Tabular Interpretation of the Temporal Description Logic LTL_{ALC}

Valerii Reznichenko[1], Inna Chystiakova[1]

[1] *Institute of Software Systems of the National Academy of Sciences of Ukraine, Academician Glushkov avenue, 40, Kyiv, 03187, Ukraine*

Abstract

Description logic is widely used to describe and represent knowledge in the Semantic Web. It is a modern and powerful mechanism that provides the possibility of extracting knowledge from already existing ones. Thanks to this, conceptual modeling of subject areas has become one of the fields of application of description logic. Conceptual modeling is used to create databases and knowledge bases. A key issue in of the subject area modeling is the ability to monitor the dynamics of changes in the state of the subject area over time. It is necessary to describe not only the current actual state of the database/knowledgebase but also time-varying states. Temporal description logic is used to solve this problem. It has the same set of algorithmic problems that are presented in traditional description logic, but there are additional issues related to the description of knowledge in time. This refers to the form of time (continuous or discrete), time structure (moments of time, intervals, chains of intervals), time linearity (linear or branched), domain (present, past, future), the concept of "now", etc. An urgent task today is to develop approach of such interpretation of temporal description logic that may be supported by databases. The paper presents an algorithm of tabular interpretation of the temporal description logic LTL_{ALC}. Only the future tense is considered. The algorithm contains tabular notations of the following components of the LTL_{ALC}: concepts, concept constructors, roles, role constructors, TBox and ABox. Numerous examples are used to illustrate the application of the algorithm.

Keywords

Description logic, semantic web, temporal description logic, DL, LTL, ALC, LTL_{ALC}

1. Introduction

Description logics (DL) are a family of knowledge representation formalisms. They have different application areas, including semantic web ontologies. Particularly, OWL and OWL 2 of the W3C are grounded on the corresponding DL. Conceptual domain modeling problem that considers the use of withdrawal mechanisms became one of the DL application areas. Specially, it refers to the database and data knowledge issues. For example, extended ER-modeling language and extended UML class diagram were described with the help of DL. This allowed to use DL reasoners to check the software descriptions consistency (integrity) and perform influence of new implicitly specified knowledge.

Tracking the dynamics of software state changes over time is the key problem during the software conceptual modeling. That is the situation, when software conceptual informational model and the correspondence database track both the current state and all the prehistory. Such a temporal database should both allow an ability to fix a lot of states in the past (and probably in the future) and give an opportunity to operate these sets of states with their temporal integrity constraints descriptions. To satisfy these needs, a big number of publications on the temporal DL (TDL) theme have been appeared the last several years.

For example, such TDLs were proposed and discovered in modern literature: ITL Interval temporal logic) [1], LTL (Linear temporal logic) [2], STL (Signal temporal logic) [3], TTL

13th International Scientific and Practical Conference from Programming UkrPROGP’2022, October 11-12, 2022, Kyiv, Ukraine
EMAIL: reznichenko.valery47@gmail.com (A. 1); inna_islyamova@ukr.net (A. 2)
ORCID: 0000-0002-4451-8931 (A. 1); 0000-0001-7946-3611 (A. 2))

(Timestamp temporal logic) [4], PSL (Property specification language) [5], CTL* which generalizes LTL and CTL [6], HML (Hennessy–Milner logic) [7], MTL (Metric temporal logic) [8], MITL (Metric interval temporal logic) [8], TPTL (Timed propositional temporal logic) [9], TLTL (Truncated Linear Temporal Logic) [10].

This paper is dedicated to the table interpretation of the temporal description logic LTL_{ALC}. A survey of the LTL temporal description logic is also given. Publication includes the designation of the database that supports the interpretation of LTL and a set of temporal operations in such a database. The results are presented in a table form.

To define temporal interpretation of any DL there are three main questions:

1. What DL will be used?
2. What TDL will be used?
3. What interpretation of time is chosen?

According to the first question, there is a whole DL family characterized by the composition of elements (individuals, concepts, roles) and a set of constructors and axioms defined on these elements. About the second question, a choice of the TDL depends on the goals of the research. Third question can be answered according to the set of aspects:

- continuous and discrete time,
- linear and branching time,
- present, past, future,
- definition of "now",
- time structure (points of time, intervals, chains of intervals),
- ways to measure time.

As for this research, the following variants were chosen:

4. Linear discrete non-branching time, represented as an integer time axis with a linear ordering specified on it. (N<). Only the future time is considered.
5. LTL (Linear time temporal logic).
6. ALC (Attributive Language with Complements).

In this research the focus is made on the issues that related to informational models. The list of issues looks like below: syntax and semantics of the language, data model (data structure, operations, integrity constraints) and query language.

The aim of the present work is to show the algorithm for application the temporal logic LTL to the DL ALC. Section 2 is dedicated to some preliminaries that touch a brief survey of DL ALC and LTL basics. The rules of the DL temporal interpretation presented in Section 3. The temporal DL LTL_{ALC} interpretation (including concepts, roles, Tbox, Abox) can be found in Section 4. Section 5 is dedicated to the main conclusions and future research.

2. Preliminaries

2.1. Linear time temporal logic

LTL (Linear time temporal logic) is the logic that considers causal relationships in terms of time. The concepts of LTL are formed using the concept constructors of DL ALC enriched with the temporal constructors. This logic is used to describe the sequence of events and their relationship over time.

There are such temporal operators in the LTL logic:

$\Diamond$ ($\Diamond\varphi$) – existence in some moment of time in future

$\Box$ ($\Box\varphi$) – global existence in some moment of time (forever)

$\bigcirc$ ($\bigcirc\varphi$) – existence in the next moment of time in future

$\boldsymbol{\mathcal{U}}$ ($\varphi\ \boldsymbol{\mathcal{U}}\ \psi$) – until. φ exists until ψ in the current moment of time or in the future

$\boldsymbol{\mathcal{R}}$ ($\varphi\ \boldsymbol{\mathcal{R}}\ \psi$) – release. ψ has to be true until and including the point where φ first becomes true.

Every temporal operator can be written with the help of operators "Until" ($\boldsymbol{\mathcal{U}}$) and "Next" ($\bigcirc$). For example, existence and global existence operators can be written like below:

$\Diamond\varphi \equiv \top\ \boldsymbol{\mathcal{U}}\varphi$

$\Box\varphi \equiv \neg\Diamond\neg\varphi\ \equiv \bot\boldsymbol{\mathcal{R}}\varphi$

$\varphi \mathcal{R} \psi \equiv \neg (\neg\varphi \mathcal{U} \neg\psi)$

Equivalent transformation rules

There are the following rules for equivalent transformations of LTL formulas.

1. Duality (moving negation).

a) Operator ○ is self-dual:

$\neg\bigcirc\varphi \equiv \bigcirc\neg\varphi$

b) Operators □ and ◊ are dual:

$\neg\Box\varphi \equiv \Diamond\neg\varphi$

$\Box\varphi \equiv \neg\Diamond\neg\varphi$

$\neg\Diamond\varphi \equiv \Box\neg\varphi$

$\Diamond\varphi \equiv \neg\Box\neg\varphi$

c) Operators $\mathcal{U}$ and $\mathcal{R}$ are dual:

$\neg(\varphi \mathcal{U} \psi) \equiv (\neg\varphi \mathcal{R} \neg\psi)$

$\neg(\varphi \mathcal{R} \psi) \equiv (\neg\varphi \mathcal{U} \neg\psi)$

$(\varphi \mathcal{U} \psi) \equiv \neg(\neg\varphi \mathcal{R} \neg\psi)$

$(\varphi \mathcal{R} \psi) \equiv \neg(\neg\varphi \mathcal{U} \neg\psi)$

2. Distributivity rules.

a) Distributivity of ○ with respect to ∨, ∧ and $\mathcal{U}$:

$\bigcirc(\varphi \vee \psi) \equiv \bigcirc \varphi \vee \bigcirc\psi$

$\bigcirc(\varphi \wedge \psi) \equiv \bigcirc\varphi \wedge \bigcirc\psi$

$\bigcirc(\varphi \mathcal{U} \psi) \equiv \bigcirc\varphi \mathcal{U} \bigcirc\psi$

b) Distributivity of ◊ with respect to ∨ and □ with respect to ∧:

$\Diamond(\varphi \vee \psi) \equiv \Diamond\varphi \vee \Diamond\psi$

$\Box(\varphi \wedge \psi) \equiv \Box \varphi \wedge \Box\psi$

3. Special properties.

$\Diamond\varphi \equiv \Diamond\Diamond\varphi$

$\Box\varphi \equiv \Box\Box\varphi$

2.2. Description logic ALC

Traditionally, DL is aimed at presenting terminological knowledge about software. Individuals, concepts, and roles are the main objects of such presentation. There are the following three types of sets in DL:

- $N_I = \{a_1, a_2,..., a_m\}$ – set of individuals names;
- $N_C = \{A_1, A_2,..., A_n\}$ – set of atomic concepts names;
- $N_R = \{R_1, R_2,..., R_k\}$ – set of atomic roles names.

In relation to concepts and roles, constructors are defined that allow to build new concepts / roles from already defined individuals, concepts, and roles. Description logic in a DL family is specified by a set of valid concept and role constructors.

ALC (Attributive Language with Complement) is the simplest option in the DL family. It contains the minimum set of constructors, which is mandatory for all other DL in the family.

ALC syntax. Syntax ALC (a set of valid concepts is given by the following inductive definition.

- ⊤ - Thing (universal concept);
- A – atomic concept (every atomic concept is concept);
- C – arbitrary concept, ¬C – negotiation of C is also a concept;
- C ⊓ D – concept intersection is also a concept, if C and D are concepts;
- ∃R.C – existential restriction is also a concept, if C is a concept and R – atomic role.

Usually, ALC also include the following additional concepts and constructors, which are defines with the help of concepts and constructors defined above:

- ⊥ ≡ ¬⊤ – Nothing (empty concept);
- C ⊔ D ≡ ¬ (¬C ⊓ ¬D) – concept union is also a concept, if C and D are concepts;
- ∀R.C ≡ ¬∃R.¬C – universal restriction is also a concept, if C is a concept and R – atomic role.

Usually, to declare ALC syntax an extended version is used. It defines as follows:

$\top \mid \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C$

ALC semantics. ALC semantics (just like any other DL) is based on the interpretation notion. It looks like this: $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, where Δ is a non-empty set, called the domain of interpretation (domain) and a $\cdot^{\mathcal{I}}$ is an interpretation function. The interpretation function assigns:

- to each individual name $a \in N_I$ the element $a^{\mathcal{I}} \in \Delta$;
- to each atomic concept name $A \in N_C$ the arbitrary subset $A^{\mathcal{I}} \subseteq \Delta$;
- to each atomic role name $R \in N_R$ the arbitrary subset $R^{\mathcal{I}} \subseteq \Delta \times \Delta$.

The interpretation function $\cdot^{\mathcal{I}}$ extends to ALC compound concepts as follows:

- $\top^{\mathcal{I}} = \Delta$, $\bot^{\mathcal{I}} = \emptyset$;
- $(\neg C)^{\mathcal{I}} = \Delta \setminus C^{\mathcal{I}}$;
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$;
- $(\exists R.C)^{\mathcal{I}} = \{e \in \Delta \mid$ exist $d \in \Delta$ such that $(e, d) \in R^{\mathcal{I}}$ и $d \in C^{\mathcal{I}}\}$;
- $(\forall R.C)^{\mathcal{I}} = \{ e \in \Delta \mid$ for all $d \in \Delta$ such, that $(e, d) \in R^{\mathcal{I}}$, occurs $d \in C^{\mathcal{I}}\}$.

Terminologies and statements (TBox and ABox). Concepts describe facts that exist in the subject area, and their constructors allow to perform operations on them. They work as the basis to describe subject area knowledge. There are two types of knowledge: *intensional knowledge* (general knowledge about concepts) and *extensional knowledge* (knowledge about individual objects). Intensional knowledge is more stable and permanent. Extensional knowledge is more exposed modifications. According to this division, knowledge that is recorded using DL can be subdivided into:

- a set of terminological axioms (TBox);
- a set of statements (facts) about individuals (ABox).

Both TBox and ABox form a knowledge base.

TBox. This is a finite set of terminological axioms (terminologies) of the form:

- $C \sqsubseteq D$ – concept inclusion axioms;
- $C \equiv D$ – axioms of concept identity.

Obviously, these axioms are represented in the following way through the axioms of inclusion:

$C \sqsubseteq D \land D \sqsubseteq C$

ABox. This is a finite set of assertion axioms (statements) of the form:

- $C(a)$ – individual a is an instance of concept C;
- $R(a, b)$ – individuals a and b are connected with the role R.

Model and satisfiability. Concept C is *satisfiable* if interpretation $\mathcal{I}$ exist, which is $C^{\mathcal{I}} \neq \emptyset$. Such interpretation is called *concept model*.

Interpretation $\mathcal{I}$ is the model of TBox T, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for each $C \sqsubseteq D$ from T.

Interpretation $\mathcal{I}$ is the model of ABox A, if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for each $C(a) \in A$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ for each $R(a, b) \in A$.

Concept C is satisfiable with respect to the TBox T if a common model for C and T exists.

3. DL temporal interpretation

There are many ways to use standard temporal logic operators in DL. The following way is chosen in this paper. Temporal operators are used as additional concept constructors that allows to describe temporal behavior of individuals which belong to concepts. In addition, temporal operators can be applied to the roles, TBoxes, and ABoxes. All these options are united by a single notion of *temporal DL interpretation*.

DL temporal interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ consists of non-empty domain Δ and $\cdot^{\mathcal{I}}$ – interpretation function, which maps:

- each individual name $a \in N_I$ into element $a^{\mathcal{I}} \in \Delta$;
- each concept name $A \in N_C$ into subset $A^{\mathcal{I}} \subseteq \mathbb{N} \times \Delta$;
- each role name $R \in N_R$ into subset $R^{\mathcal{I}} \subseteq \mathbb{N} \times \Delta \times \Delta$.

Here time is presented with natural numbers $\mathbb{N}$. On these numbers the order is given $< (\mathbb{N}, <)$. For example, $(n, d) \in A^{\mathcal{I}}$ means that individual d in the interpretation $\mathcal{I}$ is an element of A at time point n.

Same for roles. Thus, concepts/roles (composition of concepts/roles) change over time, and this is the essence of their temporal interpretation. In turn, in the DL temporal interpretation defined above, the names of individuals do NOT change in time. That is, they are interpreted in the same way at all points in time. In this sense they are said to be *rigid*.

Note, the DL temporal interpretation defined above is a special case of a first-order temporal structure without function symbols and an equality predicate and provided there are no predicates higher than binary (two-place predicate).

There is an equivalent representation of temporal interpretation $\mathcal{J}$ as an infinite sequence $\mathcal{J}(0)$, $\mathcal{J}(1)$,... of non-temporal interpretations, that are defined on the same domain Δ and with a fixed interpretation of individual names.

Constant domain assumption is a restriction on domain, which doesn`t allow domain to change in time. It means that a set of the domain individuals can`t change in time. There are alternative variants of the temporal interpretation, that includes domains expanding $\Delta^{\mathcal{J}(0)} \subseteq \Delta^{\mathcal{J}(1)} \subseteq \ldots$, domains decreasing $\Delta^{\mathcal{J}(0)} \supseteq \Delta^{\mathcal{J}(1)} \supseteq \ldots$ and just varying domains.

Concept constructors in the DL temporal interpretation. Concept constructors have traditional interpretation in the DL temporal interpretation. That is, they are interpreted in a standard way for DL at every moment of time, that is, regardless of other moments of time. Here are examples with the following preliminaries:

- X-axis means the time;
- Y-axis means concept individuals;
- table is the temporal concept interpretation;
- red table cell[2] means that individual belongs to the concept in the corresponding moment of time.

Temporal concept "Thing" ⊤. Table representation of the temporal concept Thing is shown on figure 1.

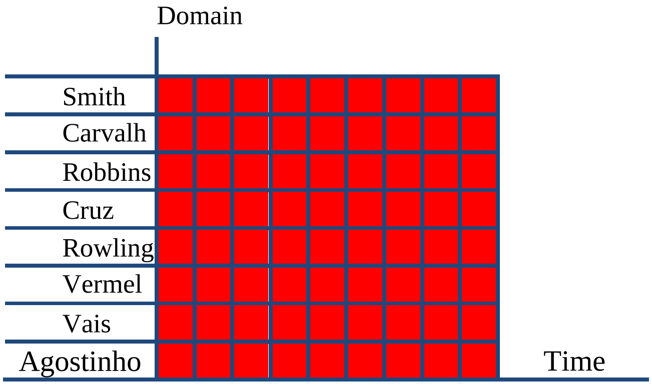


Figure 1: Temporal concept "Thing" ⊤

Temporal concept "Nothing" ⊥. Table representation of the temporal concept Nothing is shown on Figure 2.

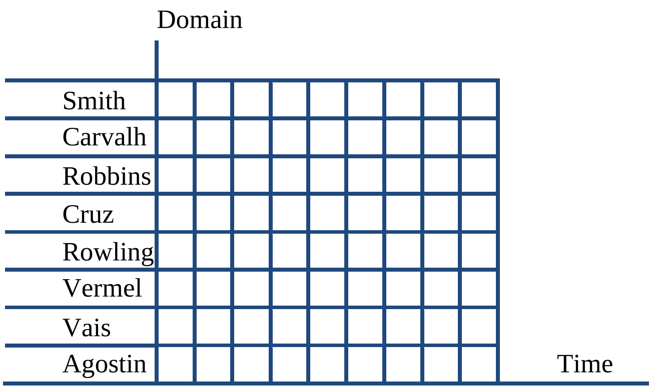


Figure 2: Temporal concept "Nothing" ⊥

[2] In the paper version of the article, red cells are represented in black.

Temporal concept “Department staff” (Concept C). Table representation of the temporal concept “Department staff” (Concept C) is shown on Figure 3.

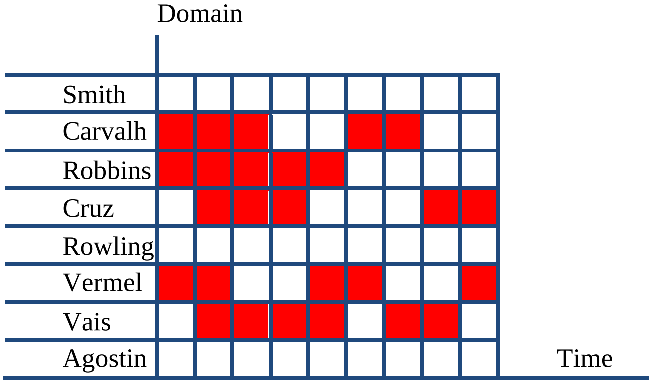


Figure 3*:* Temporal concept “Department staff” (Concept C)

Temporal concept “Articles publication” (Concept D). Table representation of the temporal concept “Articles publication” (Concept D) is shown on Figure 4.

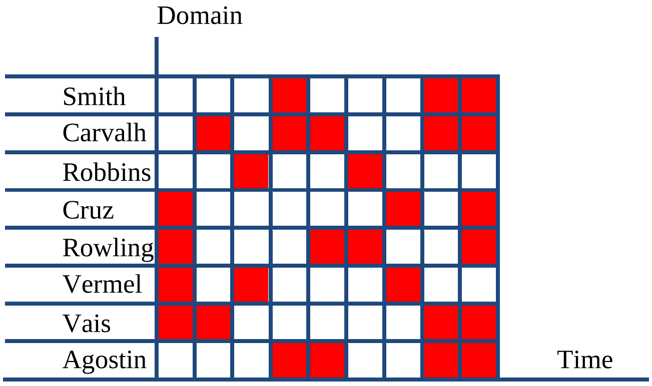


Figure 4: Temporal concept “Articles publication” (Concept D)

Temporal concept “Articles publication by department staff” (Concept C ⊓ D). Table representation of the temporal concept “Articles publication by department staff” (Concept C ⊓ D) is shown on Figure 5.

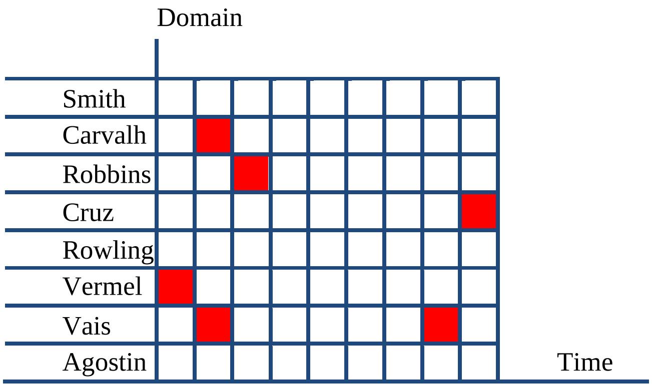
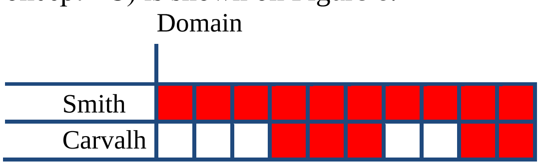


Figure 5: “Articles publication by department staff” (Concept C ⊓ D)

Temporal concept NOT “Department staff” (¬C). Table representation of the temporal concept “Department staff” (Concept ¬C) is shown on Figure 6.



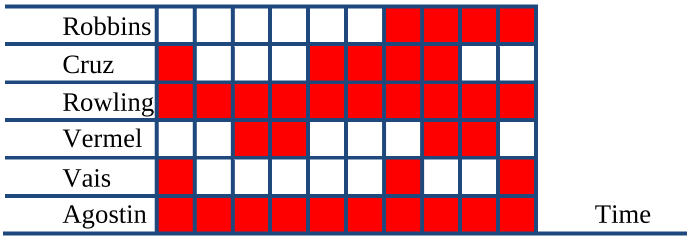


Figure 6: Temporal concept NOT "Department staff" (¬C)

Note. In LTL the time axis is infinite. In order to use finite axis, we make the following assumption. On the time axis there exists a point of time that has the following property: for any row of the table

- if at this moment of time the cell is red, then all the remaining cells to the right are red up to the infinity;
- if at this moment of time the cell is white, then all the remaining cells to the right are white up to the infinity.

In all previous and subsequent examples, it is assumed that such a point on the time axis has been reached.

4. Temporal interpretation of DL LTL_{ALC}

4.1. Temporal interpretation of DL LTL_{ALC} concepts

In previous section we described the meaning of the temporal DL interpretation and showed how the concept constructors are used. Concept constructors are operations over the columns of the tables (in examples, given in this paper). In turn, to use LTL it is requires include to the DL temporal operators (constructors). They are operations over the rows of the table.

The syntax for including temporal statements in DL is very simple. If C and D are concepts (atomic or arbitrary) then ○C, ◊C, ☐C, C $\mathcal{U}$ D и C $\mathcal{R}$ D are also concepts. These operators have the following semantics:

$(\bigcirc C)^{\mathcal{I}} = \{(n, d) \mid (n+1, d) \in C^{\mathcal{I}}\}$ – Next

$(\Diamond C)^{\mathcal{I}} = \{(n, d) \mid \exists m \geq n \ (m, d) \in C^{\mathcal{I}}\}$ – Exist

$(\Box C)^{\mathcal{I}} = \{(n, d) \mid \forall m \geq n \ (m, d) \in C^{\mathcal{I}}\}$ – Always

$(C \ \mathcal{U} \ D)^{\mathcal{I}} = \{(n, d) \mid \exists m \geq n \ ((m, d) \in D^{\mathcal{I}} \wedge (k, d) \in C^{\mathcal{I}} \text{ for } n \leq k < m)\}$ – Until

$(C \ \mathcal{R} \ D)^{\mathcal{I}} = (\neg(\neg C \ \mathcal{U} \ \neg D))^{\mathcal{I}}$ – Release

Let us explain the essence of these operators with examples.

○C means that "in the next moment of time C will take place". The following Figure 7 shows five examples where a pair of rows represent the separate example. The first row is the meaning of C, and the second one is the meaning of ○C. As can be seen from the examples, the operator ○ means shifting the red cells C to the left.

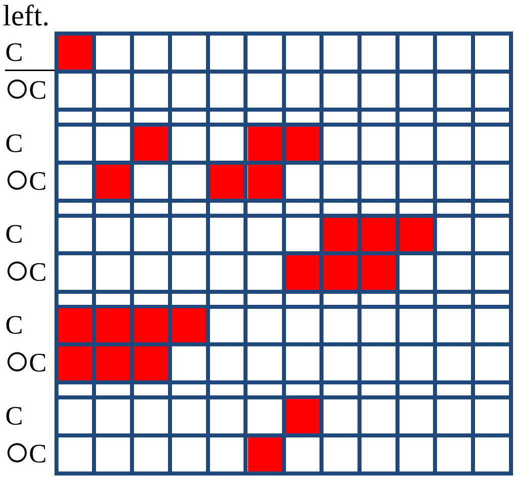


Figure 7: Examples of the operator ○C usage

The following two figures show the temporal meanings of the concepts C (Figure 8) and ◊C (figure 9). The action of the operator ◊ can be described by the following procedure:

- for each individual (row) the first red cell from the right side is found and all subsequent cells, up to the first, are converted to red;
- if there is no such cell, then the entire row remains white.

$C^{\mathcal{I}}$

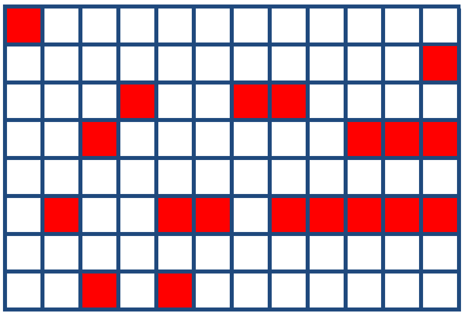


Figure 8: Temporal interpretation of C

$(\Diamond C)^{\mathcal{I}} = \{(n, d) \mid \exists m \geq n\ (m, d) \in C^{\mathcal{I}}\}$

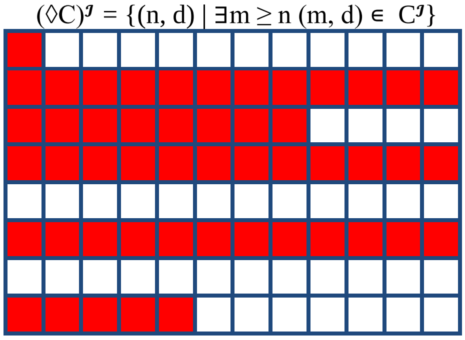


Figure 9: Temporal interpretation of ◊C

The following two figures show the temporal meanings of the concepts C (Figure 10) and ☐C (Figure 11). The action of the operator ☐ can be described by the following procedure:

- for each individual (row) the first white cell from the right side is found and all subsequent cells, up to the first, are converted to white;
- if there is no such cell, then the entire row remains red.

$C^{\mathcal{I}}$

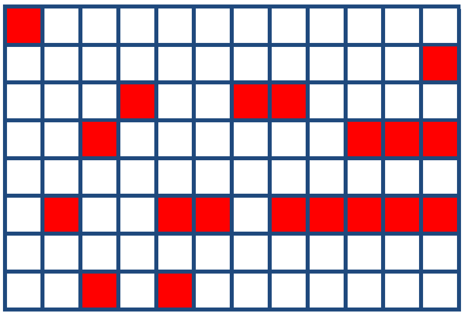


Figure 10: Temporal interpretation of C

$(\Box C)^{\mathcal{I}} = \{(n, d) \mid \forall m \geq n\ (m, d) \in C^{\mathcal{I}}\}$

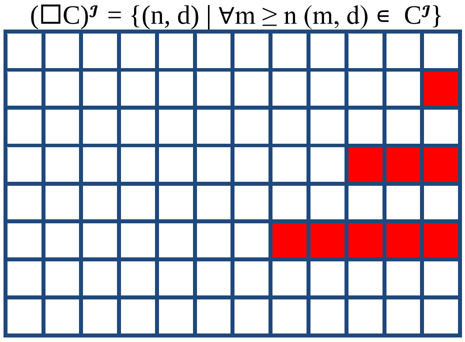


Figure 11: Temporal interpretation of ☐C

As can be seen from these examples, the operators ☐ and ◊ are dual, that is:

$\neg\Box C \equiv \Diamond\neg C \quad \Box C \equiv \neg\Diamond\neg C$

$\neg\Diamond C \equiv \Box\neg C \quad \Diamond C \equiv \neg\Box\neg C$

The following Figure 12 shows an example of the $\mathcal{U}$ operator. The result of the operator's action is those red cells C that are obtained by executing the following procedure. For each continuous range of red cells of C:

- within this range the rightmost red cell of D is found. The range of C is cut from the right side by the rightmost red cell of D;
- if there is no red cell of D within the range of red cells of C, then this range is not included into the result of $\mathcal{U}$.

In other words, sometime in the future D will be true, and until then C will be true.

$(C\ \mathcal{U}\ D)^{\mathcal{I}} = \{(n, d) \mid \exists m \geq n\ ((m, d) \in D^{\mathcal{I}} \wedge (k, d) \in C^{\mathcal{I}}\ \text{for } n \leq k < m)\}$

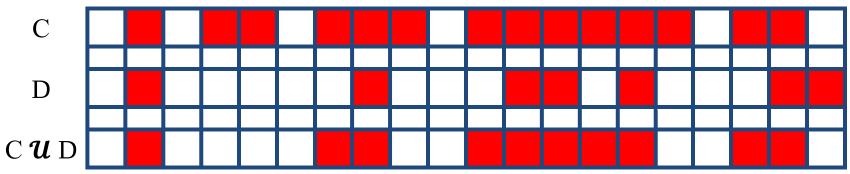


Figure 12: Temporal interpretation of the concept C $\mathcal{U}$ D

The following Figure 13 shows an example of the $\mathcal{R}$ operator. The result of the operator's action includes the following cells:

- all the (range of) red cells of C and
- all those (the range of) red cells of D which are adjacent to the left of the red ranges of C or overlap them.

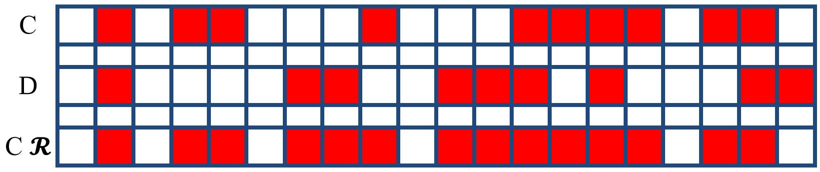


Figure 13: Temporal interpretation of the concept C $\mathcal{R}$ D

4.1.1. Temporal constructor ∃R.C (∀R.C)

Role R is a binary relation. The first members (from the left side) of the role will be called "predecessors" (R-predecessors) in this paper. The second members (from the right side) of the role will be called "followers" (R-followers). In the general case, a role is a binary relation of the m:n type. That means many R-successors can correspond to each R-predecessor and vice versa. Concept constructor ∃R.C build a concept from the role (R) and concept (C) in the following way. The result of the operation is such a set of R-predecessors for which *at least* one R-follower belongs to the concept C. In turn, the constructor ∀R.C defines such a set of R-predecessors, in which *all* R-successors belong to the concept C.

These constructors are dual. That means:

$\forall R.C \equiv \neg\exists R.\neg C, \quad \exists R.C \equiv \neg\forall R.\neg C$

Figure 14 illustrates the essence of these constructors. The following image illustrates the meaning of the temporal role R. X-axis means role followers; Y-axis means role predecessors. Red cells of the table mean which R-predecessor and which R-follower are in the R relation.

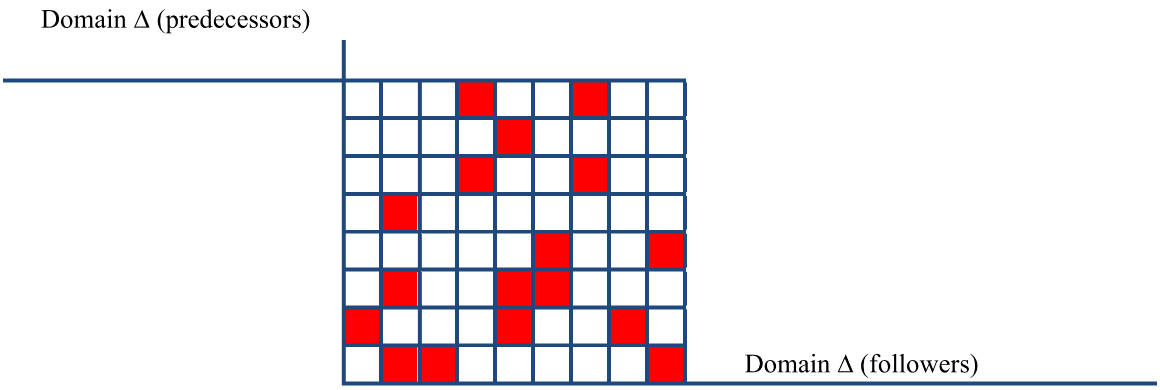
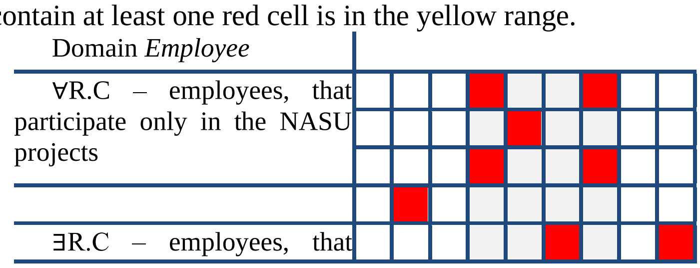


Figure 14:. Temporal role R

Figure 15 illustrates the meaning of the ∀R.C and ∃R.C. The subject area is participation of department staff in projects. X-axis shows projects, Y-axis shows department staff. Grey cells correspond to the NASU projects (concept C). Result of concept ∀R.C contains such rows, which contain all the red cells inside the yellow range. Result of concept ∃R.C contains such rows, which contain at least one red cell is in the yellow range.



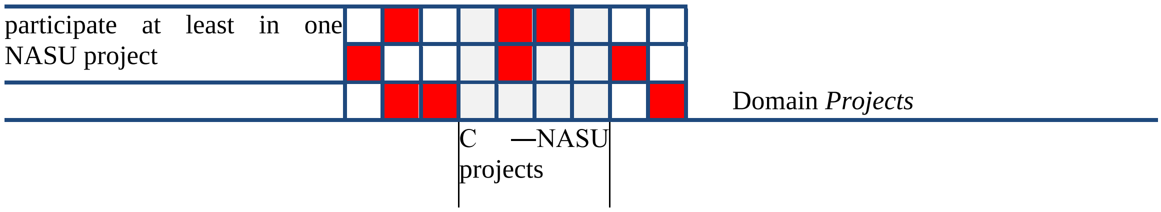


Figure 15: Temporal concepts ∀R.C and ∃R.C

The temporal interpretation of the constructor ∃R.C is as follows:

$(\exists R.C)^{\mathcal{I}} = \{(n, d) \in \Delta \mid \exists(n, c) \in C^{\mathcal{I}} \land (n, d, c) \in R^{\mathcal{I}}\}$

4.2. Temporal interpretation of DL LTL_{ALC} roles

In terms of temporal interpretation and application of temporal operators role and concept have similar meaning. Temporal interpretation for concepts defines with the help of individuals. Temporal interpretation for roles defines with the help of pair of individuals. Formal interpretation of the role temporal interpretation:

$R^{\mathcal{I}} \subseteq \mathbb{N} \times \Delta \times \Delta$

Here is an example. The subject area is the same (participation of department staff in projects). Role is participation of employees in projects. Domain of the role is the following set of pairs:

Smith, project1	Smith, project2
Cruz, project1	Cruz, project2
Robbins, project1	Robbins, project2

The temporal interpretation of the role looks like this (fig.16):

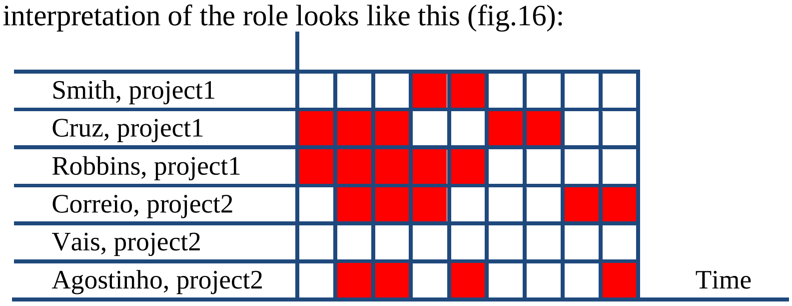


Figure 16: Temporal interpretation of role "Participation of the employee in projects" (domain Employee x Project)

Role constructors are absent in the DL ALC. Traditional temporal operators can be applied to the roles in TLT_{ALC}. Semantics is the following:

$(\bigcirc R)^{\mathcal{I}} = \{(n, d, d') \mid (n+1, d, d') \in R^{\mathcal{I}}\}$

$(\Diamond R)^{\mathcal{I}} = \{(n, d, d') \mid \exists m \geq n\ (m, d, d') \in R^{\mathcal{I}}\}$

$(\Box R)^{\mathcal{I}} = \{(n, d) \mid \forall m \geq n\ (m, d, d') \in R^{\mathcal{I}}\}$

$(R\ \mathcal{U}\ S)^{\mathcal{I}} = \{(n, d, d') \mid \exists m \geq n\ ((m, d, d') \in S^{\mathcal{I}} \land (k, d, d') \in R^{\mathcal{I}} \text{ for } n \leq k < m)\}$

Figure 17–20 shows how to use these operators over the roles.

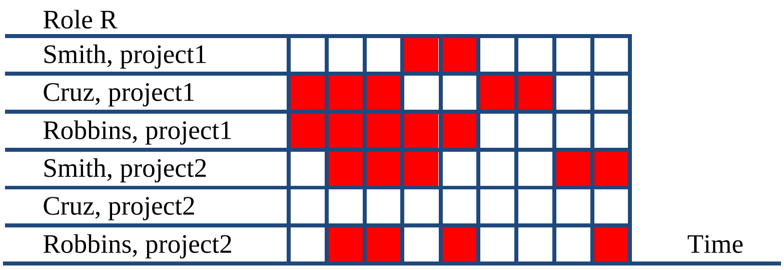


Figure 17: Temporal role R

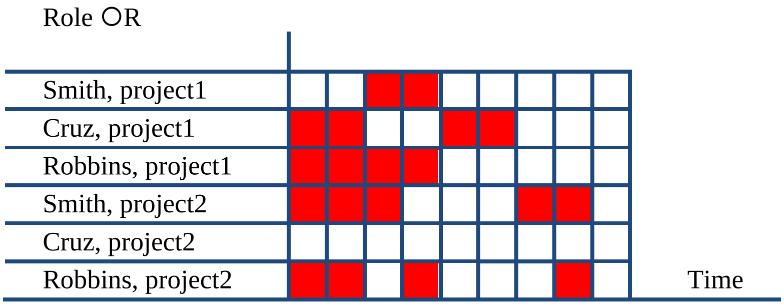


Figure 18: Temporal role ○R

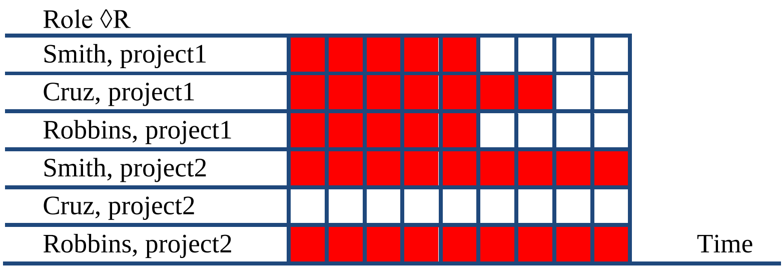


Figure 19: Temporal role ◊R

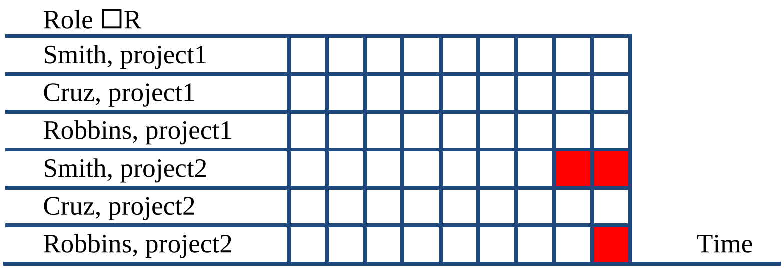


Figure 20: Temporal role ☐R

4.3. Temporal interpretation of DL LTL_{ALC} TBox

Development over time of concepts and roles in TDL was shown in previous sections. But there are such temporal statements that can't be described with the help of temporal concepts and roles. They require terminological axiom (concept inclusion) usage. For example, *fixed* concept means, that concept doesn't change in time. It keeps the set of its individuals regardless of time. In other words, the fixed concept will be interpreted as a row with all the red cells or a row with all the white cells. The fixed concept can be represented with the following concept inclusion:

$C \sqsubseteq \Box C$, $\neg C \sqsubseteq \Box \neg C$

TTBox definition requires to define TTBox formulas, which involve the use of temporal operators. There are two situations:

- temporal operators are applied only to the terminological axiom in general, but aren`t applied to their concepts. This situation is called temporal ALC TBox (ALC TBox).
- temporal operators are applied both to the terminological axiom and to their concepts (except roles). This situation is called temporal LTL_{ALC} TBox ($LTL_{\mathcal{ALC}}$ TBox).

4.3.1. Temporal ALC TBox

Syntactic rules for constructing formulas ALC TBox:

$C \sqsubseteq D$, $\neg\varphi$, $\varphi \wedge \psi$, $\bigcirc\varphi$, $\varphi \mathcal{U} \psi$

where:

C and D – atomic concepts with temporal interpretation

φ and ψ – temporal TBox

Semantics of these formulas defines as follows:

$\mathcal{I}, n \vDash C \sqsubseteq D \Leftrightarrow \{d \mid (n, d) \in C^{\mathcal{I}}\} \subseteq \{d \mid (n,d) \in D^{\mathcal{I}}\}$

$\mathcal{I}, n \vDash \neg \varphi \quad \Leftrightarrow \mathcal{I}, n \nvDash \varphi$

$\mathcal{I}, n \vDash \varphi \wedge \psi \Leftrightarrow \mathcal{I}, n \vDash \varphi$ and $\mathcal{I}, n \vDash \psi$

$\mathcal{I}, n \vDash \bigcirc\varphi \Leftrightarrow \mathcal{I}, n+1 \vDash \varphi$

$\mathcal{I}, n \vDash \varphi \mathcal{U} \psi \Leftrightarrow \exists m \geq n \{\mathcal{I}, m \vDash \psi$ and $\forall n \leq k < m\ \mathcal{I}, k \vDash \varphi\}$

The record $\mathcal{I}, n \vDash \varphi$ means: φ is true in the n moment of time in interpretation $\mathcal{I}$. Definition if the terminological axiom is true or false depends on each moment of time (locally). It is not a general definition (globally). Here are graphical examples of temporal ALC TBox and the result of temporal axioms. Figures 21–23 demonstrate temporal concepts “Institute staff” (C), “Department staff” (D) and “Project staff” (E).

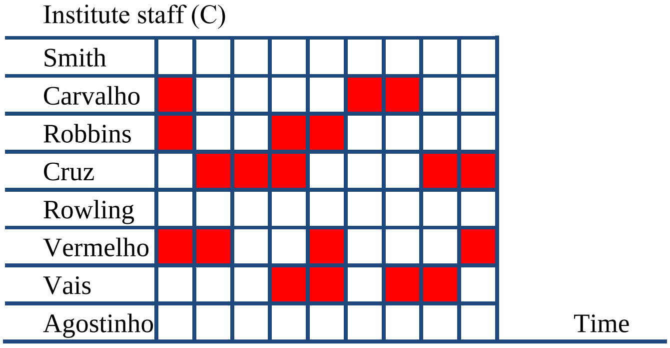


Figure 21: Temporal concept “Institute staff” (C)

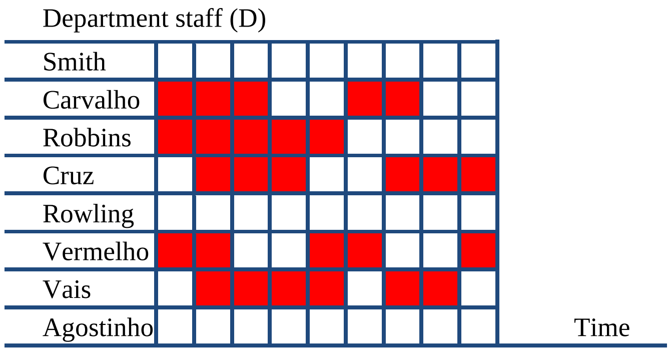


Figure 22: Temporal concept “Department staff” (D)

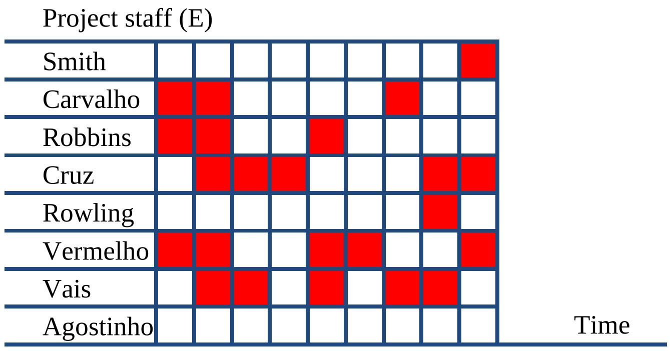


Figure 23: Temporal concept “Project staff” (E)

Here are examples of the result of the different terminological axioms. Result contains its truth values at the corresponding time. In these examples result is shown as a concept, that contains single individual, that belongs to this concept in the corresponding moment of time, when the terminological axiom is true.

1. Axiom C $\sqsubseteq$ D is true for all the moments of time. It means that all the department staff employees are institute employees.

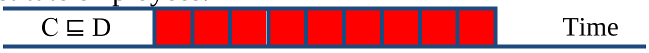


Figure 24: Temporal axiom C $\sqsubseteq$ D

2. Axiom D $\sqsubseteq$ C is true for 1,5,8,9 moments of time. So, department will include all the institute staff on that moments of time.

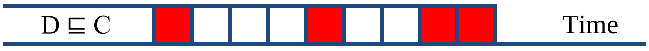


Figure 25: Temporal axiom D $\sqsubseteq$ C

3. Axiom E $\sqsubseteq$ D is true for 1–7 moments of time. So, only institute staff will work on the project on that moment of time.

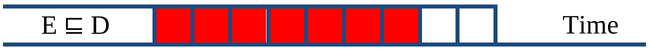


Figure 26: Temporal axiom E $\sqsubseteq$ D

4. Axiom D $\sqsubseteq$ E is true for 1, 8, 9 moments of time. In the moment 1 axiom E $\sqsubseteq$ D is true. So, at this moment only institute employees will work on the project and no one else. As for 8 and 9 moments of time, all the institute staff and someone else will work on the project.

Figure 27: Temporal axiom D $\sqsubseteq$ E

5. Axiom C $\sqsubseteq$ E is true for 1, 2, 3, 5, 7, 8, 9 moments of time. So, only department staff will work on the project at these moments of time.

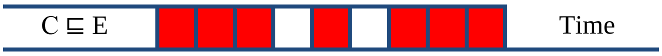


Figure 28: Temporal axiom C $\sqsubseteq$ E

6. Axiom E $\sqsubseteq$ C is true for 1, 4, 5, 7 moments of time. In the moments of time 1, 5, 7 C $\sqsubseteq$ E is true. So, at these moments of time only department staff will work on the project and no one else. At the 4th moment of time only department employees, but not all of them will work on the project.

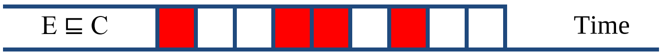


Figure 29: Temporal axiom E $\sqsubseteq$ C

7. Axiom E $\sqsubseteq$ D $\wedge$ D $\sqsubseteq$ E is true for 1 moment of time. In this moment of time all the institute, but no one else will work on the project.

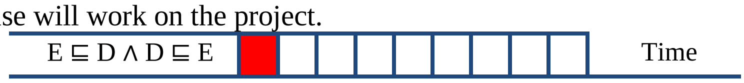


Figure 30: Temporal axiom E $\sqsubseteq$ D $\wedge$ D $\sqsubseteq$ E

8. Axiom C $\sqsubseteq$ E $\wedge$ E $\sqsubseteq$ D is true for the 1–7 moments of time. So, at these moments of time only employees that are both department employees and institute employees will work on the project.

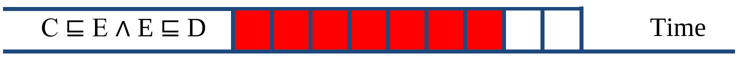


Figure 31: Temporal axiom C $\sqsubseteq$ E $\wedge$ E $\sqsubseteq$ D

9. Axiom ◊(E ⊑ C) is true for the 1–7 moments of time (as terminological axiom E ⊑ C is true in the 1, 4, 5, 7 moments of time).

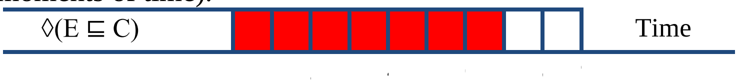


Figure 32: Temporal axiom ◊(E ⊑ C)

10. Axiom □(C ⊑ E) is true for the 7, 8, 9 moments of time (because C ⊑ E is true for the 1, 2, 3, 5, 7, 8, 9 moments of time).

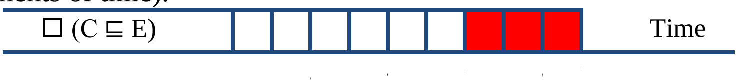


Figure 33: Temporal axiom □(C ⊑ E)

11. Axiom ◊□(C ⊑ E) is true for the 1–9 moments of time.

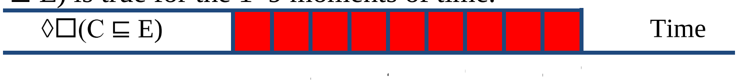


Figure 34: Temporal axiom ◊□(C ⊑ E)

12. Axiom ○□(C ⊑ E) is true for the 6, 7, 8 moments of time.

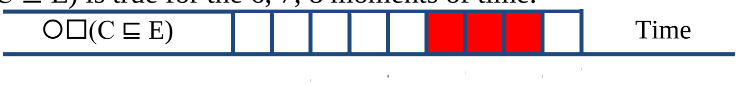


Figure 35: Temporal axiom ○□(C ⊑ E)

4.3.2. Temporal LTL_{ALC} TBox

Expressive possibilities obtained when the temporal operators are applied to concepts are incomparable (mismatched) in comparison when temporal operators are applied to the terminological axioms. Here is an example in LTL_{ALC} TBox (next two examples are taken from [2]):

independent_country ⊑ □independent_country

It say that the extension of the concept "independent_country" cannot decrease, but this cannot be expressed by a temporal ALC TBox. On the other hand, the assertion that eventually all European countries will be EU members forever can be expressed by the temporal ALC TBox:

◊□(european_country ⊑ member_EU)

But it is not expressible in the non-temporal LTL_{ALC} TBox.

If temporal operators are applied only to the concepts and terminological axioms. interaction between TL and DL components in TDL is severely limited. In its turn it can be expressed quite developed types of relationships in the temporal LTL_{ALC} TBox. For example:

$\bigcirc\neg(\top \sqsubseteq A) \Leftrightarrow \neg(\top \sqsubseteq \bigcirc A)$

4.4. Temporal interpretation of DL LTL_{ALC} ABox

To define temporal ABox two options should be considered:

- ABox that has temporal interpretation. It is called non-temporal ABox.
- ABox that has temporal interpretation and temporal operators. It is called temporal ABox.

Usual ABox contains the following types of predicates: C(a) and R(A, b). There are no constructors for them as for concepts. As regards to their temporal interpretation, its semantics defines as follows:

$\mathcal{J}, n \models C(a) \Leftrightarrow (n, a^{\mathcal{J}}) \in C^{\mathcal{J}}$

$\mathcal{J}, n \models R(a, b) \Leftrightarrow (n, a^{\mathcal{J}}, b^{\mathcal{J}}) \in R^{\mathcal{J}}$

Note that the truth value of ABox is determined for each point of time (locally), and not in general (globally).

The temporal ABox assumes the use of the same temporal operators as TBox:

C(a), R(a,b), ¬φ, φ ∧ ψ, ○φ, φ𝓤ψ where:

- C(A) and R(a, b) – atomic statements ABox with temporal interpretation
- φ and ψ – temporal ABox

Semantics of these formulas defines as follows:

$\mathcal{I}, n \vDash \neg\varphi \Leftrightarrow \mathcal{I}, n \nvDash \varphi$

$\mathcal{I}, n \vDash \varphi \wedge \psi \Leftrightarrow \mathcal{I}, n \vDash \varphi$ and $\mathcal{I}, n \vDash \psi$

$\mathcal{I}, n \vDash \bigcirc\varphi \Leftrightarrow \mathcal{I}, n+1 \vDash \varphi$

$\mathcal{I}, n \vDash \varphi\mathcal{U}\psi \Leftrightarrow \exists m \geq n \{\mathcal{I}, m \vDash \psi$ and $\forall n \leq k < m\ \mathcal{I}, k \vDash \varphi\}$

Here are several examples:

- C(Smith) – becomes true in the moments 1, 7, 8.
- C(Cruz) – becomes true in the moments 1, 4, 5.
- E(Smith) – becomes true in the moments 1, 2, 7.
- E(Robbins) – becomes true in the moments 2, 3, 4, 8, 9.
- C(Smith) ∧ C(Cruz) – becomes true in the moment 1. This is the moment when these two employees work at the same time.
- C(Smith) ∧ E(Cruz) – becomes true in the moment 8. This is the moment when Smith was a department employee and Cruz worked on the project.
- C(Smith) ∧ E(Smith) – becomes true in the moment 1 and 7. These are moments, when Smith was a department employee and worked on the project.
- C(Smith) ∧ ¬E(Smith) – becomes true in the moment 8. These are moments when Smith was a department employee and didn't work on the project.
- ◊C(Smith) – becomes true in the moment 1–5.
- E(Robbins) – becomes true in the moments 8, 9.

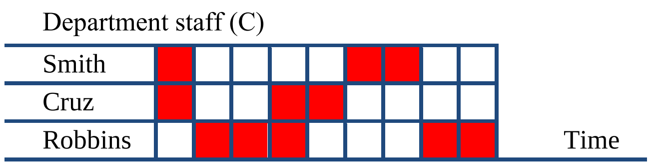


Figure 36: Temporal concept "Department staff" (C)

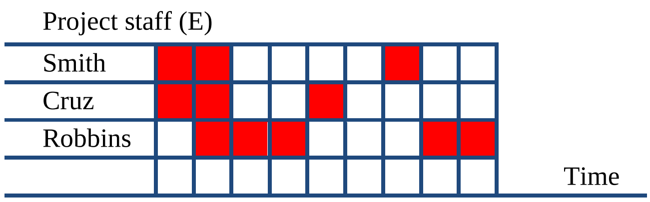


Figure 37: temporal concept "Project staff" (E)

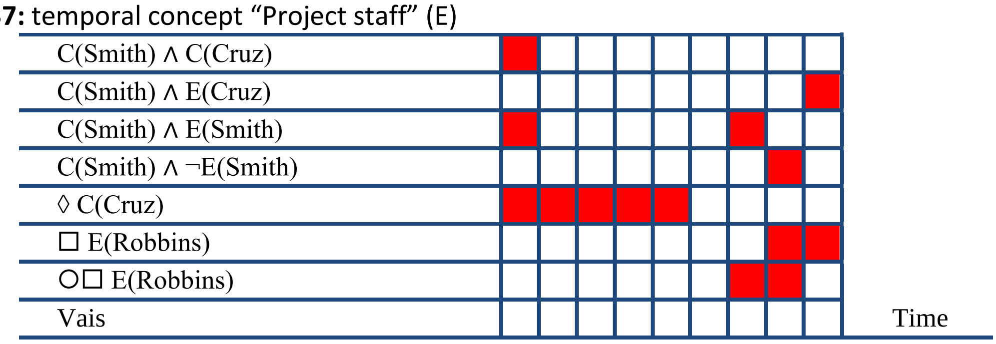


Figure 38: Examples of temporal ABox

5. Conclusions

By using numerous examples, the paper briefly describes the possibilities of applying the LTL logic to the description logic ALC. There are following issues for the future research:

- applying DL to the temporal logic CTL;
- extension of the DL family logic to which temporality is applied;
- temporal query languages syntax and semantics.

6. References

[1] A. Cau, B. Moszkowski, Interval Temporal Logic, 2021. URL: http://antonio-cau.co.uk/ITL/.

[2] C. Lutz, F. Wolter, and M. Zakharyaschev, Temporal description logics: a survey,. in 15th International Symposium on Temporal Representation and Reasoning IEEE Computer Society Press, 2008, pp. 3–14.

[3] A. Donzé, On signal temporal logic, in: International conference on runtime verification. Springer, 2013, pp. 382–383 .

[4] M. Mehrabian, M. Khayatian, A. Shrivastava, J.C. Eidson, P. Derler, H.A. Andrade, Y.-S. Li-Baboud, E. Griffor, M. Weiss, K. Stanton, Timestamp Temporal Logic (TTL) for Testing the Timing of Cyber-Physical Systems, *ACM Transactions on Embedded Computer Syst*ems 16(5), (2017), pp. 169:1 – 169:20.

[5] C. Eisner, D. Fisman A Practical Introduction to PSL, 2006, URL: https://link.springer.com/book/10.1007/978-0-387-36123-9.

[6] Wanwei Liu (2017) Basic Temporal Logic – LTL, CTL, CTL*, 2017. URL: https://data.educoder.net/api/attachments/667?disposition=inline.

[7] J. Ferlez, R. Cleaveland, S. Marcus, Bisimulation and Hennessy-Milner Logic for Generalized Synchronization Trees, in Combined Workshop on Expressiveness in Concurrency and Structural Operational Semantics (EXPRESS/SOS 2017). EPTCS 255, 2017, pp. 35–50

[8] CS INTRANET, Metric Temporal Logic: Tools and Experiments, 2017. URL: https://cgi.csc.liv.ac.uk/~ullrich/MTL/

[9] P. Bouyer, F. Laroussinie, N. Markey, J. Ouaknine, J. Worrel, Timed temporal logics, 2017. URL: https://hal.archives-ouvertes.fr/hal-01566439/document

[10] C. Eisner, D. Fisman, J. Havlicek, Y. Lustig, A. Mcisaac, D.V. Campenhout, Reasoning with Temporal Logic on Truncated Paths, 2017. URL: https://www.cs.bgu.ac.il/~dana/documents/EFHLMV_CAV03_full.pdf