Proceedings of ICTAMI
International Conference on Theory and Applications in Mathematics and Informatics
17^{th}–20^{th} of September, 2015, Alba-Iulia, Romania
Computer Science Section

pp. 141-146

SEVERAL REMARKS ON GENERATING A THEORETICAL MODEL OF MODAL LOGIC

B.D. Bucur

Abstract. The aim of this paper is to establish a connection between modal logics and labeled graphs, which is useful in solving the problem of *undeterminism.*

2010 *Mathematics Subject Classification*: 03B99, 05C78, 68R10.

Keywords: labeled graphs, transition systems, undeterminism.

1. Introduction

The most famous theories about modal logic are based on the model built by Saul Kripke which, in a restricted sense, refers to necessity and possibility.

The semantics of modal logics consists of a non-empty set G, whose elements are called *possible worlds,* a binary relation R between the elements of G called *accessibility relation* and a labeling function which describes every situation. Modal logic makes use of the modal operators $\Box$ (necessary) and $\Diamond$ (possible), see, e.g., [3].

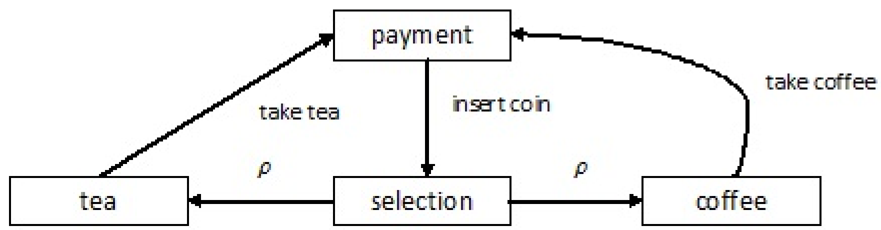
Definition 1 ([3])**.** *A Kripke model is a tuple* (S, R, L) *where* S *is a set of states (possible worlds),* R *an accessibility (transition) relation with* $R \subseteq S \times S$ *such that* $\forall s_1 \in S\ \exists s_2 \in S$ *with* $(s_1, s_2) \in R$ *and* $L : S \to 2^{AP}$ *a labeling function such that* $\forall s \in S$, $L(s)$ *represents all the atomic propositions true in* s *and* AP *is the set of atomic propositions.*

2. Transition systems

Transition systems are concepts used in computer science. They consist of states and transitions among them. The set of states can be countable or uncountable, and so can the set of transitions.

Definition 2 ([1])**.** *Formally, a transition system* ST *is a triple* $(S, A, \to)$*, where* S *is a set of states,* A *a set of actions and* $\to \subseteq S \times A \times S$ *is the transition relation.*

For example, we can consider a warm drinks vending machine, which sells coffee or tea.



$$S = \{\text{payment}, \ \text{tea}, \ \text{selection}, \ \text{coffee}\},$$

$$A = \{\text{take tea}, \ \rho, \ \text{take coffee}, \ \text{insert coin}\}.$$

We can consider the transition system ST, a tuple $(S, A, \rightarrow, P, L)$ where S is a set of states, A a set of actions, $\rightarrow \subseteq S \times A \times S$ is a transition relation, P a set of atomic propositions and $L : S \rightarrow 2^P$ a label function.

To a transition system we can associate a set of atomic propositions which depend on the properties taken into account. Thus we can obtain a variety of choices which a logical analysis is able to predict. From the point of view of transition mechanisms, the choice is arbitrary.

3. Undeterminism

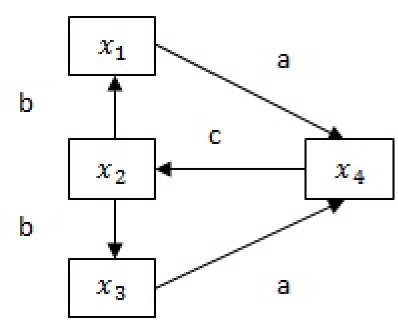
What is important for modeling transition systems is the *undeterminism,* which is more than a theoretical concept. It allows for freedom in modeling the computation systems. Intuitively, a transition system begins with an initial state and evolves to another state according to the accessibility relation. If from one state there can be more transitions, then the choice of the next state is undeterministic. That is, the result of the selection is not a priori known, therefore one can draw no conclusion regarding the probability with which a certain transition is chosen. The same aspect is met also in the case when there is not only one state, but a set of initial states, in which case the undeterministic factor plays a role. The analysis of these choices is useful in modeling conflict situations which may appear in case of processes executed in parallel, but also in modeling unknown interfaces (see [2]).

However, it is useful to take into account the observable, deterministic behavior, related to various observable notions. In this way, the determinism agrees with the executable actions which are observable, or it is related to labels and relies only on the atomic propositions which take place and are observable. To simplify, the atomic propositions of the system are statements on which the binary operation yes/no acts. If the names of the actions are not relevant, as transition represents an internal process, we can use symbols, or even omit them in certain cases.

4. Labeled graphs

Definition 3 ([4]). *A labeled graph is a tuple $LG = (S, E, T, f)$ where S is a finite set of elements representing the vertices of LG, E is a set of elements used to label the edges of the graph, T is a set of binary relations on S and $f : E \to T$ a surjective function.*

Remark 1. *In the graphical representation of this structure, the vertices are drawn as boxes which contain their names. An edge from $x_i \in S$ to $x_j \in S$ is labeled by $a \in E$ if and only if $(x_i, x_j) \in f(a)$.*



Example 1. *Consider*

- $S = \{x_1, x_2, x_3, x_4\}$;
- $L = \{a, b, c\}$;
- $T = \{\rho_1, \rho_2, \rho_3\}$;
- $\rho_1 = \{(x_1, x_4), (x_3, x_4)\}$;
- $\rho_2 = \{(x_2, x_1), (x_2, x_3)\}$;
- $\rho_3 = \{(x_4, x_2)\}$;

- $f(a) = \rho_1$;
- $f(b) = \rho_2$;
- $f(c) = \rho_3$.

5. Labeled graphs and transition systems

We notice the following:

1. A labeled graph can be interpreted as a transition system.
2. The states of ST are the vertices of the graph.
3. The actions of ST can be associated with the labels of the graph.

We define the labeled graph associated to a transition system, and denote this by $LG(ST)$, to be a tuple (S, E, T, f), where S is the set of states, E the set of actions, T the set of atomic propositions and $f : E \to T$ a surjective function.

Let $LG_1(ST) = (S_1, E_1, T_1, f_1)$ and $LG_2(ST) = (S_2, E_2, T_2, f_2)$ two labeled graphs associated to a transition system. Consider the function

$$g : S_1 \to S_2$$

and define

$$\overline{g} : 2^{S_1 \times S_1} \to 2^{S_2 \times S_2}$$

by

$$\overline{g}(\emptyset) = \emptyset$$

and

$$\overline{g}(R) = \{(x_1, x_2) \in S_2 \times S_2 : \exists (a_1, a_2) \in R \text{ with } g(a_i) = x_i \text{ for } i = 1, 2\}.$$

Definition 4. *We say that LG_1 is included in LG_2, and denote this by $LG_1 \subseteq LG_2$, if the following conditions hold:*

1. $E_1 \subseteq E_2$;
2. *there is an injective function $g : S_1 \to S_2$ for which*

$$\overline{g}(f_1(a) \subseteq f_2(a) \quad \forall a \in E_1.$$

Proposition 1. *The inclusion relation previously defined is reflexive and transitive.*

Proof. It is clear that $LG \subseteq LG$, therefore the relation is reflexive.

We assume now $LG_1 \subseteq LG_2$ and $LG_2 \subseteq LG_3$ and prove that $LG_1 \subseteq LG_3$. Since $LG_1 \subseteq LG_2$, there is an injective function $g_1 : S_1 \to S_2$ so that

$$\overline{g_1}(f_1(a)) \subseteq f_2(a) \quad \forall a \in E_1.$$

In a similar way, there is an injective function $g_2 : S_2 \to S_3$ so that

$$\overline{g_2}(f_2(a)) \subseteq f_3(a) \quad \forall a \in E_2.$$

Clearly, from $E_1 \subseteq E_2$ and $E_2 \subseteq E_3$ we get $E_1 \subseteq E_3$. It means that we can define an injective function $g_3 : S_1 \to S_3$ by $g_3 = g_2 \circ g_1$ such that

$$\overline{g_2}(\overline{g_1}(a)) \subseteq \overline{g_2}(f_2(a)) \quad \forall a \in E_1.$$

But $\overline{g_2}(f_2(a)) \subseteq f_3(a) \Rightarrow \overline{g_2}(\overline{g_1}(f_1(a))) \subseteq f_3(a)$, therefore

$$\overline{g_3}(f_1(a)) \subseteq f_3(a) \quad \forall a \in L_1,$$

hence $LG_1 \subseteq LG_3$.

Definition 5. *If $LG_1 \subseteq LG_2$ and $LG_2 \subseteq LG_1$ we say that LG_1 and LG_2 are isomorphic and denote this by $LG_1 \sim LG_2$.*

6. Conclusions

Modal logic studies reasonings which imply the use of the terms “necessary” and “possible”. In a broader sense, modal logic covers a family of logics with a set of norms and a variety of different symbols. Representing them through labeled graphs allows for a graphical representation which is useful in the theoretical analysis of their properties.

References

[1] C. Baier and J.-P. Katoen, *Principles of model checking*, The MIT Press Cambridge, London, 2008.

[2] P. Blackburn and J. van Benthem, *Modal logic: a semantic perspective*, 2010.

[3] S. Kripke, *Semantical considerations on modal logic*, 1963.

[4] N. Tandareanu, *Baze de cunostinte*, Universitatea din Craiova, 2004.

Beatrice Daniela Bucur
Department of Computer Science,
University of Pitesti,
Pitesti, Romania
email: *ciolan_b@yahoo.com*