Advanced Studies in Contemporary Mathematics, 6 (2003), No. 1, pp. 57-62

ALGEBRAIC STRUCTURES ARE LABELED TRANSITION SYSTEMS!

Apostolos Syropoulos

366, 28th October Str
GR-671 00 Xanthi, GREECE
e-mail: apostolo@obelix.ee.duth.gr

Abstract Any algebraic structure can be viewed as a labeled transition system and we demonstrate this thesis by representing MV-algebras as labeled transition systems. We do this by rigorously defining the representation mechanism. The generated LTS can be depicted by transition multi-graphs and we provide some simple examples.

2000 Mathematics Subject Classification. Primary 30D35; Secondary 13F99, 13C20
Keywords and phrases. Algebraic Structures, Labeled Transition Systems, Category Theory, Concurrency, and Multi-graphs

1 Introduction

In many instances it is useful to be able to view a particular mathematical structure in an unconventional way. For this purpose, we usually employ the mathematical *representation* of one mathematical structure in terms of another one. In other words, we employ the language of Category Theory (e.g., see [McL98]) to define collections of similar entities and transformations between them. And if this transformation has a number of properties, then we say that we have a representation of one mathematical structure in terms of another one. In this paper, we show that MV-algebras, in particular, and algebraic structures, in general, can be represented as labeled transition systems. We employ, MV-algebras merely as a demonstration tool.

1.1 Labeled Transition Systems

Labeled transition systems (or LTS, for short) are a frequently used model of concurrency [Mil99]. They consist of a set of states and set of transitions from one state to another. The thesis of this paper is that all algebraic structures can be represented as LTS and we show this by representing MV-algebras as LTS.

1.2 MV-algebras

MV-algebras are models of Łukasiewicz's infinite-valued propositional logic (see, e.g., [Mun00]). An MV-algebra is a set with an associative, commutative operation $\boxplus$, a neutral element $\mathbf{0}$, and a unary negation operation $\neg$. To pass from MV-algebras to Łukasiewicz's infinite-valued propositional logic, one writes $\neg x \to y$ instead of $x \boxplus y$ and rewrites every valid equation $x = y$ as a tautology $x \leftrightarrow y$.

1.3 Structure of the Paper

We start with an elementary introduction to MV-algebras and the basic theory of LTS. Then we describe the representation of MV-algebras as LTS and we study the properties of this representation. The generated LTS can be depicted by transition multi-graphs, and show how this can be done by two simple examples. Finally, we conclude with some remarks concerning our work.

2 Short Introduction to the theory of MV-algebras

In this section we define MV-algebras and homomorphisms between them.

Definition 2.1 An MV-algebra is a quadruple $\langle A, \boxplus, \neg, \mathbf{0}\rangle$, where $\boxplus$, $\neg$, and $\mathbf{0}$ are a binary, a unary and constant, respectively, having the following properties:

i) $x \boxplus (y \boxplus z) = (x \boxplus y) \boxplus z$,

ii) $x \boxplus y = y \boxplus x$,

iii) $x \boxplus \mathbf{0} = x$,

iv) $\neg\neg x = x$,

v) $x \boxplus \neg\mathbf{0} = \neg\mathbf{0}$, and

vi) $\neg(\neg x \boxplus y) \boxplus y = \neg(\neg y \boxplus x) \boxplus x$.

Usually, an MV-algebra $\langle A, \boxplus, \neg, \mathbf{0}\rangle$, is denoted by the name of its underlying set, e.g., A.

Definition 2.2 A function $h : A \to B$, between two MV-algebras A and B, is a *homomorphism* iff it has the following properties for each $x, y \in A$:

i) $h(\mathbf{0}_A) = \mathbf{0}_B$,

ii) $h(x \boxplus y) = h(x) \boxplus' h(y)$, and

iii) $h(\neg x) = \neg' h(x)$.

Since homomorphisms are actually functions, they compose the usual way. In addition, the identity homomorphism is just the identify function of the underlying set. So, we can construct a category with objects all the MV-algebras and with arrows the homomorphisms between them. We call this category **MV**.

3 MV-algebras and LTS

An LTS consists of a set of states, with an initial state, together with transitions between states which are labeled to specify the kind of events they represent.

Definition 3.1 A *LTS* is a structure (S, i, L, tran), where

- S is a set of *states* with *initial state* i,
- L is a set of *labels*, and
- $\mathrm{tran} \subseteq S \times L \times S$ is the *transition relation*. Usually a transition (s, a, s') is denoted $s \xrightarrow{a} s'$.

Given two LTS $T_0 = (S_0, i_0, L_0, \mathrm{tran}_0)$ and $T_1 = (S_1, i_1, L_1, \mathrm{tran}_1)$, a *morphism* $f : T_0 \to T_1$ is a pair $f = (\sigma, \lambda)$ where

- $\sigma : S_0 \to S_1$, is a function between sets of states, and
- $\lambda : L_0 \to_* L_1$, is a partial function between sets of labels such that

$$\sigma(i_0) = i_1 \qquad \text{and} \tag{1}$$

$$(s, a, s') \in \text{tran}_0 \Rightarrow (\sigma(s), \lambda(a), \sigma(s')) \in \text{tran}_{1*}. \tag{2}$$

Given a third LTS $T_2 = (S_2, i_2, L_2, \text{tran}_2)$ and a morphism $g = (\sigma', \lambda') : T_1 \to T_2$, their composite is the morphism $f \circ g = (\sigma \circ \sigma', \lambda \circ \lambda')$. Obviously, morphism composition is an associative operation since function composition is an associative operation. For an LTS $T_0 = (S_0, i_0, L_0, \text{tran}_0)$ the morphism $\text{id}_{T_0} = (\text{id}_{S_0}, \text{id}_{L_0})$, where id_{S_0} is the identity function of the set of states and id_{L_0} is the identity function of the set of labels, vacuously satisfy equations 1 and 2. It is now trivial to define the category **LTS** with objects the LTS and with arrows the morphisms between the LTS. We now proceed with the definition of the functor Ψ that maps MV-algebras to LTS. We start with object part of the functor:

Definition 3.2 Given an MV-algebra $\langle A, \boxplus, \neg, \mathbf{0}\rangle$, the functor Ψ maps A to the LTS $(A, \mathbf{0}, A, A_{\boxplus})$, where $(s, a, s') \in A_{\boxplus}$ iff $s \boxplus a = s'$.

And now we define the arrow part of Ψ:

Definition 3.3 If $h : A \to B$ is homomorphism between two MV-algebras A and B, then $\Psi(h) = (h, h)$.

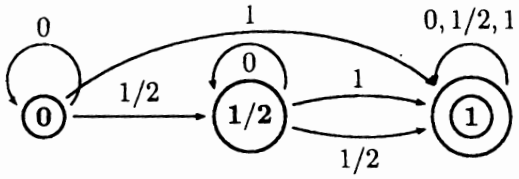
Theorem 3.1 *Functor Ψ is faithful and injective on objects.*

Proof. That the functor Ψ is injective on objects is obvious from its definition. Next, we prove that the functor is faithful. Suppose that A and B are two MV-algebras and that $h_1, h_2 : A \to B$ are two homomorphisms, then

$$\begin{aligned} \Psi(h_1) &= \Psi(h_2) \Rightarrow \\ (h_1, h_1) &= (h_2, h_2) \Rightarrow \\ h_1 &= h_2 \end{aligned}$$

Which proves that the Ψ is faithful. □

Example 3.1 If we consider the set $D = \{0, 1/2, 1\}$ and the operations $a \boxplus b = \min(1, a + b)$ and $\neg a = 1 - a$, then $\langle D, \boxplus, \neg, 0\rangle$ is an MV-algebra. The LTS system generated by $\Psi(D)$ is depicted below:

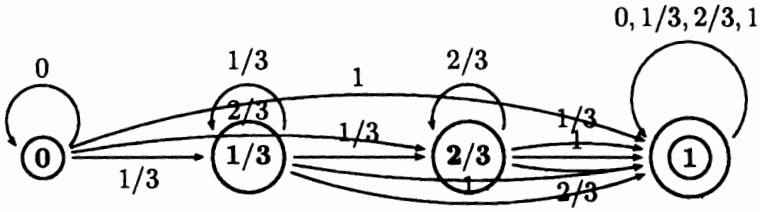


In the above (non-deterministic) automaton we have chosen **1** to be the accepting state. Note, that for any MV-algebra the generated LTS can have $\neg\mathbf{0}$ as its accepting state. Of course, we can choose any other node to be the accepting state, but $\neg\mathbf{0}$ is the only one to which there transitions from any other node, while there no transitions from this node to any other node. It is interesting to see what is the accepting language of this automaton. For reasons of clarity, we set $a = 0$, $b = 1/2$ and $c = 1$, then the language generated is $a^* \cdot (c \cdot ((a+b+c)^* \cdot \epsilon) + b \cdot (b^* \cdot (a \cdot ((a+b+c)^* \cdot \epsilon) + b \cdot ((a+b+c)^* \cdot \epsilon))))$.

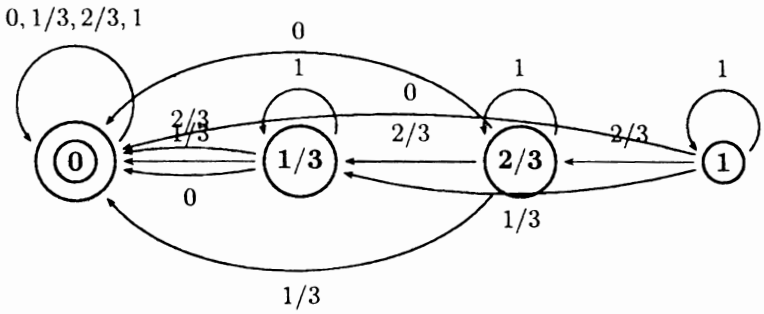
4 Discussion

Each MV-algebra is equipped with two additional operations: $x \odot y \stackrel{\text{def}}{=} \neg(\neg x \boxplus \neg y)$ and $x \ominus y \stackrel{\text{def}}{=} x \odot \neg y$. It is not difficult to generate an LTS from the $\odot$ operation of an MV-algebra. Actually, it is now a straightforward exercise to define a new functor $\Psi_\odot$ that will map a given MV-algebra A to the LTS $\Psi_\odot(A)$ having the property that $s \xrightarrow{a} s'$ iff $s \odot a = s'$. The arrow part of the functor is identical to that of the functor Ψ and of course $\Psi_\odot$ has the same properties as Ψ. (Note that $h(x \odot y) = h(x) \odot' hy$.)

Example 4.1 We consider the MV-algebra with underlying set the set $E = \{0, 1/3, 2/3, 1\}$ and the operations of example 3.1. Then the following transition graph depicts the LTS generated by this algebra:



Note, that it is possible to use the labels $\neg 1/3$ and $\neg 0$ instead of $2/3$ and 1 and so to have a input and output labels. The following transition graph depicts the automaton generated by $\Psi_\odot(E)$:



The above two automata have the same set of agents and of course the same set of labels. It is interesting to note that whenever there is a transition $p \xrightarrow{a} p'$ in the first LTS, then the transition $p' \xrightarrow{\neg a} p$ exists in the second LTS. However, it is not clear whether this observation can be generalized and whether there is any deep meaning in this similarity. But, it holds that this observation is valid at least for the MV-algebras that are sub-algebras of the algebra defined in the unit interval.

5 Conclusions

We have defined the functor Ψ by means of it we map every MV-algebra to a LTS. In general, LTS are a frequently used model of concurrency and so we have actually managed to make MV-algebras models of a particular class of concurrent systems. As we have already mentioned in the introduction, MV-algebras are a model of Lukasiewicz's infinite-valued propositional logic. So, we have LTS that are models of a particularly interesting logic. The connection between logic and concurrency is something that was evident only in "classical" logics like linear logic [Gir95]. The present work extends this connection to many-valued logics and hopefully will allow researchers to investigate further this connection.

References

[Gir95] Jean-Yves Girard. Linear logic: Its syntax and semantics. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, pages 1–42. Cambridge University Press, 1995. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.

[McL98] Saunders McLane. *Category Theory for the Working Mathematician.* Springer-Verlag, New York, second edition, 1998.

[Mil99] R. Milner. *Communicating and Mobile Systems: The π-calculus.* Cambridge University Press, 1999.

[Mun00] Daniele Mundici. MV-algebras and infinite-valued logic. In Iturrioz Luisa, Orlowska Ewa, and Turunen Esko, editors, *Atlas of Many-Valued Structures.* Dept. of Information Tecnology, Tempere Univ. of Technology, 2000. Mathematics Report 75.