Position papers of the Federated Conference on Computer Science and Information Systems, pp. 33–36 DOI: 10.15439/2017F548 ISSN 2300-5963 ACSIS, Vol. 12

Rough Sets for Trees of Executions

Krzysztof Pancerz
Department of Computer Science, Faculty of Mathematics and Natural Sciences
University of Rzeszów, Poland
Email: kpancerz@ur.edu.pl

***Abstract*—In the paper, we propose to use rough sets to express some properties (reachability of states) of systems whose underlying models of behaviour are trees of executions. By analogy with the modal operators of branching time temporal logics, we define positive regions, boundary regions, and negative regions of anticipations of distinguished states (states of interest) in the modelled systems. Instead of a temporal logic approach, we propose to use a set theoretic approach.**

I. Introduction

ROUGH sets [1] are an appropriate tool to deal with doubtful concepts in the universe U of discourse. A general idea of rough sets is to approximate a given set X of objects (states) of interest by other sets of objects (states), which are called the upper approximation $Upp(X)$ and the lower approximation $Low(X)$ of X, where $Low(X) \subseteq X \subseteq Upp(X)$. Approximation can be either exact (if $Low(X) = Upp(X)$) or rough (if $Low(X) \subset Upp(X)$). In terms of modal logics, the lower approximation can be identified with the necessity property, whereas the upper approximation can be identified with the possibility property (cf. [2]). Based on approximations, $Low(X)$ and $Upp(X)$, of the set X, the whole universe U of objects can be divided into three disjoint regions, the positive region $Pos(X)$, the negative region $Neg(X)$, and the boundary region $Bnd(X)$, where $Pos(X) = Low(X)$, $Neg(X) = U - Upp(X)$, and $Bnd(X) = Upp(X) - Low(X)$.

In [3] and [4], we proposed to use rough sets to describe some ambiguities in anticipation of states in systems whose dynamics is modelled by transition or timed transition systems. Analogously to rough approximation of sets, considered in rough set theory, we defined rough anticipation of states over transition (timed transition) systems. Anticipation of states was made via direct predecessor states of the anticipated ones. Therefore, this anticipation was called predecessor anticipation. We distinguished two kinds of anticipations, called the lower predecessor anticipation and the upper predecessor anticipation. Let X be a distinguished set of states we are interested to reach in a system whose behaviour is described by a transition system TS. The lower predecessor anticipation $LowPredAnt(X)$ consists of all states from which TS surely goes to the states in X as results of any events occurring at these states. It is necessary that each next state of any state from $LowPredAnt(X)$ is one of the states belonging to X. The upper predecessor anticipation consists of all states from which TS possibly goes to the states in X as results of some events occurring at these states. At least one next state of any state from $UppPredAnt(X)$ is the state belonging to X.

In this paper, we deal with a problem of identifying positive, negative and boundary regions of states of systems whose underlying models of behaviour are trees of executions. The proposed approach is patterned upon the temporal logic of branching time [5]. In this logic, the underlying model is mainly a tree of all possible computations. However, the presented approach is general and it can be applied to any system with the underlying tree model of behaviour. Instead of a temporal logic approach, we propose to use a set theoretic approach. We do not need to consider descriptions of system behaviours in terms of logical formulas. Therefore, we can build a tree of all possible paths of computations, paths of executions, paths of propagations, etc. Further, for simplicity, we will use a notion of a tree of executions. However, as it was mentioned earlier, any kind of dynamic actions can be considered (e.g. computations, executions, propagations, etc.). It is not necessary to identify nodes of trees with some propositions which can hold at these nodes. We can distinguish any states that should be reached, events that should happen, etc. Further, for simplicity, we will use a notion of a state. However, as it was mentioned earlier, any kind of entities can be considered (states, events, etc.).

In our approach, three kinds of anticipations of states, belonging to branches of a tree $\mathbf{T}$ of executions, are considered (see Section III). Let x be a state anticipating another states in a branch, we can distinguish:

- G-anticipation if $x \in T$ begins a branch consisting of states that are only the distinguished ones.
- F-anticipation if $x \in T$ begins a branch consisting of at least one state that is the distinguished one.
- X-anticipation if $x \in T$ begins a branch such that the next state in the branch is the distinguished one.

Because, a given state x can begin more than one branch, according to rough set theory, each kind of anticipation can be considered as either certain, possible, or impossible.

II. Temporal Logic Background

There is a variety of formal models of time. In instant-based models of time, the primitive temporal entities are time instants [6]. The flow of time is represented as a set of time instants with a binary relation of precedence on it. Two main types of instant-based models are usually considered:

1) Models with linear orderings of time instants, reflecting the idea that the time flow is a succession of time instants.

©2017, PTI

2) Models with partial orderings of time instants, reflecting the idea that the past is determined, while the future can be undetermined, branching into many possible time lines (there exist alternative futures).

With each time instant, a state (event, etc.) can be identified.

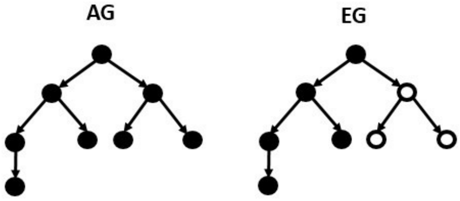

Fig. 1. AG and EG operators of branching time temporal logics

The term of temporal logic is broadly used to cover all approaches to representation and reasoning about time and temporal information within a logical framework [6]. In case of models with linear orderings of time instants, temporal logics are referred to as linear time temporal logics. Formulas of linear time temporal logics are interpreted over sequences of nodes corresponding to time instants. In case of models with partial orderings of time instants, temporal logics are referred to as branching time temporal logics. Formulas of branching time temporal logics are interpreted over trees of nodes corresponding to time instants.

Let T be a tree and s be a node in T. Let a be a proposition which can hold at some nodes in T. The following main modal operators are used in branching time temporal logics (cf. [5]):

- AGa holds at s if and only if a is true at all nodes of the subtree rooted at s (including s).
- EGa holds at s if and only if there is a path starting with s such that a is true at all nodes on this path.
- AFa holds at s if and only if on every path starting with s, there is some node at which a is true.
- EFa holds at s if and only if there is a path starting with s such that a is true at some node on this path.
- AXa holds at s if and only if a is true at every immediate successor of s.
- EXa holds at s if and only if a is true at some of immediate successor of s.

The meaning of the main operators of branching time temporal logics can be graphically explained as it is shown in Figures 1, 2, and 3. Filled circles represent nodes at which the proposition a is true. One can see that the first symbol (A or E) denotes quantification over branches, whereas the second symbol (G, F, or X) denotes quantification over states in the branches.

Temporal logics are used in various areas ranging from computer science (e.g. specification and verification of concurrent programs and systems), artificial intelligence (e.g. temporal representation and reasoning), and linguistics, to natural, cognitive and social sciences. In case of branching time temporal logics, the underlying model is a tree of all possible paths of computations, paths of executions, paths of propagations, etc.

III. Definitions and Example

In this section, the main idea of our approach is presented. Theoretical description is supplemented with a simple example illustrating the proposed approach.

A tree $\mathbf{T}$ is a partially ordered set (poset) $\mathbf{T} = (T, R_<)$ such that for each $x \in T$ the set $\{y : (y, x) \in R_<\}$ is well-ordered by the binary relation $R_<$.

For a given tree $\mathbf{T} = (T, R_<)$, we can consider its subtree $\mathbf{T}^x$ rooted at $x \in T$, i.e., $\mathbf{T}^x = (T^x, R_<)$ such that $T^x = \{y \in T : (x, y) \in R_< \text{ or } y = x\}$.

Let $\mathbf{T} = (T, R_<)$ be a tree. A segment $]a, b[$, where $a, b \in T$, is a set $]a, b[= \{x \in T : (a, x) \in R_< \text{ and } (x, b) \in R_<\}$. An element b is called a successor to an element a. An element a is called a predecessor to an element b. If $]a, b[= \emptyset$, then an element b is called an immediate successor to an element a and an element a is called an immediate predecessor to an element b.

Let $\mathbf{T} = (T, R_<)$ be a tree and $x \in T$. The set of all immediate successors of x is denoted by $Succ(x)$. The set of all immediate predecessors of x is denoted by $Pred(x)$. A leaf of $\mathbf{T}$ is any element $x \in T$ such that $Succ(x) = \emptyset$. A chain of $\mathbf{T}$ is any linearly ordered subset of T. A branch of $\mathbf{T}$ is any maximal (with respect to a number of elements) chain of $\mathbf{T}$. A set of all leaves of $\mathbf{T}$ is denoted by $Leaves(\mathbf{T})$. A set of all chains of $\mathbf{T}$ is denoted by $Chains(\mathbf{T})$. A set of all branches of $\mathbf{T}$ is denoted by $Branches(\mathbf{T})$.

Further, to refer to trees of executions, each element of a tree will be called a state.

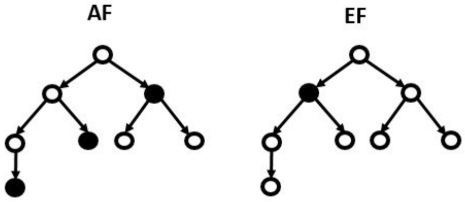

Fig. 2. AF and EF operators of branching time temporal logics

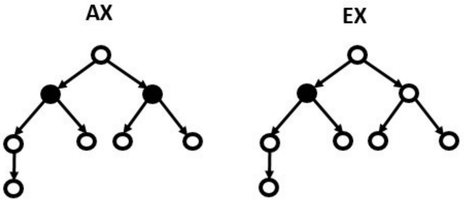

Fig. 3. AX and EX operators of branching time temporal logics

Let $\mathbf{T} = (T, R_<)$ be a tree and $S \subseteq T$ be a set of distinguished states in the tree $\mathbf{T}$. We can identify, in the set of states in the tree $\mathbf{T}$, the following regions:

- $PosAnt^G(S)$ - a positive region of G-anticipation of states from S.
- $BndAnt^G(S)$ - a boundary region of G-anticipation of states from S.

- $NegAnt^G(S)$ - a negative region of G-anticipation of states from S.
- $PosAnt^F(S)$ - a positive region of F-anticipation of states from S.
- $BndAnt^F(S)$ - a boundary region of F-anticipation of states from S.
- $NegAnt^F(S)$ - a negative region of F-anticipation of states from S.
- $PosAnt^X(S)$ - a positive region of X-anticipation of states from S.
- $BndAnt^X(S)$ - a boundary region of X-anticipation of states from S.
- $NegAnt^X(S)$ - a negative region of X-anticipation of states from S.

The division of regions given above corresponds to quantification over branches in the temporal logic of branching time.

Formal definitions of regions mentioned above are as follows. Let $\mathbf{T} = (T, R_<)$ be a tree and $S \subseteq T$. For each $x \in T$:

- $x \in PosAnt^G(S)$ if and only if
$$\underset{B \in Branches(\mathbf{T}^x)}{\forall} \underset{y \in B}{\forall} y \in S,$$
i.e., all branches started with x consist of only states belonging to S.
- $x \in BndAnt^G(S)$ if and only if
$$x \notin PosAnt^G(S) \text{ and } \underset{B \in Branches(\mathbf{T}^x)}{\exists} \underset{y \in B}{\forall} y \in S,$$
i.e., there is at least one branch started with x consisting of only states belonging to S, however not all branches started with x satisfy this condition.
- $x \in NegAnt^G(S)$ if and only if
$$x \notin PosAnt^G(S) \text{ and } x \notin BndAnt^G(S),$$
i.e., there is no branch started with x consisting of only states belonging to S.
- $x \in PosAnt^F(S)$ if and only if
$$\underset{B \in Branches(\mathbf{T}^x)}{\forall} \underset{y \in B}{\exists} y \in S,$$
i.e., all branches started with x consist of at least one state belonging to S.
- $x \in BndAnt^F(S)$ if and only if
$$x \notin PosAnt^F(S) \text{ and } \underset{B \in Branches(\mathbf{T}^x)}{\exists} \underset{y \in B}{\exists} y \in S,$$
i.e., there is at least one branch started with x consisting of at least one state belonging to S, however not all branches started with x satisfy this condition.
- $x \in NegAnt^F(S)$ if and only if
$$x \notin PosAnt^F(S) \text{ and } x \notin BndAnt^F(S),$$
i.e., there is no branch started with x consisting of at least one state belonging to S.
- $x \in PosAnt^X(S)$ if and only if
$$\underset{y \in Succ(x)}{\forall} y \in S,$$
i.e., all branches started with x are such that an immediate successor of x belongs to S.
- $x \in BndAnt^X(S)$ if and only if
$$x \notin PosAnt^X(S) \text{ and } \underset{y \in Succ(x)}{\exists} y \in S,$$
i.e., there is at least one branch started with x such that an immediate successor of x belongs to S, however not all branches started with x satisfy this condition.
- $x \in NegAnt^X(S)$ if and only if
$$x \notin PosAnt^X(S) \text{ and } x \notin BndAnt^X(S),$$
i.e., there is no branch started with x such that an immediate successor of x belongs to S.

One can see that:

- if $x \in Leaves(\mathbf{T})$ and $x \in S$, then $x \in PosAnt^G(S)$ and $x \in PosAnt^F(S)$,
- if $x \in Leaves(\mathbf{T})$ and $x \notin S$, then $x \in NegAnt^G(S)$ and $x \in NegAnt^F(S)$,
- if $x \in Leaves(\mathbf{T})$, then $x \in NegAnt^X(S)$,
- if $x \notin Leaves(\mathbf{T})$ and $x \in PosAnt^G(S)$, then $x \in PosAnt^F(S)$ and $x \in PosAnt^X(S)$.

Let us consider a tree $\mathbf{T}$ shown in Figure 4. For the set

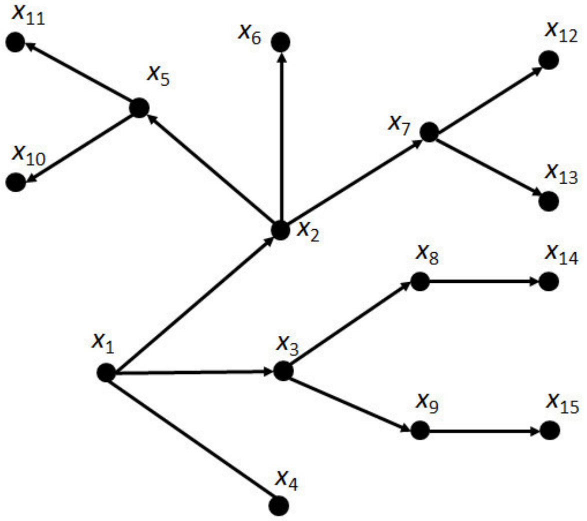

Fig. 4. A tree $\mathbf{T}$ modeling some executions.

$$S = \{x_3, x_7, x_8, x_{12}, x_{13}, x_{14}\}$$

of distinguished states of the tree $\mathbf{T}$, we obtain:

- $x_1 \in NegAnt^G(S)$ and $x_1 \in BndAnt^F(S)$ and $x_1 \in BndAnt^X(S)$,
- $x_2 \in NegAnt^G(S)$ and $x_2 \in BndAnt^F(S)$ and $x_2 \in BndAnt^X(S)$,
- $x_3 \in BndAnt^G(S)$ and $x_3 \in PosAnt^F(S)$ and $x_3 \in BndAnt^X(S)$,
- $x_4 \in NegAnt^G(S)$ and $x_4 \in NegAnt^F(S)$ and $x_4 \in NegAnt^X(S)$,

- $x_5 \in NegAnt^G(S)$ and $x_5 \in NegAnt^F(S)$ and $x_5 \in NegAnt^X(S)$,
- $x_6 \in NegAnt^G(S)$ and $x_6 \in NegAnt^F(S)$ and $x_6 \in NegAnt^X(S)$,
- $x_7 \in PosAnt^G(S)$ and $x_7 \in PosAnt^F(S)$ and $x_7 \in PosAnt^X(S)$,
- $x_8 \in PosAnt^G(S)$ and $x_8 \in PosAnt^F(S)$ and $x_8 \in PosAnt^X(S)$,
- $x_9 \in NegAnt^G(S)$ and $x_9 \in NegAnt^F(S)$ and $x_9 \in NegAnt^X(S)$,
- $x_{10} \in NegAnt^G(S)$ and $x_{10} \in NegAnt^F(S)$ and $x_{10} \in NegAnt^X(S)$,
- $x_{11} \in NegAnt^G(S)$ and $x_{11} \in NegAnt^F(S)$ and $x_{11} \in NegAnt^X(S)$,
- $x_{12} \in PosAnt^G(S)$ and $x_{12} \in PosAnt^F(S)$ and $x_{12} \in NegAnt^X(S)$,
- $x_{13} \in PosAnt^G(S)$ and $x_{13} \in PosAnt^F(S)$ and $x_{13} \in NegAnt^X(S)$,
- $x_{14} \in PosAnt^G(S)$ and $x_{14} \in PosAnt^F(S)$ and $x_{14} \in NegAnt^X(S)$,
- $x_{15} \in NegAnt^G(S)$ and $x_{15} \in NegAnt^F(S)$ and $x_{15} \in NegAnt^X(S)$.

We leave the reader with the proof of the assignments above.

IV. Conclusions

We have shown that rough sets can be used to express some properties (reachability of states) of systems whose underlying models of behaviour are trees of executions. The proposed approach is patterned upon the temporal logic of branching time, however a set theoretic approach causes that we do not need to consider system behaviours in terms of logical formulas. A challenging problem for further investigation is to consider anticipations of states in terms of the Variable Precision Rough Set Model [7] as well as fuzzy rough sets and rough fuzzy sets [8].

References

[1] Z. Pawlak, *Rough Sets. Theoretical Aspects of Reasoning about Data*. Dordrecht: Kluwer Academic Publishers, 1991.

[2] Y. Yao and T. Lin, “Generalization of rough sets using modal logics,” *Intelligent Automation and Soft Computing*, vol. 2, no. 2, pp. 103–120, 1996. doi: 10.1080/10798587.1996.10750660

[3] K. Pancerz and A. Schumann, “Rough set models of Physarum machines,” *International Journal of General Systems*, vol. 44, no. 3, pp. 314–325, 2015. doi: 10.1080/03081079.2014.997529

[4] A. Schumann and K. Pancerz, “Roughness in timed transition systems modeling propagation of plasmodium,” in *Rough Sets and Knowledge Technology*, ser. Lecture Notes in Artificial Intelligence, D. Ciucci, G. Wang, S. Mitra, and W.-Z. Wu, Eds. Springer International Publishing, 2015, vol. 9436, pp. 482–491.

[5] M. Ben-Ari, A. Pnueli, and Z. Manna, “The temporal logic of branching time,” *Acta Informatica*, vol. 20, no. 3, pp. 207–226, 1983. doi: 10.1007/BF01257083

[6] V. Goranko and A. Galton, “Temporal logic,” in *The Stanford Encyclopedia of Philosophy*, winter 2015 ed., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2015.

[7] W. Ziarko, “Variable precision rough set model,” *Journal of Computer and System Sciences*, vol. 46, no. 1, pp. 39–59, 1993. doi: 10.1016/0022-0000(93)90048-2

[8] D. Dubois and H. Prade, “Rough fuzzy sets and fuzzy rough sets,” *International Journal of General Systems*, vol. 17, no. 2-3, pp. 191–209, 1990. doi: 10.1080/03081079008935107