. PROOF OF LOGICAL EQUIVALENCE

From the definitions and constructions mentioned in the previous sections it is intuitively clear that, with respect to the empirical logics, generalized urn models and finite automata models are equivalent. Every logic associated with a generalized urn model can be interpreted as an automaton partition logic associated with some (Mealy) automaton (actually an infinity thereof). Conversely, any logic associated with some (Mealy) automaton can be interpreted as a logic associated with some generalized urn model (an infinity thereof). We shall proof these claims by explicit construction. Essentially, the lookup function Λ and the output function λ will be identified. Again, the restriction to Mealy automata is for convenience only. The considerations are robust with respect to variations of finite input/output automata.

2.1. Direct Construction of AM From GUM

To define an APL associated with a Mealy automaton $\mathcal{A} = \langle S, I, O, \delta, \lambda \rangle$ from a GUM $\mathcal{U} = \langle U, C, L, \Lambda \rangle$, let $u \in U$, $c \in C$, $v \in L$, and $s, s' \in S$, $i \in I$, $o \in O$, and assume $|U| = |S|$, $|C| = |I|$, $|L| = |O|$. The following identifications can be made with the help of the bijections t_S, t_I and t_O:

$$\begin{aligned} & t_S(u) = s, t_I(c) = i, t_O(v) = o, \\ & \delta(s, i) = s_i \quad \text{for fixed } s_i \in S \text{ and arbitrary } s \in S,\ i \in I, \\ & \lambda(s, i) = t_O\big(\Lambda\big(t_S^{-1}(s), t_I^{-1}(i)\big)\big). \end{aligned} \tag{1}$$

More generally, one could use equivalence classes instead of a bijection. Since the input-output behavior is equivalent and the automaton transition function is trivially $|L|$-to-one, both entities yield the same propositional calculus.

2.2. Direct Construction of GUM From AM

Conversely, consider an arbitrary Mealy automaton $\mathcal{A} = \langle S, I, O, \delta, \lambda \rangle$ and its associated propositional calculus APL.

Just as before, associate with every single automaton state $s \in S$ a ball type u, associate with every input symbol $i \in I$ a unique color c, and associate with every output symbol $o \in O$ a unique symbol v; i.e., again $|U| = |S|$, $|C| = |I|$, $|L| = |O|$. The following identifications can be made with the help of the bijections τ_U, τ_C and τ_L:

$$\tau_U(s) = u,\ \tau_C(i) = c,\ \tau_L(o) = v,\ \Lambda(u, c) = \tau_L\big(\lambda\big(\tau_U^{-1}(u), \tau_C^{-1}(c)\big)\big). \tag{2}$$

A direct comparison of (1) and (2) yields

$$\tau_U^{-1} = t_S, \quad \tau_C^{-1} = t_I, \quad \tau_L^{-1} = t_O. \tag{3}$$

2.3. Schemes Using Dispersion-Free States

Another equivalence scheme uses the fact that both automaton partition logics and the logic of generalized urn models have a separating (indeed, full) set of dispersion-free states. (In what follows, the terms "dispersion-free state" "two-valued state" "valuation" "dispersion-free probability measure" are synonyms for measures which take on only the values 0 and 1. We thereby explicitly exclude dispersion-free measures which take on other values, such as $1/2$ and 0, as introduced by Wright (1978)). Stated differently, given a finite atomic logic with a separating set of states, then the enumeration of the complete set of dispersion-free states enables the explicit construction of generalized urn models and automaton logics whose logic corresponds to the original one.

This can be achieved by "inverting" the set of two-valued states as follows. (The method is probably best understood by considering the examples below). Let us start with an atomic logic with a separating set of states.

(i) In the first step, every atom of this lattice is labeled 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 $A = \{1, 2, \ldots, n\}$.

(ii) Then, all two-valued states of this lattice are labeled consecutively by natural numbers, starting from "m_1" to "m_r," where r stands for the number of two-valued states. The set of states is denoted by $M = \{m_1, m_2, \ldots, m_r\}$.

(iii) Now partitions are defined as follows. For every atom, a set is created whose members are the 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

$$p_i(a) = \{k \mid m_k(a) = 1,\ k \in M\}.$$

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.

(iv) Let there be t partitions labeled by "1" through "t." The partition logic is obtained by a pasting of all partitions $\mathcal{P}_j$, $1 \leq j \leq t$.

(v) In the following step, a corresponding GUM or automaton model is obtained from the partition logic just constructed.

(a) A GUM is obtained by the following identifications (see also (Wright, 1978, p. 271)).

- Take as many ball types as there are two-valued states; i.e., r types of balls.
- Take as many colors as there are subalgebras or partitions; i.e., t colors.
- Take as many symbols as there are elements in the partition(s) with the maximal number of elements; i.e., $\max_{1 \leq j \leq t} |\mathcal{P}_j| \leq n$. 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 v_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).
- The ith ball type is painted by colored symbols as follows: Find the atoms for which the ith two-valued state m_i is 1. Then paint

the symbol corresponding to every such lattice atom on the ball, thereby choosing the color 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 colors 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 (Svozil, 1993, pp. 154–155)).

- Take as many automaton states as there are two-valued states; i.e., r automaton states.
- Take as many input symbols as there are subalgebras or partitions; i.e., t symbols.
- 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., $\max_{1\leq j\leq t} |\mathcal{P}_j| \leq n+1$.
- 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

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

with $1 \leq k \leq r$ and $1 \leq l \leq t$. Here, the additional output symbol "$*$" is needed.

- 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.

2.4. Example 1: The Generalized Urn Logic L_{12}

In what follows, we shall illustrate the above constructions with a couple of examples. First, consider the generalized urn model

$$\langle \{u_1, \ldots, u_5\}, \{\text{red, green}\}, \{1, \ldots, 5\}, \Lambda \rangle$$

with Λ listed in Tables I and II.

The associated Mealy automaton can be directly constructed as follows. Take $t_S = t_O = \text{id}$, where id represents the identity function, and take $t_I(\text{red}) = 0$ and $t_I(\text{green}) = 1$, respectively. Furthermore, fix a (five×two)-to-one transition

Table I. Ball Types in Wright's Generalized Urn Model Wright (1990) (cf. Also [Svozil (1998) p. 143ff])

Ball type	Red	Green
1	1	3
2	1	4
3	2	3
4	2	4
5	5	5

function by $\delta(.,.) = 1$. The transition and output tables are listed in Tables I and II. Both empirical structures yield the same propositional logic L_{12}.

2.5. Example 2: The Automaton Partition Logic L_{12}

Let us start with an automaton whose transition and output tables are listed in Tables I and II and indirectly construct a logically equivalent GUM by using dispersion-free states. The first thing to do is to figure out all dispersion-free states of L_{12}. There are five of them, which we might write in vector form; i.e., in lexicographic order:

$$\begin{gathered} m_1 = (0,0,0,0,1),\ m_2 = (0,1,0,1,0),\ m_3 = (0,1,1,0,0), \\ m_4 = (1,0,0,1,0),\ m_5 = (1,0,1,0,0). \end{gathered} \tag{4}$$

Now define the following GUM as follows. There are two subalgebras with the atoms 1, 2, 5 and 3, 4, 5, respectively. Since there are five two-valued measures corresponding to five ball types. They are colored according to the coloring rules defined above and Λ as listed in Table III.

2.6. Example 3: GUM of the Kochen–Specker "Bug" Logic

Another, less simple example, is a logic which is already mentioned by Kochen and Specker (1967) (this is a subgraph of their Γ_1) whose automaton partition logic is depicted in Fig. 1. (It is called "bug" by Professor Specker (1999)

Table II. Transition and Output Table of an Associated Automaton Model

	δ					λ				
State	1	2	3	4	5	1	2	3	4	5
0	1	1	1	1	1	1	1	2	2	5
1	1	1	1	1	1	3	4	3	4	5

Table III. Representation of the Sign Coloring Scheme Λ

	Colors									
Ball type	c_1(Red)					c_2(Green)				
1	$*$	$*$	$*$	$*$	5	$*$	$*$	$*$	$*$	5
2	$*$	2	$*$	$*$	$*$	$*$	$*$	$*$	4	$*$
3	$*$	2	$*$	$*$	$*$	$*$	$*$	3	$*$	$*$
4	1	$*$	$*$	$*$	$*$	$*$	$*$	$*$	4	$*$
5	1	$*$	$*$	$*$	$*$	$*$	$*$	3	$*$	$*$

Note. "$*$" means no sign at all (Black) for the corresponding atom

because of the similar shape with a bug). There are 14 dispersion-free states which are listed in Table IV. The associated GUM is listed in Table IV.

3. DISCUSSION

We have explicitly demonstrated the logical equivalence of generalized urn models and and the logic of finite automata, both by a direct construction and by an indirect construction utilizing the set of two-valued states. This logical equivalence stresses the importance of these empirical structures.

GUMs and automata are capable to serve as models for particular types of lattices with a sufficient number of two-valued states (e.g., with a separating set of states). Yet, it is this very property which makes impossible the realization of other, more exotic states, which have no classical and not even a quantum mechanical counterpart. Take, as an example, the Wright state (Wright, 1978; Svozil, 1998) on the pentagon (or any n-agon, with odd $n > 3, n = 2k + 1, k = 2, 3, \ldots$) Greechie diagram with value $1/2$ on the five vertices and 0 on each middle atom (three atoms per subalgebra). The 11 two-valued measures suffice to generate GUMs

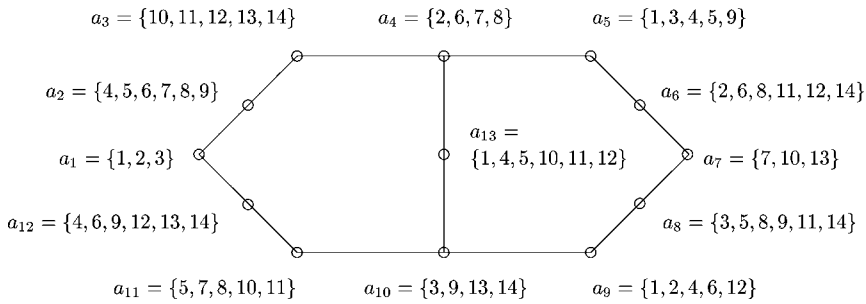

Fig. 1. Greechie diagram of automaton partition logic with a nonfull set of dispersion-free measures.

Table IV. Dispersion-Free States of the Kochen–Specker "Bug" Logic with 14 Dispersion-Free States and the Associated GUM (All Blank Entries "∗"Have Been Omitted).

	Lattice atoms													Colors						
m_r and ball type	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	c_1	c_2	c_3	c_4	c_5	c_6	c_7
1	1	0	0	0	1	0	0	0	1	0	0	0	1	1	5	5	9	9	1	13
2	1	0	0	1	0	1	0	0	1	0	0	0	0	1	4	6	9	9	1	4
3	1	0	0	0	1	0	0	1	0	1	0	0	0	1	5	5	8	10	3	10
4	0	1	0	0	1	0	0	0	1	0	0	1	1	2	5	5	9	9	12	13
5	0	1	0	0	1	0	0	1	0	0	1	0	1	2	5	5	8	11	11	13
6	0	1	0	1	0	1	0	0	1	0	0	1	0	2	4	6	9	9	12	4
7	0	1	0	1	0	0	1	0	0	0	1	0	0	2	4	7	7	11	11	4
8	0	1	0	1	0	1	0	1	0	0	1	0	0	2	4	6	8	11	11	4
9	0	1	0	0	1	0	0	1	0	1	0	1	0	2	5	5	8	10	12	10
10	0	0	1	0	0	0	1	0	0	0	1	0	1	3	3	7	7	11	11	13
11	0	0	1	0	0	1	0	1	0	0	1	0	1	3	3	6	8	11	11	13
12	0	0	1	0	0	1	0	0	1	0	0	1	1	3	3	6	9	9	12	13
13	0	0	1	0	0	0	1	0	0	1	0	1	0	3	3	7	7	10	13	10
14	0	0	1	0	0	1	0	1	0	1	0	1	0	3	3	6	8	10	12	10

and finite automata with that logical structure, but none such model realizes the Wright state.

ACKNOWLEDGMENTS

This manuscript has been prepared on the request of a friend and colleague Norbert Brunner who was considerably involved in the foundational stages of automaton partition logic.

REFERENCES

Calude, C., Calude, E., Svozil, K., and Yu, S. (1997). Physical versus computational complementarity I. *International Journal of Theoretical Physics* **36**(7), 1495–1523.

Dvurečenskij, A., Pulmannová, S., and Svozil, K. (1995). Partition logics, orthoalgebras and automata. *Helvetica Physica Acta* **68**, 407–428.

Kochen, S. and Specker, E. P. (1967). The problem of hidden variables in quantum mechanics. *Journal of Mathematics and Mechanics* **17**(1), 59–87. Reprinted in (Specker, 1990, pp. 235–263).

Schaller, M. and Svozil, K. (1996). Automaton logic. *International Journal of Theoretical Physics* **35**(5), 911–940.

Specker, E. (1990). *Selecta*. Birkhäuser Verlag, Basel.

Specker, E. (1999). (Private communication).

Svozil, K. (1993). *Randomness & Undecidability in Physics*, World Scientific, Singapore.

Svozil, K. (1998). *Quantum Logic*, Singapore, Springer.

Wright, R. (1978). The state of the pentagon. A nonclassical example. In *Mathematical Foundations of Quantum Theory*, A. R. Marlow, ed., Academic Press, New York, pp. 255–274.

Wright, R. (1990). Generalized urn models. *Foundations of Physics* **20**, 881–903.