Bridging the gap between modal temporal logics and constraint-based QSR[*] as an $\mathcal{ALC}(\mathcal{D})$ spatio-temporalisation with weakly cyclic TBoxes[**]

Amar Isli

Fachbereich Informatik, Universität Hamburg,
Vogt-Kölln-Strasse 30, D-22527 Hamburg, Germany
isli@informatik.uni-hamburg.de

Abstract. The aim of this work is to provide a family of qualitative theories for spatial change in general, and for motion of spatial scenes in particular. Motion of an n-object spatial scene is seen as a change (over time) of the (qualitative) spatial relations between the different objects involved in the scene —if, for instance, the spatial relations are those of the well-known Region-Connection Calculus $\mathcal{RCC}8$, then the objects of the scene are seen as regions of a topological space, and motion of the scene as a change in the $\mathcal{RCC}8$ relations on the different pairs of the objects. To achieve this, we consider a spatio-temporalisation $\mathcal{MTALC}(\mathcal{D}_x)$, of the well-known $\mathcal{ALC}(\mathcal{D})$ family of Description Logics (DLs) with a concrete domain: the $\mathcal{MTALC}(\mathcal{D}_x)$ concepts are interpreted over infinite k-ary trees, with the nodes standing for time points; the roles split into $m+n$ immediate-successor (accessibility) relations, which are antisymmetric and serial, and of which m are general, not necessarily functional, the other n functional; the concrete domain $\mathcal{D}_x$ is generated by an $\mathcal{RCC}8$-like spatial Relation Algebra (RA) x. The (long-term) goal is to design a platform for the implementation of flexible and efficient domain-specific languages for tasks involving spatial change. In order to capture the expressiveness of most modal temporal logics encountered in the literature, we introduce weakly cyclic Terminological Boxes (TBoxes) of $\mathcal{MTALC}(\mathcal{D}_x)$, whose axioms capture the decreasing property of modal temporal operators. We show the important result that satisfiability of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept with respect to a weakly cyclic TBox is decidable in nondeterministic exponential time, by reducing it to the emptiness problem of a weak alternating automaton augmented with spatial constraints, which we show to remain decidable, although the accepting condition of a run involves, additionally to the standard case, consistency of a CSP (Constraint Satisfaction Problem) potentially infinite. The result provides a tableaux-like satisfiability procedure which we will discuss. Finally, given the importance and cognitive plausibility of continuous change in the real physical world, we provide a discussion showing that our decidability result extends to the case where the nodes of the k-ary tree-structures are interpreted as (durative) intervals, and each of the $m+n$ roles as the *meets* relation of Allen's RA of interval relations.

Key words: Spatio-temporal reasoning, Qualitative reasoning, Modal temporal logics, Alternating automata, Description logics, Concrete domain, Constraint satisfaction.

---

[*] Qualitative Spatial Reasoning.

[**] This work was partly supported by the EU project "*Cognitive Vision systems*" (CogVis), under grant *CogVis IST 2000-29375.*

1 Introduction

Modal temporal logics are well-known in computer science in general, and in Artificial Intelligence (AI) in particular. Important issues that need to be addressed, when defining a modal temporal logic, include the ontological issue of whether to choose points or intervals as the primitive objects, and the issue of whether time is a total (linear) or a partial order. For more details, the reader is referred to books such as van Benthem's [76], or to survey articles such as Vila's [79].

Qualitative Spatial Reasoning (henceforth QSR), and more generally Qualitative Reasoning (QR), differs from quantitative reasoning by its particularity of remaining at a description level as high as possible. In other words, QSR sticks at the idea of "making only as many distinctions as necessary" [14], idea borrowed to naïve physics [35]. The core motivation behind this is that, whenever the number of distinctions that need to be made is finite, the reasoning issue can get rid of the calculation details of quantitative models, and be transformed into a simple matter of symbol manipulation; in the particular case of constraint-based QSR, this generally means a finite RA [75] (see also [17, 46, 54]), with tables recording the results of applying the different operations to the different atoms, and the reasoning issue reduced to a matter of table look-ups: a good illustration to this is the well-known topological calculus $\mathcal{RCC}8$ [67, 18]. One plausible way of responding to criticisms against QSR languages (which include Forbus, Nielsen and Faltings' [22] poverty conjecture, and Habel's [32] argument that such languages suffer from not having "the ability to refine discrete structures if necessary"), and against QR languages in general, is to define such languages according to cognitive adequacy criteria [69]. For more details on QSR, and on QR in general, the reader is referred to survey articles such as [14, 16].

Considered separately, modal temporal logics and constraint-based QSR have each an important place in AI. However, an important goal for research in AI, which has not received enough attention so far, is to define well-founded languages combining modal temporal logics and QSR languages. Such languages are key representational tools for tasks involving qualitative spatial change. Examples of such tasks include satellite-like surveillance of large-scale geographic spaces, and (qualitative) path planning for robot navigation [20, 25, 41, 48, 72, 83].

The goal of the present work is to enhance the expressiveness of modal temporal logics with qualitative spatial constraints. What we get is a family of qualitative theories for spatial change in general, and for motion of spatial scenes in particular. The family consists of domain-specific spatio-temporal (henceforth s-t) languages, and is obtained by spatio-temporalising a well-known family of description logics (DLs) with a concrete domain, known as $\mathcal{ALC}(\mathcal{D})$ [3]. $\mathcal{ALC}(\mathcal{D})$ originated from a pure DL known as $\mathcal{ALC}$ [73], with $m \geq 0$ roles all of which are general, not necessarily functional relations, and which Schild [71] has shown to be expressively equivalent to Halpern and Moses' $\mathcal{K}_{(m)}$ modal logic [33]. $\mathcal{ALC}(\mathcal{D})$ is obtained by adding to $\mathcal{ALC}$ functional roles (better known as abstract features), a concrete domain $\mathcal{D}$, and concrete features (which refer to objects of the concrete domain). The spatio-temporalisation of $\mathcal{ALC}(\mathcal{D})$ is obtained, as the name suggests, by performing two specialisations at the same time:

1. temporalisation of the roles, so that they consist of $m + n$ immediate-successor (accessibility) relations $R_1, \ldots R_m, f_1, \ldots, f_n$, of which the R_i's are general, not necessarily functional relations, and the f_i's functional relations; and

2. spatialisation of the concrete domain $\mathcal{D}$: the concrete domain is now $\mathcal{D}_x$, and is generated by a spatial RA x, such as the Region-Connection Calculus RCC8 [67].

The final spatio-temporalisation of $\mathcal{ALC}(\mathcal{D})$ will be referred to as $\mathcal{MTALC}(\mathcal{D}_x)$ ($\mathcal{MTALC}$ for $\mathcal{M}$odal $\mathcal{T}$emporal $\mathcal{ALC}$). To summarise, $\mathcal{MTALC}(\mathcal{D}_x)$ verifies the following:

1. the (abstract) domain (i.e., the set of worlds in modal logics terminology) of $\mathcal{MTALC}(\mathcal{D}_x)$ interpretations is a universe of time points;
2. the roles consist of $m+n$ immediate-successor relations $R_1, \ldots R_m, f_1, \ldots, f_n$, of which the R_i's are general, not necessarily functional relations, and the f_i's are functional relations;
3. the roles are antisymmetric and serial, and we denote, as usual, the transitive closure and the reflexive-transitive closure of a relation R by R^+ and R^*, respectively;
4. the concrete domain $\mathcal{D}_x$ is generated by an $\mathcal{RCC}$8-like constraint-based qualitative spatial language x; and
5. the concrete features are functions from the abstract domain onto objects of the concrete domain: in the case of x being $\mathcal{RCC}$8, for instance, the objects of the concrete domain are regions of a topological space.

When viewed as a domain-specific high-level vision system, a theory of the family has the following properties:

1. "the eyes" of the system are the concrete features: with each object O_i of the (changing) spatial domain at hand, we associate one and only one concrete feature g_i, which is given the task of "keeping an eye" on O_i's position as time flows;
2. the (concrete) feature chains other than the concrete features, which consist of finite chains (compositions) of abstract features terminated by a concrete feature, and allow to access from a given node of an interpretation, the value of a concrete feature at another, future node, constitute "the predictive" engine of the system;
3. the objects of the concrete domain are concrete objects of the spatial domain at hand; and
4. the predicates of the concrete domain constitute "the high-level memory" of the system, able of representing knowledge on objects of the spatial domain at hand, as seen by the concrete features or predicted by the feature chains, as spatial constraints on tuples of the corresponding concrete features or feature chains.

The idea of domains in LP (Logic Programming) [7, 15, 44, 45, 77] has led to CLP (Constraint Logic Programming) with a specific domain, such as $CLP(\mathbb{Q})$ and $CLP(\mathbb{N})$, known to be efficient implementations of CLP with, respectively, the rationals and the integers as a specific domain. The motivation behind the extension of $\mathcal{ALC}$ to $\mathcal{ALC}(\mathcal{D})$ was similar, in that with $\mathcal{ALC}(\mathcal{D})$ we can refer directly to objects of the domain we are interested in, thanks to the concrete features, and to the (concrete) feature chains in general, and record knowledge on these objects, as "seen" by the concrete features or "predicted" by the feature chains, thanks to the predicates. This allows the objects of the domain at hand, and the knowledge on them, to be isolated from the knowledge on the abstract objects, sufficiently enough to allow for an easy and efficient implementation. In this respect, $\mathcal{MTALC}(\mathcal{D}_{\mathcal{RCC}8})$, for instance, will be an implementation of $\mathcal{MTALC}$, the temporal component, with a concrete domain $\mathcal{D}_{\mathcal{RCC}8}$ generated by $\mathcal{RCC}8$.

$\mathcal{MTALC}(\mathcal{D}_x)$ is the result of combining a temporalisation of a pure DL language, $\mathcal{ALC}$ [73], with a spatialisation of a constraint-based language reflected by a concrete domain $\mathcal{D}$. The discussion of the previous paragraph, on the separation of the knowledge on the objects of the domain at hand, from the knowledge on the abstract objects, leads to an important computational advantage in the use of this way of getting spatio-temporal languages, instead of combining two or more modal logics, which is known to potentially lead to undecidable spatio-temporal languages —even when the combined parts are tractable [8, 9]! With DLs, it is known that, as long as the pure DL and the constraint-based language reflected by the concrete domain are decidable, the resulting DL with a concrete domain is so that satisfiability of a concept w.r.t. an acyclic TBox is decidable [3].

Constraint-based languages candidate for generating a concrete domain for a member of our family of spatio-temporal theories, are spatial RAs for which the atomic relations form a decidable subset —i.e., such that consistency of a CSP expressed as a conjunction of n-ary relations on n-tuples of objects, where n is the arity of the RA relations, is decidable. Examples of such RAs known in the literature include, the Region-Connection Calculus $\mathcal{RCC}8$ in [67] (see also [18]), the Cardinal Directions Algebra $\mathcal{CDA}$ in [23], and the rectangle algebra in [6] (see also [30, 57]) for the binary case; and the RA $\mathcal{CYC}_t$ of 2D orientations in [42, 43] for the ternary case.

The paper, without loss of generality, will focus on two concrete domains generated by two of the three binary spatial RAs mentioned above, $\mathcal{RCC}8$ [67] and $\mathcal{CDA}$ [23]; and on a third concrete domain generated by the ternary spatial RA $\mathcal{CYC}_t$ in [42, 43]. It is known that, in the general case, satisfiability of an $\mathcal{ALC}(\mathcal{D})$ concept with respect to a cyclic Terminological Box (TBox) is undecidable (see, e.g., [52]). In this work, we define a weak form of TBox cyclicity,[1] which captures the decreasing property of modal temporal operators. The pure DL $\mathcal{MTALC}$, consisting of the temporal component of $\mathcal{MTALC}(\mathcal{D}_x)$, together with weakly cyclic TBoxes, captures the expressiveness of most modal temporal logics encountered in the literature. Similarly to *eventuality* formulas in modal temporal logics, some of the defined concepts of $\mathcal{MTALC}(\mathcal{D}_x)$ will be referred to as *eventuality* concepts, for the axioms defining them describe situations that need to be effectively satisfied sometime in the future. As an example of such defined concepts, the concept C defined by the axiom $C \doteq p \sqcup \exists f.C$, where p is an atomic proposition, and f an immediate-successor accessibility relation, associating with each state of an interpretation its immediate successor, describes the eventuality formula $\Diamond p$, of, say, Propositional Linear Temporal Logic (PLTP). For a state s of an interpretation to satisfy the eventuality formula $\Diamond p$, there should exist a descendent node of s, along the infinite path f^ω, satisfying p. The axiom $C \doteq p \sqcup \exists f.C$, however, may leave p unsatisfied, and still give the false impression that it is satisfied: this is a well-known situation in modal temporal logics, which may happen by eternally reporting the satisfiability to the next state, and which can be get rid of by having recourse to the theory of automata on infinite objects (see, e.g., [40, 78]). This discussion is important for the understanding of the way are obtained the accepting states of the

[1] Intuitively, a TBox is weakly cyclic if all its cycles are of degree 1, reflected by a defined concept appearing in the right hand side of the axiom defining it (within the scope of an exitential or universal quantifier).

(weak) alternating automaton on infinite objects, associated with the satisfiability of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept w.r.t. an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox.

In a nondeterministic automaton on infinite words, the transition function, say δ, is so that $\delta(a,q)$, where a is a letter of the alphabet and q an element of the set Q of states, is a subset of Q; and a state q accepts a word $u = av$, if and only if (iff) there exists a state $q' \in \delta(a,q)$ accepting the suffix v. In alternating automata, $\delta(a,q)$, when put in a certain normal form, is a set of subsets of Q; and a state q accepts a word $u = av$, iff there exists a set of states $Q' \in \delta(a,q)$, such that each state q' in Q' accepts the suffix v. This intuitive definition of acceptance works only when no condition is imposed on the so called run of the automaton on the the input (infinite) word. When such a condition is imposed, the acceptance requires more: in the case of Büchi automata, for instance, where the accepting condition is given by a subset $\mathcal{F}$ of the set of states, the states repeated infinitely often in any branch of the run should include a state of $\mathcal{F}$.

The theory of alternating automata [59] on infinite words, and on infinite trees in general, is a generalisation of the theory of nondeterministic automata [65]. In this work, we will mostly need weak alternating automata [58].

Furthermore, we will need to extend standard alternating automata, in order to handle interpretations "following" the evolution of a spatial scene of interest over time, by recording at each node, thanks to concrete features, the positions of the objects of the scene. The new kind of alternating automata handle spatial constraints on concrete features, and on (concrete) feature chains in general, which allow them to restrict the values of the different concrete features at the different nodes of an interpretation. As a consequence, the accepting condition will involve, not only the states infinitely often repeated in a run, but a CSP as well, which is potentially infinite. We show that the "injection" of spatial constraints into standard weak alternating automata does not compromise decidability of the emptiness problem, which we show can be achieved in nondeterministic exponential time.

We prove the important result that, satisfiability of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept with respect to an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox is decidable, by reducing it to the emptiness problem of a weak alternating automaton augmented with spatial constraints. A first discussion will show that the result provides an effective tableaux-like satisfiability procedure, with the particularity of dynamically handling spatial constraints, using constraint propagation techniques, which allows it to potentially reduce the search space. A second discussion will clarify how the decidability result extends to the case where the nodes of the k-ary tree-structures are interpreted as (durative) intervals, and each of the $m+n$ roles as the *meets* relation of Allen's RA of interval relations [1].

Weakly definable languages of infinite words (or ω-words), and of infinite k-ary trees in general, were first defined by Rabin [66]: a language L is weakly definable iff L and its complement $\overline{L}$ are Büchi (L is Büchi iff there exists a Büchi automaton accepting L). Muller, Saoudi and Schupp [58] have shown (1) that a language is weakly definable iff there exists a weak alternating automaton accepting it; and (2) that a weak alternating automaton can be simulated with a Büchi nondeterministic automaton (see also [60]). The emptiness problem of a Büchi nondeterministic automaton, in turn, is known to be trivially decidable [65].

Computing with alternating automata is easy. In particular, the complement of a language accepted by an alternating automaton M, is the language of the alternating automaton M', obtained from M by dualising the transition function (interchanging the $\wedge$ and $\vee$ operators), and complementing the accepting condition (see the complementation theorem in [59]).

The particularity of weak alternating automata is that, one can find a partition $Q = Q_1 \cup \cdots \cup Q_m$ of the set Q of states, and a partial order $\geq$ on this partition, so that if $q' \in \delta(a, q)$, then there exist $i, j \in \{1, \cdots, m\}$, such that $q \in Q_i \wedge q' \in Q_j \wedge Q_i \geq Q_j$.

1.1 The way from *LP* to distributed spatial *CLP* (*dsCLP*)

The theoretical framework of Logic Programming (*LP*) is the propositional calculus. A logic program can be seen as deciding satisfiability of a conjunction of Horn clauses. In reality, however, one cannot always restrict the knowledge on the real world, to Boolean combinations of atomic, Boolean propositions. One has to face the problem of representing, and dealing with, knowledge on a specific domain of interest, generally consisting of objects referred to by variables. Such knowledge is represented using constraints of the form $P(x_1, \ldots, x_n)$, where P is an n-ary predicate, and the x_i are variables. Such constraints are referred to as *domains* in *CLP*.

CLP incorporates the idea of domains into *LP*, so that, for instance, the user can restrict the range of a variable, or of a pair of variables, either with the use of a unary constraint (on a variable), or with the use of a binary constraint (on a pair of variables). Search algorithms based on consistency techniques [7, 15, 44, 45, 77], can then use a priori pruning during the search, generally by applying a filtering algorithm, such as arc-consistency [53, 56], at each node of the search tree, which potentially reduces the domains of the variables and of the pairs of variables (thus reducing the subtree of the search space rooted at the current node[2]). Modal temporal logics, in turn, can be seen as a mean for distributing LP, which leads to distibuted LP (dLP). Adding variable (and pair-of-variable) domains to dLP, in a similar way as in LP, leads to distributed *CLP* (*dCLP*). The general use, and understanding of, a domain in *CLP* and in *dCLP*, is as a unary constraint, of the form $R(x)$, stating that variable x is constrained to belong to the unary predicate R; or as a binary constraint, of the form $S(x, y)$, stating that the pair (x, y) of variables is constrained to belong to the binary predicate S: R is nothing else than a subset of the whole universe of values used for variables' instantiation, and S a subset of the cross product of the universe by itself (the universe is generally the set $\mathbb{R}$ of reals, or any of its subsets, such as the set $\mathbb{Q}$ of rationals, the set $\mathbb{N}$ of integers, or the set $\{0, 1\}$ symbolising the Booleans).

In QSR, restricting the domain of a variable to a strict subset of the (continuous) spatial domain at hand (the two-dimensional space, for instance), has generally no practical importance. In other words, unary constraints do not have as much importance as in traditional *CLP*. Indeed, a conjunction of QSR constraints (in other words, a QSR CSP) is always node- and arc-consistent. Examples of QSR constraint languages for which this is the case, include languages of binary relations we have already mentioned: the Region-Connection Calculus in [67, 18], the Cardinal Directions Algebra in [23] and the rectangle algebra in [6, 30, 57]. For these languages, the

[2] By current node, we mean the node of the tree-like search space where the search is.

lowest local-consistency filtering, which can potentially reduce the search space, is path consistency. This is not the end of the story: there are QSR constraint languages for which CSPs expressed in them are already strongly 3-consistent (i.e., node-, arc- and path-consistent), and for which effectiveness of local-consistency filtering starts from 4-consistency (for instance, the ternary RA of 2D orientations in [42, 43]).

It should be clear now that, in order to be able to reason qualitatively about space within *CLP* or within *dCLP*, one has to adapt it, so that the domains reflect the reality of the spatial domain at hand: the domains should be binary constraints, if the constraint language used for representing spatial knowledge is binary (such as the three mentioned above), and ternary constraints, if the language is ternary (such as the one mentioned above). To make things clearer, suppose that the representational language is $\mathcal{RCC}8$. The spatial domain at hand, used for instantiating the spatial variables, is then the set of regions of a topological space. A pair of spatial variables of a *CLP* program, if no restriction is given, is related by the $\mathcal{RCC}8$ universal relation, which is the set of the eight atoms of the language. A domain in this case is any subset of the set of all eight atoms (in other words, any $\mathcal{RCC}8$ relation), which will restrict the instantiations of pairs of variables, to those pairs of regions of the topological space at hand that are related by an atom of the domain.

We refer to *CLP* and *dCLP* with spatial variables as described above, as *sCLP* (spatial *CLP*) and *dsCLP* (distributed *sCLP*).

1.2 Associating a weak alternating automaton with the satisfiability of a concept w.r.t. a weakly cyclic TBox: an overview

Given an $\mathcal{MTALC}(\mathcal{D}_x)$ concept C and an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox $\mathcal{T}$, the problem we will be interested in is, the satisfiability of C with respect to $\mathcal{T}$. The axioms in $\mathcal{T}$ are of the form $B \doteq E$, where B is a defined concept name, and E an $\mathcal{MTALC}(\mathcal{D}_x)$ concept. Using C, we introduce a new defined concept name, B_{init}, given by the axiom $B_{init} \doteq C$. We denote by $\mathcal{T}'$ the TBox consisting of $\mathcal{T}$ augmented with the new axiom: $\mathcal{T}' = \mathcal{T} \cup \{B_{init} \doteq C\}$. The alternating automaton we associate with the satisfiability of C w.r.t. the TBox $\mathcal{T}$, so that satisfiability holds iff the language accepted by the automaton is not empty, is now almost entirely given by the TBox $\mathcal{T}'$: the defined concept names represent the states of the automaton, B_{init} being the initial state; the transition function is given by the axioms themselves. However, some modification of the axioms is needed.

Given an $\mathcal{MTALC}(\mathcal{D}_x)$ axiom $B \doteq E$ in $\mathcal{T}'$, the method to be proposed decomposes E into some kind of Disjunctive Normal Form, $dnf2(E)$, which is free of occurrences of the form $\forall R.E$. Intuitively, the concept E is satisfiable by the state consisting of the defined concept name B, iff there exists an element S of $dnf2(E)$ that is satisfiable by B. An element S of $dnf2(E)$ is a conjunction written as a set, of the form $S_{prop} \cup S_{csp} \cup S_{\exists}$, where:

1. S_{prop} is a set of primitive concepts and negated primitive concepts —it is worth noting here that, while the defined concepts (those concept names appearing as the left hand side of an axiom) define the states of our automaton, the primitive concepts (the other concept names) correspond to atomic propositions in, e.g., classical propositional calculus;

2. S_{csp} is a set of concepts of the form $\exists(u_1)\cdots(u_n).P$, where $u_1,\ldots,u_n$ are feature chains and P a relation (predicate) of an n-ary spatial RA; and
3. $S_\exists$ is a set of concepts of the form $\exists R.E_1$, where R is a role and E_1 is a concept.

The procedure ends with a TBox $\mathcal{T}'$ of which all axioms are so written. Once $\mathcal{T}'$ has been so written, we denote:

1. by $af(\mathcal{T}')$, the set of abstract features appearing in $\mathcal{T}'$; and
2. by $rrc(\mathcal{T}')$, the set of concepts appearing in $\mathcal{T}'$, of the form $\exists R.E$, with R being a general, not necessarily functional role, and E a concept.

The alternating automaton to be associated with $\mathcal{T}'$, will operate on (Kripke) structures which are infinite $m+p$-ary trees, with $m=|af(\mathcal{T}')|$ and $p=|rrc(\mathcal{T}')|$. Such a structure, say t, is associated with a truth-value assignment function π, assigning to each node, the set of those primitive concepts appearing in $\mathcal{T}'$ that are true at the node. With t are also associated the concrete features appearing in $\mathcal{T}'$: such a concrete feature, g, is mapped at each node of t, to a (concrete) object of the spatial domain in consideration (e.g., a region of a topological space if the concrete domain is generated by $\mathcal{RCC}8$).

The feature chains are of the form $f_1\ldots f_k g$,[3] with $k\geq 0$, where the f_i's are abstract features (also known, as alluded to before, as functional roles: functions from the abstract domain onto the abstract domain), whereas g is a concrete feature (a function from the abstract domain onto the set of objects of the concrete domain). The sets S are used to label the nodes of the search space. Informally, a run of the tableaux-like search space is a disjunction-free subspace, obtained by selecting at each node, labelled, say, with S, one element of $dnf2(S)$.

Let σ be a run, s_0 a node of σ, and S the label of s_0, and suppose that S_{csp} contains $\exists(u_1)(u_2).P$ (we assume, without loss of generality, a concrete domain generated by a binary spatial RA, such as $\mathcal{RCC}8$ [67, 18]), with $u_1=f_1\ldots f_k g_1$ and $u_2=f'_1\ldots f'_m g_2$. The concept $\exists(u_1)(u_2).P$ gives birth to new nodes of the run, $s_1=f_1(s_0), s_2=f_2(s_1),\ldots,s_k=f_k(s_{k-1}), s_{k+1}=f'_1(s_0), s_{k+2}=f'_2(s_{k+1}),\ldots,s_{k+m}=f'_m(s_{k+m-1})$; to new variables of what could be called the (global) CSP, $CSP(\sigma)$, of σ; and to a new constraint of $CSP(\sigma)$. The new variables are $\langle s_k,g_1\rangle$ and $\langle s_{k+m},g_2\rangle$, which denote the values of the concrete features g_1 and g_2 at nodes s_k and s_{k+m}, respectively. The new constraint is $P(\langle s_k,g_1\rangle,\langle s_{k+m},g_2\rangle)$. The set of all such variables together with the set of all such constraints, generated by node s_0, give the *CSP* $CSP_\sigma(s_0)$ of σ at s_0; and the union of all *CSP*s $CSP_s(\sigma)$, over the nodes s of σ, gives $CSP(\sigma)$. As the discussion shows, *dsCLP* does not reduce to a mere distribution of *sCLP*, consisting of *sCLP* at each node, with additionally temporal precedence on the different nodes: the feature chains make it possible to refer to the values of the different concrete features at the different nodes of a run, and restrict these values using spatial predicates.

The pruning process during the tableaux method will now work as follows. The search will make use of a data structure *Queue*, which will be handled in very much the same fashion as such a data structure is handled in local consistency algorithms,

[3] Throughout the rest of the paper, a feature chain $f_1\ldots f_k g$ is interpreted as within the Description Logics Community —i.e., as the composition $f_1\circ\ldots\circ f_k\circ g$: we remind the user that $(f_1\circ\ldots\circ f_k\circ g)(x)=g(f_k(f_{k-1}(\ldots(f_2(f_1(x))))))$.

such as arc- or path-consistency [53, 56], in standard *CSP*s. The data structure is initially empty. Then whenever a new node s is added to the search space, the global *CSP* of the run being constructed is updated, by augmenting it with (the variables and) the constraints generated, as described above, by s. Once the *CSP* has been updated, so that it includes the local *CSP* at the current node, the local consisteny pruning is applied by propagating the constraints in *Queue*. Once a run has been fully constructed, and only then, its global *CSP* is solved. In the case of a concrete domain generated by a binary, $\mathcal{RCC}8$-like RA, the filtering is achieved with a path-consisteny algorithm [1], and the solving of the global *CSP*, after a run has been fully constructed, with a solution search algorithm such as Ladkin and Reinefeld's [47]. In the case of a concrete domain generated by a ternary spatial RA, the filtering and the solving processes are achieved with the 4-consistency and the search algorithms in [42, 43].

1.3 The relation to Bayesian networks

In the case of feature chains of length one (i.e., reducing to concrete features), we will discuss how to combine the predicate concepts, of the form $\exists(g_1)(g_2).P$, with conditional probabilities, which will make the relation, at the current state, on a pair of concrete features, dependant only on the relation on the same pair at the previous state: the conditional probabilities will provide, for the relation on a pair of concrete features, the probability to be s, given that it was r at the previous state, r and s being atoms of the spatial RA x. This will give us a family of $\mathcal{ALC}(\mathcal{D})$-like languages, $\mathcal{BNALC}(\mathcal{D}_x)$, for probabilistic, Bayesian-network-like reasoning (see, e.g., [63]), about qualitative spatio-temporal knowledge. This is particularly important for prediction [70] in, for instance, scene interpretation in high-level computer vision. One possibility of setting the conditional probabilities is to learn them. Another is to assume continuous change and uniform probability distribution: the conditional probabilities can then be derived straightforwardly from what is known in QSR as the theory of *conceptual neighbourhoods* (see, e.g., [24]).

1.4 Related work

Motion and spatial change as s-t histories According to [34], s-t histories are space-time regions traced by objects over time. For the n-dimensional (n-d for short) space, a s-t history is an $n+1$-d volume. Such a history (of an object, or of a scene in general, of the n-d space) can be recorded by associating with the flow of time a camera "filming" the scene. The approach we propose respects this view of spatial change, and of motion in particular. Each member of our family of theories is an $\mathcal{ALC}(\mathcal{D})$-like DL, with temporalised roles, and a spatial concrete domain. The temporalised roles allow the DL to capture the flow of time. The spatial concrete domain, in some sense, plays the role of a camera:

1. the concrete features can be seen, as already argued, as the eyes of the camera (one eye per object of the spatial scene at hand); and
2. the knowledge on the spatial scene, as perceived by the camera's eyes, is (qualitatively) recorded by the predicates, as spatial constraints on tuples of the objects involved in the scene.

Contrary to other approaches [36, 37, 61], ours makes clear the borderline between the temporal component and the spatial component. Indeed, the general $\mathcal{ALC}(\mathcal{D})$ framework [3] was originally inspired by domain-specific Constraint Logic Programming (*CLP*) [7, 15, 44, 45, 77], which, as already explained, gave birth to many efficient and flexible domain-specific implementations of *CLP* (for instance, $CLP(\mathbb{R})$, $CLP(\mathbb{Q})$, $CLP(\mathbb{N})$, *CLP*(*Intervals*)). Each theory of our family can give birth to an efficient and flexible implementation of *dsCLP* with a specific spatial domain (each $\mathcal{RCC}8$-like RA can generate such a domain).

Approaches based on multi-dimensional modal logics Approaches based on multi-dimensional modal logics for the representation of s-t knowledge, exist in the literature [5, 8, 9, 27, 82]. Their main disadvantage is that, their spatial component, for instance, can represent only some specific spatial knowledge (e.g., topological knowledge). In our case, whenever a new $\mathcal{RCC}8$-like spatial RA is found, it can be used to generate a spatial concrete domain, and augment our family with a new theory for spatial change. The new theory, in turn, can be implemented as an efficient and flexible domain-specific, *CLP*-like language for tasks of the s-t domain. If an implementation of a theory of the family already exists, then the implementation of the new theory only needs to adapt the old implementation to the new concrete domain —which does not require much work. Another disadvantage of multi-modal logics is that, even combining tractable modal logics may lead to an undecidable multi-modal logic (see, for instance, [8, 9]).

1.5 Examples of potential applications

Geographical Information Systems (GIS) GIS is known to be one of the priviledged application domains of constraint-based QSR (see, for instance, [14]). Among QSR languages that have GIS as a direct application domain, the calculus of cardinal directions in [23] from the orientational side, and the $\mathcal{RCC}8$ calculus [67, 18] from the topological side. Each of these two languages, as already discussed, can generate one member of our family $\mathcal{MTALC}(\mathcal{D}_x)$ of qualitative theories for spatial change, which can be used for geographic change.

High-level computer vision By high-level computer vision (see, e.g., [62]), we mean that it is not necessary to have knowledge on the precise location of the different parts of, say, the moving spatial scene. Rather, we are interested in describing, qualitatively, the relative position of the different parts. The language used for the description is a qualitative language of spatial relations, in the style of the binary RAs $\mathcal{RCC}8$ [67, 18] and $\mathcal{CDA}$ [23], and the ternary RA $\mathcal{CYC}_t$ in [42, 43]: qualitative knowledge on relative position of objects of the spatial domain at hand is represented as constraints consisting of relations of the RA on n-tuples of the objects, where n is the arity of the relations of the RA.

Qualitative path planning for robot navigation There are constraint-based spatial languages in the literature considered as well-suited for path planning for robot navigation. These include Freksa's ternary calculus [25, 83] of relative orientation of 2-d points, as well as Isli and Cohn's ternary RA $\mathcal{CYC}_t$ of 2-d orientations [42, 43]. This is however misleading, for the languages are spatial, and not spatio-temporal. The best they can offer is the representation of a snapshot of a spatial change, in particular the snapshot of a motion of, say, a spatial scene. The reason to that is that the languages do not capture the flow of time at all. However, each can be used to generate a spatial concrete domain for a member of our family of theories of spatial change.

1.6 Background on binary relations

Given a set A, we denote by $|A|$ the cardinality of A. A binary relation, R, on a set S is any subset of the cross product $S \times S = \{(x, y) : x, y \in S\}$. Such a relation is reflexive iff $R(x, x)$, for all $x \in S$; it is symmetric iff, for all $x, y \in S$, $R(y, x)$, whenever $R(x, y)$; it is transitive iff, for all $x, y, z \in S$, $R(x, z)$, whenever $R(x, y)$ and $R(y, z)$; it is irreflexive iff, for all $x \in S$, $\neg R(x, x)$; it is antisymmetric iff, for all $x, y \in S$, if $R(x, y)$ and $R(y, x)$ then $y = x$; and it is serial iff, for all $x \in S$, there exists $y \in S$ such that $R(x, y)$. The transitive (resp. reflexive-transitive) closure of R is the smallest relation R^+ (resp. R^*), which includes R and is transitive (resp. reflexive and transitive). Finally, R is functional if, for all $x \in S$, $|\{y \in S : R(x, y)\}| \leq 1$; it is nonfunctional otherwise.

1.7 Background on computational complexity

The computational complexity of a given problem is a measure of the cost of solving it, in terms of the amount of time or space it needs, as a function of the problem's size. A deterministic computation is characterised by the unicity, at any time, of the step to consider next. A nondeterministic computation is one that needs to "guess", among a finite number of steps, which to consider next. There are five main complexity classes, P, PSPACE, EXP, NP, and NEXP, which characterise, respectively, the problems that are solvable in deterministic polynomial time, in deterministic polynomial space, in deterministic exponential time, in nondeterministic polynomial time, and in nondeterministic exponential time. It is known that P $\subseteq$ NP $\subseteq$ PSPACE $\subseteq$ EXP $\subseteq$ NEXP and P $\neq$ EXP.

Intuitively, a problem A is hard w.r.t. a complexity class $\mathcal{C} \in \{$P,NP,PSPACE,EXP,NEXP$\}$, or $\mathcal{C}$-hard for short, if every problem B in $\mathcal{C}$ can be polynomially reduced to A, so that an algorithm for B can be "easily" obtained from an algorithm for A. A problem is complete w.r.t. a complexity class $\mathcal{C}$, or $\mathcal{C}$-complete for short, if it is in $\mathcal{C}$ and is $\mathcal{C}$-hard. The reader is referred to [29, 38] for details.

1.8 Assumptions on the structure of time

We make the following assumptions on the structure of time:

1. time is discrete;
2. it has an initial moment with no predecessors; and

3. it is branching and infinite into the future, and all moments have the same number of immediate-successor moments.

Temporal formulas will be interpreted over temporal structures consisting of infinite k-ary Σ-trees, with $k \geq 1$ and $\Sigma = 2^{\mathcal{P}}$, $\mathcal{P}$ being a countably infinite set of atomic propositions. Such a structure is of the form $t =< K^*, \pi, R^* >$:

1. $K = \{d_1, \ldots, d_k\}$ is a set of k directions —K^* is thus the set of finite words over K, representing the set of nodes of t;
2. $\pi : \Sigma^* \rightarrow 2^{\mathcal{P}}$ is a truth assignment function, mapping each node x of t into the set of atomic propositions true at x; and
3. R is a serial, irreflexive and antisymmetric k-ary accessibility relation, mapping each node to its k immediate successors —R^* is thus the reflexive-transitive closure of R.

A simpler way of representing t is to remove the reflexive-transitive closure symbol, $*$, from R^*; i.e., as $t =< K^*, \pi, R >$. But there is even simpler: as a mapping $t : K^* \rightarrow 2^{\mathcal{P}}$.

The structure of time so described should be augmented with functions (concrete features), each of which is associated with an object of the spatial concrete domain of interest, and records, at each node (time point) of the structure, the position of that associated object.

2 A quick overview of the spatial relations to be used as the predicates of the concrete domain

We provide a quick overview of the spatial relations to be used, in the family $\mathcal{MTALC}(\mathcal{D}_x)$ of qualitative theories for spatial change, as the predicates of the concrete domain. But we first remind some general points we discussed in the previous sections:

1. $\mathcal{MTALC}(\mathcal{D}_x)$ is a spatio-temporalisation of $\mathcal{ALC}(\mathcal{D})$ [3]:
   (a) the abstract objects of $\mathcal{MTALC}(\mathcal{D}_x)$ (tree-like) interpretations are time points;
   (b) $\mathcal{MTALC}(\mathcal{D}_x)$ roles consist of $m + n$ immediate-successor accessibility relations, which are antisymmetric and serial, and of which m are general, not necessarily functional, the remaining n functional;
   (c) the concrete domain $\mathcal{D}_x$ is generated by a spatial RA x chosen as a tool for representing knowledge on n-tuples of objects of the spatial domain at hand, where n is the arity of the x relations —stated otherwise, the x relations will be used as the predicates of $\mathcal{D}_x$.
2. For clarity of presentation, we focus in this work on x being either of three RAs we have already mentioned: either of the two binary RAs $\mathcal{RCC}8$ [67, 18] and $\mathcal{CDA}$ [23], or the ternary RA $\mathcal{CYC}_t$ of 2-d orientations in [42, 43].
3. The work generalises, in an obvious way, to all spatial RAs x for which the atoms are Jointly Exhaustive and Pairwise Disjoint (henceforth JEPD), and such that the atomic relations form a decidable subclass: these include the binary rectangle algebra in [6, 30, 57], whose atomic relations form a tractable subset [6].

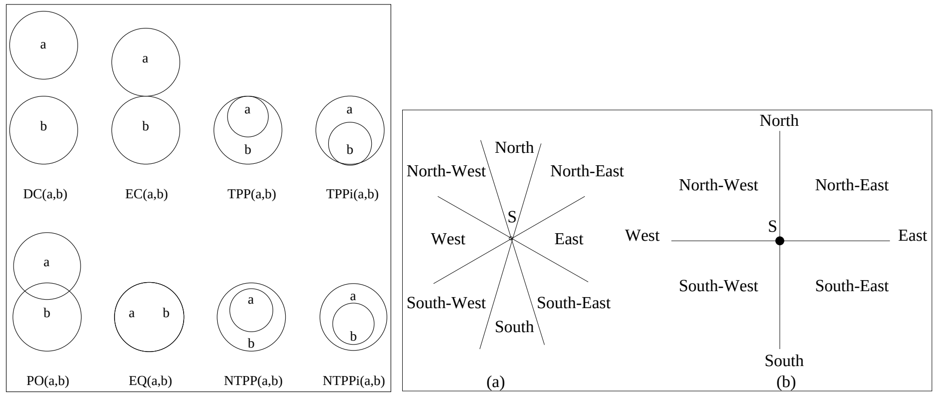

Fig. 1. (Left) An illustration of the RCC-8 atoms. (Right) Frank's cone-shaped (a) and projection-based (b) models of cardinal directions.

2.1 The RA $\mathcal{RCC}8$

Topology is one of the most developed aspects within the QSR Community. This is illustrated by the well-known RCC theory [67], from which derives the already mentioned RCC-8 calculus [67, 18]. The RCC theory, on the other hand, stems from Clarke's "calculus of individuals" [13], based on a binary "connected with" relation —sharing of a point of the arguments. Clarke's work, in turn, was developed from classical mereology [49, 50] and Whitehead's "extensionally connected with" relation [81]. The RCC-8 calculus [67, 18] consists of a set of eight JEPD atoms, DC (DisConnected), EC (Externally Connected), TPP (Tangential Proper Part), PO (Partial Overlap), EQ (EQual), $NTPP$ (Non Tangential Proper Part), and the converses, $TPPi$ and $NTPPi$, of TPP and $NTPP$, respectively. The reader is referred to Figure 1(Left) for an illustration of the atoms.

2.2 The RA $\mathcal{CDA}$

Frank's models of cardinal directions in 2D [23] are illustrated in Figure 1(Right). They use a partition of the plane into regions determined by lines passing through a reference object, say S. Depending on the region a point P belongs to, we have $No(P,S)$, $NE(P,S)$, $Ea(P,S)$, $SE(P,S)$, $So(P,S)$, $SW(P,S)$, $We(P,S)$, $NW(P,S)$, or $Eq(P,S)$, corresponding, respectively, to the position of P relative to S being *north*, *north-east*, *east*, *south-east*, *south*, *south-west*, *west*, *north-west*, or *equal*. Each of the two models can thus be seen as a binary RA, with nine atoms. Both use a global, *west-east/south-north*, reference frame. We focus our attention on the projection-based model (part (b) in Figure 1(Right)), which has been assessed as being cognitively more adequate [23].

2.3 The RA $\mathcal{CYC}_t$

The set 2D$\mathcal{O}$ of 2-d orientations is defined in the usual way, and is isomorphic to the set of directed lines incident with a fixed point, say O. Let h be the natural isomorphism, associating with each orientation x the directed line (incident with O) of orientation x. The angle $\langle x, y\rangle$ between two orientations x and y is the anticlockwise angle $\langle h(x), h(y)\rangle$. Isli and Cohn [42, 43] have defined a binary RA of 2D orientations, $\mathcal{CYC}_b$, that contains four atoms: e (equal), l (left), o (opposite) and r (right). For all $x, y \in 2\mathrm{D}\mathcal{O}$:

$$\begin{aligned} e(y,x) &\Leftrightarrow \langle x,y\rangle = 0 \\ l(y,x) &\Leftrightarrow \langle x,y\rangle \in (0,\pi) \\ o(y,x) &\Leftrightarrow \langle x,y\rangle = \pi \\ r(y,x) &\Leftrightarrow \langle x,y\rangle \in (\pi,2\pi) \end{aligned}$$

Based on $\mathcal{CYC}_b$, Isli and Cohn [42, 43] have built a ternary RA, $\mathcal{CYC}_t$, for cyclic ordering of 2D orientations: $\mathcal{CYC}_t$ has 24 atoms, thus 2^{24} relations. The atoms of $\mathcal{CYC}_t$ are written as $b_1b_2b_3$, where b_1, b_2, b_3 are atoms of $\mathcal{CYC}_b$, and such an atom is interpreted as follows: $(\forall x, y, z \in 2\mathrm{D}\mathcal{O})(b_1b_2b_3(x,y,z) \Leftrightarrow b_1(y,x) \wedge b_2(z,y) \wedge b_3(z,x))$. The reader is referred to [42, 43] for more details.

3 The $\mathcal{MTALC}(\mathcal{D}_x)$ description logics, with $x \in \{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$

Description Logics (DLs) constitute a knowledge representation family with a well-defined semantics, contrary to their ancestors, such as semantic networks [64] or frame systems [55]. Their main advantage is that they are highly expressive while still remaining decidable, or offering interesting decidable restrictions. One of the most important DLs in the literature is Schmidt-Schauss and Smolka's $\mathcal{ALC}$ [73]. $\mathcal{ALC}$ includes concepts and roles, which are, respectively, unary relations and binary relations on a set of (abstract) objects. One drawback of $\mathcal{ALC}$ is that it does not offer a way of referring to objects of a specific domain of interest, such as a spatial domain, where the objects could be regions of a topological space, orientations of the 2-dimensional space, or points. To get rid of this insufficiency, $\mathcal{ALC}$ has been extended to what is known as $\mathcal{ALC}(\mathcal{D})$, which augments $\mathcal{ALC}$ which a concrete domain D, consisting of a universe of objects, and of predicates for representing knowledge on these objects [3]. The roles in $\mathcal{ALC}$ are binary relations in the general meaning of the term, in the sense that they are not necessarily functional; in $\mathcal{ALC}(\mathcal{D})$, however, they split into general, not necessarily functional roles (referred to simply as roles), and functional roles (also referred to as abstract features).

Temporalisations of DLs are known in the literature (see, e.g., [2, 11]); as well as spatialisations of DLs (see, e.g., [31]). The present work considers a spatio-temporalisation of the well-known family $\mathcal{ALC}(\mathcal{D})$ of DLs with a concrete domain [3]. Specifically, we consider a temporalisation of the roles of the family, together with a spatialisation of its concrete domain.

3.1 Concrete domain

The role of a concrete domain in so-called DLs with a concrete domain [3], is to give the user of the DL the opportunity to "touch" at, and directly refer to, the application domain at hand. Specifically, the opportunity to represent, thanks to predicates, knowledge on objects of the application domain, as constraints on tuples of these objects. The set of objects of the application domain on which a concrete domain $\mathcal{D}$ represents knowledge, is referred to as $\Delta_{\mathcal{D}}$; and the set of predicates used by the concrete domain as a tool for representing knowledge on tuples of objects from $\Delta_{\mathcal{D}}$, constitutes the set $\Phi_{\mathcal{D}}$ of predicates of $\mathcal{D}$. As such, a concrete domain can be seen as opening a DL a window to the application domain. The difference between a DL without, and a DL with, a concrete domain, is similar to the difference between pure LP, on the one hand, and, on the other hand, CLP [7, 15, 44, 45, 77] with a specific domain such as the rationals or the integers. CLP, as already discussed, additionally incorporates the idea of specifying a domain for a variable, or for a pair of variables —i.e., restricting a variable's range to the elements of a unary relation, or a pair of variables' range to the elements of a binary relation. Formally, a concrete domain is defined as follows:

Definition 1 (concrete domain [3]). *A concrete domain $\mathcal{D}$ consists of a pair $(\Delta_{\mathcal{D}}, \Phi_{\mathcal{D}})$, where $\Delta_{\mathcal{D}}$ is a set of (concrete) objects, and $\Phi_{\mathcal{D}}$ is a set of predicates over the objects in $\Delta_{\mathcal{D}}$. Each predicate $P \in \Phi_{\mathcal{D}}$ is associated with an arity n and we have $P \subseteq (\Delta_{\mathcal{D}})^n$.*

Definition 2 (admissibility [3]). *A concrete domain $\mathcal{D}$ is admissible if:*

1. *the set of its predicates is closed under negation and contains a predicate for $\Delta_{\mathcal{D}}$; and*
2. *the satisfiability problem for finite conjunctions of predicates is decidable.*

3.2 The concrete domains $\mathcal{D}_x$, with $x \in \{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$

The set x-*at* of x atoms is the x universal relation, which we also refer to as the $\mathcal{MTALC}(\mathcal{D}_x)$ top predicate $\top^x$: $\top^x =$ x-*at*. The set N_P^x of $\mathcal{MTALC}(\mathcal{D}_x)$ predicate names, which constitutes the set of $\mathcal{MTALC}(\mathcal{D}_x)$ atomic predicates, is the set of x atomic relations: $N_P^x = \{\{r\} : r \in \text{x-}at\}$. (Possibly complex) $\mathcal{MTALC}(\mathcal{D}_x)$ predicates are obtained by considering the closure of N_P^x under the set-theoretic operations of complement with respect to $\top^x$, union and intersection. Formally, if the complement $\top^x \setminus P$ of P with respect to $\top^x$ is represented as $\overline{P}$, then we have the following:

Definition 3 ($\mathcal{MTALC}(\mathcal{D}_x)$ predicates). *The set of $\mathcal{MTALC}(\mathcal{D}_x)$ predicates is the smallest set such that:*

1. *every predicate name $P \in N_P^x$ is an $\mathcal{MTALC}(\mathcal{D}_x)$ predicate; and,*
2. *if P_1 and P_2 are $\mathcal{MTALC}(\mathcal{D}_x)$ predicates, then so are: $\overline{P_1}$, $P_1 \cap P_2$, and $P_1 \cup P_2$.*

Because the x atoms are JEPD, the set of $\mathcal{MTALC}(\mathcal{D}_x)$ predicates reduces to the set $2^{\text{x-}at}$ of all x relations: each $\mathcal{MTALC}(\mathcal{D}_x)$ predicate can be written as $\{P_1, \ldots, P_n\}$, where the P_i's are x atoms. The empty relation, $\emptyset$, represents the bottom predicate, which we also refer to as $\bot_x$.

Remark 1. We could use only the x atoms as $\mathcal{MTALC}(\mathcal{D}_x)$ predicates, since a constraint of the form $\{P_1,\ldots,P_n\}(x_1,\cdots,x_m)$, where m is the arity of the x relations, can be equivalently written as the disjunction $P_1(x_1,\cdots,x_m)\vee\ldots\vee P_n(x_1,\cdots,x_m)$; i.e., as a disjunction of constraints involving only the predicate names. However, the first form is preferred to the second, for at least two reasons:

1. **flexibility:** in the constraint community, and in particular in the constraint-based spatial reasoning community, the first form is preferred to the second -a CSP is nothing else than a conjunction of such constraints.
2. **efficiency of a priori pruning with a constraint engine embedded in the tableaux search space:** the spatial-CSP part of the label of a node of the search space, as already discussed, is nothing else than a conjunction of constraints expressed in the RA x which has given birth to the $\mathcal{MTALC}(\mathcal{D}_x)$ concrete domain —in other words, a CSP expressed in x. If we use the second form, the constraints are instantiated, in a passive generate-and-test manner, with one of the disjuncts before they are submitted to the a priori pruning with the constraint engine -they are submitted as they are, if we use the first form, which gives the pruning process more potential in preventing failures.

Let $x\in\{\mathcal{RCC}8,\mathcal{CDA},\mathcal{CYC}_t\}$. Thanks to the above discussion, the concrete domain generated by x, $\mathcal{D}_x$, can be written as $\mathcal{D}_x=(\Delta_{\mathcal{D}_x},\Phi_{\mathcal{D}_x})$, with:

$$\mathcal{D}_{\mathcal{RCC}8}=(\mathcal{RTS},2^{\mathcal{RCC}8\text{-}at})$$
$$\mathcal{D}_{\mathcal{CDA}}=(2\mathrm{D}\mathcal{P},2^{\mathcal{CDA}\text{-}at})$$
$$\mathcal{D}_{\mathcal{CYC}_t}=(2\mathrm{D}\mathcal{O},2^{\mathcal{CYC}_t\text{-}at})$$

where:

1. $\mathcal{RTS}$ is the set of regions of a topological space $\mathcal{TS}$; 2D$\mathcal{P}$ is the set of 2D points; 2D$\mathcal{O}$ is the set of 2D orientations; and
2. x-at, as we have seen, is the set of x atoms —$2^{\text{x-}at}$ is thus the set of all x relations.

3.3 Admissibility of the concrete domains $\mathcal{D}_x$, with $x\in\{\mathcal{RCC}8,\mathcal{CDA},\mathcal{CYC}_t\}$

Let x be an RA from the set $\{\mathcal{RCC}8,\mathcal{CDA},\mathcal{CYC}_t\}$.

Closure of the set of predicates, $\Phi_{\mathcal{D}_x}=2^{\text{x-}at}$, of the concrete domain $\mathcal{D}_x$ is already implicit in what has been said so far. Given a predicate P of $\mathcal{D}_x$, corresponding to the x relation $\{r_1,\ldots,r_n\}$, its negation, $\overline{P}$, is the complement of $\{r_1,\ldots,r_n\}$ w.r.t. the set x-at, which represents the x universal relation:

$$\overline{P}=\text{x-}at\setminus\{r_1,\ldots,r_n\} \quad (1)$$

The reader should have no difficulty to see that $\overline{P}$ is an element of $\Phi_{\mathcal{D}_x}$.

A unary relation R can always be written as a particular n-ary relation, with $n\geq 2$. For instance, as the relation $\{(x,\ldots,x)\in\{x\}^n|x\in R\}$. The domain $\Delta_{\mathcal{D}_x}$, which is a unary relation, can be written as a particular predicate of $\Phi_{\mathcal{D}_x}$, as follows:

1. as the predicate EQ if $x = \mathcal{RCC}8$;
2. as the predicate Eq if $x = \mathcal{CDA}$; and
3. as the predicate eee if $x = \mathcal{CYC}_t$.

In order to establish admissibility of the concrete domains $\mathcal{D}_x$, it remains to convince the reader of the decidability of the satisfiability problem for finite conjunctions of predicates of $\Phi_{\mathcal{D}_x}$. This derives from (decidability and) tractability of the subset $\{\{r\}|r \in \text{x-}at\}$ of x atomic relations, for each $x \in \{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$:

1. The $\mathcal{RCC}8$ atomic relations have been shown to form a tractable subset of $\mathcal{RCC}8$ by Renz and Nebel [68]. A problem expressed in the subset can be checked for consistency using Allen's constraint propagation algorithm [1].
2. The $\mathcal{CDA}$ atomic relations have been shown to form a tractable subset of $\mathcal{CDA}$ by Ligozat [51]. A problem expressed in the subset can be checked for consistency by applying Allen's constraint propagation algorithm [1] to each of the projections on the axes of an orthogonal system of coordinates, chosen in such a way that the x- and y-axes are, respectively, a west-east horizontal directed line (d-line) and a south-north vertical d-line —each of the projections is a problem expressed in Vilain and Kautz's temporal point algebra [80].
3. Isli and Cohn [42, 43] have provided a propagation algorithm achieving 4-consistency for CSPs expressed in their RA $\mathcal{CYC}_t$, and shown that the propagation is complete for the subset of atomic relations. Indeed, the propagation does even better than just being complete: given a CSP of $\mathcal{CYC}_t$ atomic relations, the algorithm either detects its inconsistency, if it is inconsistent, or transforms it into a CSP which is globally consistent —the property of global consistency, also called strong n-consistency in [26], where n is the size of the input CSP, is computationally important, for it implies that a solution can searched for in a backtack-free manner [26].

The situation is summarised by the following theorem:

Theorem 1. *Let $x \in \{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$. The concrete domain $\mathcal{D}_x$ is admissible.*

Remark 2. The concrete domains $\mathcal{D}_x$, $x \in \{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$, we consider in this work behave better than just being admissible. Solving the consistency problem of a conjunction of constraints expressed in the set of x atomic relations is not only decidable but tractable as well. Specifically, as already explained, such a conjunction can be solved with a path consistency algorithm such as Allen's [1], in case x is binary, and with a 4-consistency algorithm such as Isli and Cohn's [42, 43], in case x is ternary. Freksa's point-based calculus of relative orientation [25, 83], for instance, can generate an admissible concrete domain, for consistency of a conjunction of constraints expressed in the calculus is decidable [74]; however, the calculus does not verify the tractability property above (again, the reader is referred to [74]). The reason for considering only "nicely" admissible concrete domains is that we want to use *CSP* techniques for the solving of a conjunction of x constraints; namely:

1. a solution search algorithm such as Ladkin and Reinefeld's [47], which uses Allen's path-consistency algorithm [1] as the filtering procedure during the search; and
2. a solution search algorithm such as Isli and Cohn's [42, 43], which uses the 4-consistency algorithm in [42, 43] as the filtering procedure during the search.

3.4 Syntax of $\mathcal{MTALC}(\mathcal{D}_x)$ concepts, with $x \in \{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$

Let x be an RA from the set $\{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$. $\mathcal{MTALC}(\mathcal{D}_x)$, as already explained, is obtained from $\mathcal{ALC}(\mathcal{D})$ by temporalisng the roles, and spatialising the concrete domain. The roles in $\mathcal{ALC}$, as well as the roles other than the abstract features in $\mathcal{ALC}(\mathcal{D})$, are interpreted in a similar way as the modal operators of the multi-modal logic $\mathcal{K}_{(m)}$ [33] ($\mathcal{K}_{(m)}$ is a multi-modal version of the minimal normal modal system $\mathcal{K}$), which explains Schild's [71] correspondence between $\mathcal{ALC}$ and $\mathcal{K}_{(m)}$. As in $\mathcal{ALC}(\mathcal{D})$, we will suppose a countably infinite set N_R of role names (or just roles), and a countably infinite subset N_{aF} of N_R whose elements consist of abstract feature names (or just abstract features). Additionally, however, we suppose that the roles (including the abstract features) are antisymmetric and serial —the abstract features are also linear.

Definition 4 **($\mathcal{MTALC}(\mathcal{D}_x)$ concepts).** *Let x be an RA from the set $\{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$. Let N_C, N_R and N_{cF} be mutually disjoint and countably infinite sets of concept names, role names, and concrete features, respectively; and N_{aF} a countably infinite subset of N_R whose elements are abstract features. A (concrete) feature chain is any finite composition $f_1 \dots f_n g$ of $n \geq 0$ abstract features $f_1, \dots, f_n$ and one concrete feature g. The set of $\mathcal{MTALC}(\mathcal{D}_x)$ concepts is the smallest set such that:*

1. *$\top$ and $\bot$ are $\mathcal{MTALC}(\mathcal{D}_x)$ concepts*
2. *an $\mathcal{MTALC}(\mathcal{D}_x)$ concept name is an $\mathcal{MTALC}(\mathcal{D}_x)$ (atomic) concept*
3. *if C and D are $\mathcal{MTALC}(\mathcal{D}_x)$ concepts; R is a role (in general, and an abstract feature in particular); g is a concrete feature; u_1, u_2 and u_3 are feature chains; and P is an $\mathcal{MTALC}(\mathcal{D}_x)$ predicate, then the following expressions are also $\mathcal{MTALC}(\mathcal{D}_x)$ concepts:*
   (a) $\neg C$, $C \sqcap D$, $C \sqcup D$, $\exists R.C$, $\forall R.C$; and
   (b) $\exists(u_1)(u_2).P$ if x binary, $\exists(u_1)(u_2)(u_3).P$ if x ternary.

We denote by $\mathcal{MTALC}$ the sublanguage of $\mathcal{MTALC}(\mathcal{D}_x)$ given by rules 1, 2 and 3a in Definition 4, which is the temporal component of $\mathcal{MTALC}(\mathcal{D}_x)$. It is worth noting that $\mathcal{MTALC}$ does not consist of a mere temporalisation of $\mathcal{ALC}$ [73]. Indeed, $\mathcal{ALC}$ contains only general, not necessarily functional roles, whereas $\mathcal{MTALC}$ contains abstract features as well. As it will become clear shortly, a mere temporalisation of $\mathcal{ALC}$ (i.e., $\mathcal{MTALC}$ without abstract features) cannot capture the expressiveness of two well-known modal temporal logics: Propositional Linear Temporal Logic $\mathcal{PLTL}$, and the $\mathcal{CTL}$ version of the full branching modal temporal logic $\mathcal{CTL}^*$ [19]. Given two integers $p \geq 0$ and $q \geq 0$, the sublanguage of $\mathcal{MTALC}(\mathcal{D}_x)$ (resp. $\mathcal{MTALC}$) whose concepts involve at most p general, not necessarily functional roles, and q abstract features will be referred to as $\mathcal{MTALC}_{p,q}(\mathcal{D}_x)$ (resp. $\mathcal{MTALC}_{p,q}$). We discuss shortly the cases $(p, q) = (0, 0)$, $(p, q) = (0, 1)$, and $(0, q)$ with $q \geq 0$. We first define weakly cyclic TBoxes.

3.5 Weakly cyclic TBoxes

An ($\mathcal{MTALC}(\mathcal{D}_x)$ terminological) axiom is an expression of the form $A \doteq C$, A being a (defined) concept name and C a concept. A TBox is a finite set of axioms, with the condition that no concept name appears more than once as the left hand side of an axiom.

Let T be a TBox. T contains two kinds of concept names: concept names appearing as the left hand side of an axiom of T are defined concepts; the others are primitive concepts. A defined concept A "*directly uses*" a defined concept B iff B appears in the right hand side of the axiom defining A. If "*uses*" is the transitive closure of "*directly uses*" then T contains a cycle iff there is a defined concept A that "*uses*" itself. T is cyclic if it contains a cycle; it is acyclic otherwise. T is weakly cyclic if it satisfies the following two conditions:

1. Whenever A uses B and B uses A, we have $B = A$ —the only possibility for a defined concept to get involved in a cycle is to appear in the right hand side of the axiom defining it.
2. All possible occurrences of a defined concept B in the right hand side of the axiom defining B itself, are within the scope of an existential or a universal quantifier; i.e., in subconcepts of C of the form $\exists R.D$ or $\forall R.D$, C being the right hand side of the axiom, $B \doteq C$, defining B.

We suppose that the defined concepts of a TBox split into *eventuality* defined concepts and *noneventuality* defined concepts.

In the rest of the paper, unless explicitly stated otherwise, we denote concepts reducing to concept names by the letters A and B, possibly complex concepts by the letters C, D, E, general (possibly functional) roles by the letter R, abstract features by the letter f, concrete features by the letters g and h, feature chains by the letter u, (possibly complex) predicates by the letter P.

3.6 $\mathcal{MTALC}_{0,0}(\mathcal{D}_x)$: domain-specific Qualitative Spatial *CLP*

$\mathcal{MTALC}_{0,0}(\mathcal{D}_x)$ involves no roles and no abstract features. What differentiates it from the propositional calculus, is the possibility to refer to spatial variables, thanks to the concrete features, and to "qualitatively" restrict, in the case x binary, for instance, the domains of pairs of such variables, thanks to the predicates of the concrete domain. In other words, $\mathcal{MTALC}_{0,0}(\mathcal{D}_x)$ can also express constraints of the form $\exists(g_1)(g_2).P$, where g_1 and g_2 are concrete features and P is a predicate of the concrete domain (a qualitative spatial relation of the RA x). $\mathcal{MTALC}_{0,0}(\mathcal{D}_x)$ can thus bee seen as domain-specific Qualitative Spatial *CLP* (the case $x = \mathcal{RCC}8$, for instance, corresponds to the specific domain with the set of regions of a topological space, as the variables' domain, and with $\mathcal{RCC}8$ relations as constraints for restricting the variation of pairs of these variables). Item 3 in Definition 4 becomes as follows:

(3) if C and D are concepts; g_1, g_2 and g_3 are concrete features; and P is a predicate, then the following expressions are also concepts:
   (a) $\neg C$, $C \sqcap D$, $C \sqcup D$; and
   (b) $\exists(g_1)(g_2).P$ if x binary, $\exists(g_1)(g_2)(g_3).P$ if x ternary.

3.7 $\mathcal{MTALC}_{0,1}(\mathcal{D}_x)$

$\mathcal{MTALC}_{0,1}(\mathcal{D}_x)$ is the sublanguage of $\mathcal{MTALC}(\mathcal{D}_x)$, with no nonfunctional roles, and one abstract feature which we refer to as f. $\mathcal{MTALC}_{0,1}(\mathcal{D}_x)$ with weakly cyclic

TBoxes subsumes the Propositional Linear Temporal Logic, $\mathcal{PLTL}$ (see, for instance, [19]). The feature chains of $\mathcal{MTALC}_{0,1}(\mathcal{D}_x)$ are of the form $f \ldots fg$ (a finite chain of the f symbol, followed by a concrete feature). Item 3 in Definition 4 becomes as follows:

(3) if C and D are concepts; u_1, u_2 and u_3 are feature chains; and P is a predicate, then the following expressions are also concepts:
   (a) $\neg C$, $C \sqcap D$, $C \sqcup D$, $\exists f.C$, $\forall f.C$; and
   (b) $\exists(u_1)(u_2).P$ if x binary, $\exists(u_1)(u_2)(u_3).P$ if x ternary.

Well-formed formulas (WFFs) of $\mathcal{PLTL}$, over an alphabet $\mathcal{P}$ of atomic propositions, are defined as follows, where $\bigcirc$, $\Box$, $\Diamond$ and U are the standard temporal operators *next*, *necessarily*, *eventually* and *Until*:

1. **true** and **false** are WFFs
2. an atomic proposition is a WFF
3. if ϕ and ψ are WFFs then so are $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\bigcirc\phi$, $\Box\phi$, $\Diamond\phi$ and $\phi U \psi$

A state s of a linear temporal structure $t = < S, \pi, R^* >$ satisfies a $\mathcal{PLTL}$ formula ϕ, denoted by $t, s \models \phi$, is defined inductively as follows:

1. $t, s \models p$ iff $p \in \pi(s)$, for all atomic propositions $p \in \mathcal{P}$
2. $t, s \models \neg\phi$ iff it is not the case that $t, s \models \phi$
3. $t, s \models \phi \wedge \psi$ iff $t, s \models \phi$ and $t, s \models \psi$
4. $t, s \models \phi \vee \psi$ iff $t, s \models \phi$ or $t, s \models \psi$
5. $t, s \models \bigcirc\phi$ iff $t, s' \models \phi$, where s' is the immediate successor of s in t —i.e., s' is such that $f(s, s')$
6. $t, s \models \Box\phi$ iff $t, s' \models \phi$, for all s' such that $f^*(s, s')$
7. $t, s \models \Diamond\phi$ iff $t, s' \models \phi$, for some s' such that $f^*(s, s')$
8. $t, s \models \phi U \psi$ iff for some s' such that $f^*(s, s')$:
   (a) $t, s' \models \psi$; and
   (b) $t, s'' \models \phi$, for all s'' such that $f^*(s, s'')$ and $f^+(s'', s')$

Formulas of the form $\Diamond\phi$ or $\phi U \psi$ are eventuality formulas. A state s of a structure satisfies $\Diamond\phi$ (resp. $\phi U \psi$) iff, there exists a successor state s' of s such that s' satisfies ϕ (resp. s' satisfies ψ and all states between s and s', not necessarily including s', satisfy ϕ). The Boolean operators $\wedge$ and $\vee$ are associated with the operators $\sqcap$ and $\sqcup$, respectively. Each atomic proposition p from $\mathcal{P}$ is associated with a primitive concept A_p. With each $\mathcal{PLTL}$ WFF, ϕ, we associate the $\mathcal{MTALC}_{0,1}(\mathcal{D}_x)$ defined concept B_ϕ, defined inductively as follows:

1. $B_p \doteq A_p$, for all formulas reducing to an atomic proposition p
2. $B_{true} \doteq \top$
3. $B_{false} \doteq \bot$
4. $B_{\neg\phi} \doteq \neg B_\phi$
5. $B_{\phi\wedge\psi} \doteq B_\phi \sqcap B_\psi$
6. $B_{\phi\vee\psi} \doteq B_\phi \sqcup B_\psi$
7. $B_{\bigcirc\phi} \doteq \exists f.B_\phi$
8. $B_{\Box\phi} \doteq B_\phi \sqcap \exists f.B_{\Box\phi}$

9. $B_{\Diamond\phi} \doteq B_\phi \sqcup \exists f.B_{\Diamond\phi}$
10. $B_{\phi U\psi} \doteq B_\psi \sqcup (B_\phi \sqcap \exists f.B_{\phi U\psi})$

Each of the defined concepts $B_{\Box\phi}$, $B_{\Diamond\phi}$ and $B_{\phi U\psi}$ "directly uses" itself. More generally, the procedure is such that, whenever a defined concept B_ϕ "directly uses" a defined concept B_ψ, ψ is either ϕ (B_ψ is then enclosed within the scope of an existential or universal quantifier), or a strict subformula of ϕ. This ensures that the TBox is weakly cyclic.

The axioms defining $B_{\Diamond\phi}$ and $B_{\phi U\psi}$ do not correspond to equivalences. The intuitive reason behind it is that, they may raise the illusion that, for instance, a temporal structure satisfies a concept of the form $B_{\Diamond\phi}$, even if we report indefinitely its satisfiability from the current state of the structure to the next, without satisfying ϕ. Such defined concepts will be referred to as *eventuality* concepts; these will be used in the determination of the accepting states of the weak alternating automaton to be associated with the satisfiability of a concept w.r.t. a weakly cyclic TBox.

3.8 $\mathcal{MTALC}_{0,q}(\mathcal{D}_x)$, with $q \geq 0$

$\mathcal{MTALC}_{0,q}(\mathcal{D}_x)$, with $q \geq 0$, has no nonfunctional role and q abstract features. Item 3 in Definition 4 becomes as follows:

(3) if C and D are concepts; f is an abstract feature; u_1, u_2 and u_3 are feature chains; and P is a predicate, then the following expressions are also concepts:
   (a) $\neg C$, $C \sqcap D$, $C \sqcup D$, $\exists f.C$, $\forall f.C$; and
   (b) $\exists(u_1)(u_2).P$ if x binary, $\exists(u_1)(u_2)(u_3).P$ if x ternary.

We now consider the restricted version, $\mathcal{CTL}$, of the full branching modal temporal logic, $\mathcal{CTL}^*$ [19]. *State* formulas (true or false of states) and *path* formulas (true or false of paths) of $\mathcal{CTL}$, over an alphabet $\mathcal{P}$ of atomic propositions, are defined by rules S1-S2-S3-S4-P0 below, where the symbols $\mathcal{A}$ and $\mathcal{E}$ denote, respectively, the path quantifiers "for all futures" (along all paths) and "for some future" (along some path):

S1 **true** and **false** are state formulas
S2 an atomic proposition is a state formula
S3 if ϕ and ψ are state formulas then so are $\neg\phi$, $\phi \wedge \psi$ and $\phi \vee \psi$
S4 if ϕ is a path formula then $\mathcal{A}\phi$ and $\mathcal{E}\phi$ are state formulas
P0 if ϕ and ψ are state formulas then $\bigcirc\phi$, $\Box\phi$, $\Diamond\phi$ and $\phi U\psi$ are path formulas

The language of $\mathcal{CTL}$, i.e., the set of well-formed formulas (WFFs) of $\mathcal{CTL}$, is the set of all $\mathcal{CTL}$ state formulas. Given a branching temporal structure $t =< S, \pi, R^* >$, a full path of t is an infinite sequence $s_0, s_1, s_2, \ldots$ such that, for all $i \geq 0$, $R(s_i, s_{i+1})$. As in [19], we use the convention that $x = (s_0, s_1, s_2, \ldots)$ denotes a full path, and that x^i denotes the suffix path $(s_i, s_{i+1}, \ldots)$. We denote by $t, s \models \phi$ (resp. $t, x \models \phi$) the fact that state formula (resp. path formula) ϕ is true in structure t at state s_0 (resp. of path x). $t, s \models \phi$ and $t, x \models \phi$ are defined inductively as follows:

S1a $t, s \models$ **true**

S1b $t, s \not\models$ **false**
S2a $t, s \models p$ iff $p \in \pi(s)$, for all atomic propositions $p \in \mathcal{P}$
S3a $t, s \models \neg\phi$ iff $t, s \not\models \phi$
S3b $t, s \models \phi \wedge \psi$ iff $t, s \models \phi$ and $t, s \models \psi$
S3c $t, s \models \phi \vee \psi$ iff $t, s \models \phi$ or $t, s \models \psi$
S4a1 $t, s \models \mathcal{A} \bigcirc \phi$ iff for all s' such that $R(s, s')$, $t, s' \models \phi$
S4a2 $t, s \models \mathcal{A}\Box\phi$ iff $t, s \models \phi$ and, for all s' such that $R(s, s')$, $t, s' \models \mathcal{A}\Box\phi$
S4a3 $t, s \models \mathcal{A}\Diamond\phi$ iff $t, s \models \phi$ or, for all s' such that $R(s, s')$, $t, s' \models \mathcal{A}\Diamond\phi$
S4a4 $t, s \models \mathcal{A}(\phi U\psi)$ iff $t, s \models \psi$; or $t, s \models \phi$ and, for all s' such that $R(s, s')$, $t, s \models \mathcal{A}(\phi U\psi)$
S4b1 $t, s \models \mathcal{E} \bigcirc \phi$ iff for some s' such that $R(s, s')$, $t, s' \models \phi$
S4b2 $t, s \models \mathcal{E}\Box\phi$ iff $t, s \models \phi$ and, for some s' such that $R(s, s')$, $t, s' \models \mathcal{E}\Box\phi$
S4b3 $t, s \models \mathcal{E}\Diamond\phi$ iff $t, s \models \phi$ or, for some s' such that $R(s, s')$, $t, s' \models \mathcal{E}\Diamond\phi$
S4b4 $t, s \models \mathcal{E}(\phi U\psi)$ iff $t, s \models \psi$; or $t, s \models \phi$ and, for some s' such that $R(s, s')$, $t, s' \models \mathcal{E}(\phi U\psi)$

It is worth noting that $\mathcal{CTL}$ WFFs prefixed by a quantifier are of the form $\mathcal{A}\chi$ or $\mathcal{E}\chi$, where χ is a path formula of the form $\bigcirc\phi$, $\Box\phi$, $\Diamond\phi$ or $\phi U\psi$, ϕ and ψ being state formulas.

Similarly to the case of $\mathcal{PLTL}$, this leads us to the following. The Boolean operators $\wedge$ and $\vee$ are associated with the operators $\sqcap$ and $\sqcup$, respectively. Each atomic proposition p from $\mathcal{P}$ is associated with a primitive concept A_p. With each $\mathcal{CTL}$ WFF, ϕ, we associate the $\mathcal{MTALC}_{0,q}$ defined concept B_ϕ, defined recursively by the steps below. Initially, there is no abstract feature, and no general role. The abstract features are created by the procedure as needed. We make use of a general role R which we initialise to the empty role. Whenever a fresh abstract feature, say f, is created, it is added to R. So doing, the general role R, by the time the procedure will have completed, will be the union of all the abstract features in the TBox created for the input formula. If the created abstract features are $f_1, \ldots, f_n$, then $R = f_1 \cup \ldots \cup f_n$. If C is a concept, then $\exists R.C$ is synonymous with $\exists f_1.C \sqcup \ldots \sqcup \exists f_n.C$, and $\forall R.C$ with $\forall f_1.C \sqcap \ldots \sqcap \forall f_n.C$:

1. $B_p \doteq A_p$, for all formulas consisting of an atomic proposition p
2. $B_{true} \doteq \top$
3. $B_{false} \doteq \bot$
4. $B_{\neg\phi} \doteq \neg B_\phi$
5. $B_{\phi\wedge\psi} \doteq B_\phi \sqcap B_\psi$
6. $B_{\phi\vee\psi} \doteq B_\phi \sqcup B_\psi$
7. $B_{\mathcal{A}\bigcirc\phi} \doteq \forall R.B_\phi$
8. $B_{\mathcal{A}\Box\phi} \doteq B_\phi \sqcap \forall R.B_{\mathcal{A}\Box\phi}$
9. $B_{\mathcal{A}\Diamond\phi} \doteq B_\phi \sqcup \forall R.B_{\mathcal{A}\Diamond\phi}$
10. $B_{\mathcal{A}(\phi U\psi)} \doteq B_\psi \sqcup (B_\phi \sqcap \forall R.B_{\mathcal{A}(\phi U\psi)})$
11. $B_{\mathcal{E}\bigcirc\phi} \doteq \exists f.B_\phi$, where f is a fresh abstract feature which we add to R $(R \leftarrow R \cup f)$
12. $B_{\mathcal{E}\Box\phi} \doteq B_\phi \sqcap \exists f.B_{\mathcal{E}\Box\phi}$, where f is a fresh abstract feature $(R \leftarrow R \cup f)$
13. $B_{\mathcal{E}\Diamond\phi} \doteq B_\phi \sqcup \exists f.B_{\mathcal{E}\Diamond\phi}$, where f is a fresh abstract feature $(R \leftarrow R \cup f)$
14. $B_{\mathcal{E}(\phi U\psi)} \doteq B_\psi \sqcup (B_\phi \sqcap \exists f.B_{\mathcal{E}(\phi U\psi)})$, where f is a fresh abstract feature $(R \leftarrow R \cup f)$

The decreasing property explained for the case $\mathcal{MTALC}_{0,1}(\mathcal{D}_x)$ ensures that, given an input formula, the procedure outputs a TBox which is weakly cyclic.

We define the set of subformulas of a formula ϕ, *Subf*(ϕ), inductively in the following obvious way:

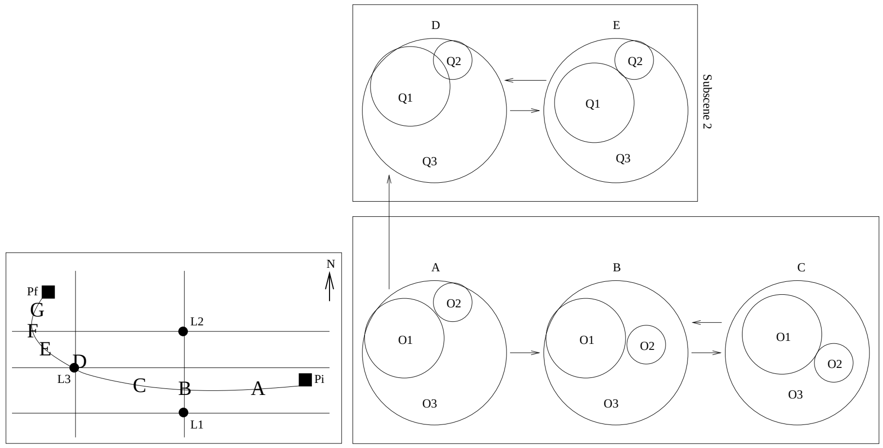

Fig. 2. (Left) Illustration of $\mathcal{MTALC}_{0,1}(\mathcal{D}_{\mathcal{CDA}})$: the upward arrow pointing at N indicates North. (Right) Illustration of $\mathcal{MTALC}_{0,2}(\mathcal{D}_{\mathcal{RCC}8})$.

1. $Subf(p) = \{p\}$, for all formulas consisting of an atomic proposition p
2. $Subf(true) = \{true\}$
3. $Subf(false) = \{false\}$
4. $Subf(\neg\phi) = \{\neg\phi\} \cup Subf(\phi)$
5. $Subf(\phi \wedge \psi) = \{\phi \wedge \psi\} \cup Subf(\phi) \cup Subf(\psi)$
6. $Subf(\phi \vee \psi) = \{\phi \vee \psi\} \cup Subf(\phi) \cup Subf(\psi)$
7. $Subf(\mathcal{A} \bigcirc \phi) = \{\mathcal{A} \bigcirc \phi\} \cup Subf(\phi)$
8. $Subf(\mathcal{A}\Box\phi) = \{\mathcal{A}\Box\phi\} \cup Subf(\phi)$
9. $Subf(\mathcal{A}\Diamond\phi) = \{\mathcal{A}\Diamond\phi\} \cup Subf(\phi)$
10. $Subf(\mathcal{A}(\phi U \psi)) = \{\mathcal{A}(\phi U \psi\} \cup Subf(\psi) \cup Subf(\phi)$
11. $Subf(\mathcal{E} \bigcirc \phi) = \{\mathcal{E} \bigcirc \phi\} \cup Subf(\phi)$
12. $Subf(\mathcal{E}\Box\phi) = \{\mathcal{E}\Box\phi\} \cup Subf(\phi)$
13. $Subf(\mathcal{E}\Diamond\phi) = \{\mathcal{E}\Diamond\phi\} \cup Subf(\phi)$

Given a formula ϕ, the defined concept B_ϕ associated with ϕ by the procedure described above, is so that all defined concepts $B_{\phi_1}, \ldots, B_{\phi_n}$ which B_ϕ "directly uses", and different from B_ϕ itself, verify the decreasing property $\text{size}(\phi_1) + \ldots + \text{size}(\phi_n) < \text{size}(\phi)$, where $\text{size}(\psi)$, for a formula ψ, is the size of ψ in terms of number of symbols. This ensures that the number of defined concepts in the TBox associated with a formula ϕ is linear, and bounded by $\text{size}(\phi)$.

It is important to note that, given the fact that formulas of the form $\mathcal{A}\Diamond\phi$, $\mathcal{A}(\phi U \psi)$, $\mathcal{E}\Diamond\phi$ or $\mathcal{E}(\phi U \psi)$ are eventualities, the defined concepts of the form $B_{\mathcal{A}\Diamond\phi}$, $B_{\mathcal{A}(\phi U \psi)}$, $B_{\mathcal{E}\Diamond\phi}$ or $B_{\mathcal{E}(\phi U \psi)}$, created by the procedure above, should be marked as eventuality defined concepts.

Before giving the formal semantics of $\mathcal{MTALC}(\mathcal{D}_x)$, we provide some examples.

4 Examples

We now provide illustrating examples. Each of Examples 1 and 4 uses an acyclic TBox, which includes feature chains other than concrete features; whereas each of

Examples 2 and 3 represents a non-terminating physical system, and uses a weakly cyclic TBox. The TBox of Example 3 uses an eventuality concept.

Example 1 (illustration of $\mathcal{MTALC}_{0,1}(\mathcal{D}_{\mathcal{CDA}})$). Consider a satellite-like high-level surveillance system, aimed at the surveillance of flying aeroplanes within a three-landmark environment. The basic task of the system is to situate qualitatively an aeroplane relative to the different landmarks, as well as to relate qualitatively the different positions of an aeroplane while in flight. If the system is used for the surveillance of the European sky, the landmarks could be capitals of European countries, such as Berlin, London and Paris. For the purpose, the system uses a high-level spatial description language, such as a QSR language, which we suppose in this example to be the Cardinal Directions Algebra $\mathcal{CDA}$ [23]. The example is illustrated in Figure 2(Left). The horizontal and vertical lines through the three landmarks partition the plane into 0-, 1- and 2-dimensional regions, as shown in Figure 2(Left). The flight of an aeroplane within the environment, as tracked by the surveillance system, starts from some point P_i in Region A (initial region), and ends at some point P_f in Region G (final, or goal region). Immediately after the initial region, the flight "moves" to Region B, then to Region C, ..., then to Region F, and finally to the goal region G. The tracking of the system consists of qualitative knowledge on how it "sees" the aeroplane at each moment of the flight being tracked —within the same region, the knowledge is constant. The tracking consists thus of recording successive snapshots of the flight, one per region. A snapshot is a conjunction of constraints giving the $\mathcal{CDA}$ relation relating the aeroplane to each of the three landmarks, situating thus the aeroplane at the corresponding moment. The entire flight consists of a succession of subflights, $f_A, f_B, \ldots, f_G$, such that f_B immediately follows f_A, f_C immediately follows f_B, ..., and f_G immediately follows f_F. Subflight f_X, $X \in \{A, \ldots, G\}$, takes place in Region X, and gives rise to a defined concept B_X describing the panorama of the aeroplane O while in Region X, and saying which subflight takes place next, i.e., which Region is flied over next. We make use of the concrete features g_{l1}, g_{l2}, g_{l3} and g_o, which have the task of "referring", respectively, to the actual positions of landmarks l_1, l_2, l_3, and of the aeroplane O. As roles, only one abstract feature is needed, which we refer to as f, which is the linear-time immediate successor function. The acyclic TBox composed of the following axioms describes the flight:

$$
\begin{aligned}
B_A &\doteq \exists(g_o)(g_{l1}).NE \sqcap \exists(g_o)(g_{l2}).SE \sqcap \exists(g_o)(g_{l3}).SE \sqcap \exists f.B_B \\
B_B &\doteq \exists(g_o)(g_{l1}).No \sqcap \exists(g_o)(g_{l2}).So \sqcap \exists(g_o)(g_{l3}).SE \sqcap \exists f.B_C \\
B_C &\doteq \exists(g_o)(g_{l1}).NW \sqcap \exists(g_o)(g_{l2}).SW \sqcap \exists(g_o)(g_{l3}).SE \sqcap \exists f.B_D \\
B_D &\doteq \exists(g_o)(g_{l1}).NW \sqcap \exists(g_o)(g_{l2}).SW \sqcap \exists(g_o)(g_{l3}).Eq \sqcap \exists f.B_E \\
B_E &\doteq \exists(g_o)(g_{l1}).NW \sqcap \exists(g_o)(g_{l2}).SW \sqcap \exists(g_o)(g_{l3}).NW \sqcap \exists f.B_F \\
B_F &\doteq \exists(g_o)(g_{l1}).NW \sqcap \exists(g_o)(g_{l2}).We \sqcap \exists(g_o)(g_{l3}).NW \sqcap \exists f.B_G \\
B_G &\doteq \exists(g_o)(g_{l1}).NW \sqcap \exists(g_o)(g_{l2}).NW \sqcap \exists(g_o)(g_{l3}).NW
\end{aligned}
$$

The concept B_A, for instance, describes the snapshot of the plane while in Region A. It says that the aeroplane is northeast landmark L_1 ($\exists(g_o)(g_{l1}).NE$); southeast landmark L_2 ($\exists(g_o)(g_{l2}).SE$); and southeast landmark L_3 ($\exists(g_o)(g_{l3}).SE$). The concept also says that the subflight to take place next is f_B ($\exists f.B_B$).

One might want as well the system to track how the aeroplane's different positions during the flight relate to each other. For example, that the aeroplane, while in region C, remains northwest of its position while in region B; or, that the position, while in the goal region G, remains northwest of the position while in region E. These two constraints can be injected into the TBox by modifying the axioms B_B and B_E as follows:

$$B_B \doteq \exists(g_o)(g_{l1}).No \sqcap \exists(g_o)(g_{l2}).So \sqcap \exists(g_o)(g_{l3}).SE \sqcap \exists(g_o)(fg_o).SE \sqcap \exists f.B_C$$
$$B_E \doteq \exists(g_o)(g_{l1}).NW \sqcap \exists(g_o)(g_{l2}).SW \sqcap \exists(g_o)(g_{l3}).NW \sqcap \exists(g_o)(ffg_o).SE \sqcap \exists f.B_F$$

Example 2 (illustration of $\mathcal{MTALC}_{0,2}(\mathcal{D}_{\mathcal{RCC}8})$). Consider the moving spatial scene depicted in Figure 2(Right), consisting of two subscenes: a subscene 1, composed of three objects o1, o2 and o3; and a subscene 2, also composed of three objects, q1, q2 and q3:

1. For subscene 1, three snapshots of three submotions are presented, and labelled A, B and C; the arrows show the transitions from the current submotion to the next. The motion is cyclic. It starts with the configurtion A, with o1 touching o2 and tangential proper part of o3, and o2 tangential proper part of o3. The scene's configuration then "moves" to configuration B, involving the change of the $\mathcal{RCC}8$ relation on the pair (o2,03) from TPP to its conceptual neighbour $NTPP$. The next submotion is given by configuration C, involving the object o1 to move completely inside o3, becoming thus $NTPP$ to it. From C, the motion "moves" back to the submotion B, and repeats the submotions B and C in a non-terminating loop.
2. For subscene 2, two snapshots of two submotions are presented and labelled D and E; the arrows show the transitions from the current submotion to the next. The motion is cyclic. It starts with the configurtion D, with q1 partially overlapping q2 and tangential proper part of q3, and q2 tangential proper part of q3. The scene's configuration then "moves" to configuration E, involving the change of the $\mathcal{RCC}8$ relation on the pair (q1,q2) from PO to its conceptual neighbour EC, as well as the change of the $\mathcal{RCC}8$ relation on the pair (q1,q3) from TPP to its conceptual neighbour $NTPP$. From E, the motion "moves" back to D, and repeats the steps in a non-terminating loop.
3. The scene's motion starts with submotion A. The immediate successors of submotion A are submotions B and D, in an incomparable order (the branching from A to B and D is thus an and-branching, and not an or-branching). We make use of the concrete features g_1, g_2 and g_3 to refer to the actual regions corresponding to objects o1, o2 and o3 in Subscene 1, and of the concrete features h_1, h_2 and h_3 to refer to the actual regions corresponding to objects q1, q2 and q3 in Subscene 2.

We make use of two abstract features, f_1 for the infinite path recording Subscene 1 and starting at A, and f_2 for the infinite path recording Subscene 2 and also starting at A. The weakly cyclic TBox composed of the following axioms represents the described moving spatial scene:

$$\begin{aligned}
B_i &\doteq B_A \sqcap \exists f_1.B_{BC} \sqcap \exists f_2.B_{DE} \\
B_{BC} &\doteq B_B \sqcap \exists f_1.(B_C \sqcap \exists f_1.B_{BC}) \\
B_{DE} &\doteq B_D \sqcap \exists f_2.(B_E \sqcap \exists f_2.B_{DE}) \\
B_A &\doteq \exists(g_1)(g_2).EC \sqcap \exists(g_1)(g_3).TPP \sqcap \exists(g_2)(g_3).TPP \\
B_B &\doteq \exists(g_1)(g_2).EC \sqcap \exists(g_1)(g_3).TPP \sqcap \exists(g_2)(g_3).NTPP \\
B_C &\doteq \exists(g_1)(g_2).EC \sqcap \exists(g_1)(g_3).NTPP \sqcap \exists(g_2)(g_3).NTPP \\
B_D &\doteq \exists(h_1)(h_2).PO \sqcap \exists(h_1)(h_3).TPP \sqcap \exists(h_2)(h_3).TPP \\
B_E &\doteq \exists(h_1)(h_2).EC \sqcap \exists(h_1)(h_3).NTPP \sqcap \exists(h_2)(h_3).TPP
\end{aligned}$$

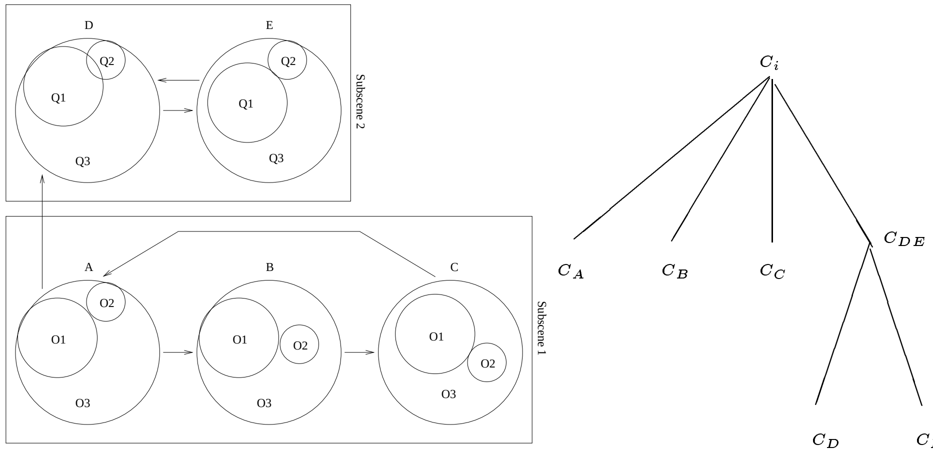

Fig. 3. (Left) Illustration of $\mathcal{MTALC}_{0,1}(\mathcal{D}_{\mathcal{RCC}8})$; (Right) The partial order on the defined concepts of the TBox in Example 3.

The defined concepts B_A, B_B, B_C (resp. B_D, B_E) describe the snapshot of Subscene 1 (resp. Subscene 2) during Submotions A, B, C (resp. D, E). The concept B_A, for instance, says that o1 and o2 are related by the EC relation ($\exists(g_1)(g_2).EC$); that o1 and o3 are related by the TPP relation ($\exists(g_1)(g_3).TPP$); and that o2 and o3 are also related by the TPP relation ($\exists(g_2)(g_3).TPP$). The concept B_{BC} describes the cyclic part of Subscene 1, consisting in repeating indefinitely Submotions B and C. Similarly, the concept B_{DE} describes the cyclic part of Subscene 2, consisting in repeating indefinitely Submotions D and E. The defined concept B_i describes the initial state of the physical system, which starts with Submotion A and then "moves" to the cyclic submotion B_{BC} along the path f_1, and to the other cyclic submotion, B_{DE}, along the path f_2.

Example 3 (illustration of $\mathcal{MTALC}_{0,1}(\mathcal{D}_{\mathcal{RCC}8})$). Consider a physical system similar to the one of the previous example, except that (see Figure 3(Left)):

1. in Subscene 1, from Configuration C, the motion "moves" back, not to configuration B, but to the very first configuration, A; and
2. the branching from the initial configuration A to the two immediate successors, B and D, is not an and-branching, rather an or-branching: from A, the system nondeterministically chooses configuration B or configuration D as the next configuration.

We suppose that the configuration of Subscene 2 is reachable, in the sense that the system will at some point enter configuration D, and then go forever in the repeating of Subscene 2. The system can thus be seen as "repeat Subscene 1 until Subscene 2 is reached". We make use of one abstract feature, which we denote by f. The defined concepts B_A, B_B, B_C, B_D and B_E remain the same as in the previous example. The weakly cyclic TBox composed of the following axioms represents the described moving spatial scene:

$$B_i \doteq B_A \sqcap \exists f.(B_B \sqcap \exists f.(B_C \sqcap \exists f.B_i) \sqcup B_{DE})$$

$$B_{DE} \doteq B_D \sqcap \exists f.(B_E \sqcap \exists f.B_{DE})$$
$$B_A \doteq \exists(g_1)(g_2).EC \sqcap \exists(g_1)(g_3).TPP \sqcap \exists(g_2)(g_3).TPP$$
$$B_B \doteq \exists(g_1)(g_2).EC \sqcap \exists(g_1)(g_3).TPP \sqcap \exists(g_2)(g_3).NTPP$$
$$B_C \doteq \exists(g_1)(g_2).EC \sqcap \exists(g_1)(g_3).NTPP \sqcap \exists(g_2)(g_3).NTPP$$
$$B_D \doteq \exists(h_1)(h_2).PO \sqcap \exists(h_1)(h_3).TPP \sqcap \exists(h_2)(h_3).TPP$$
$$B_E \doteq \exists(h_1)(h_2).EC \sqcap \exists(h_1)(h_3).NTPP \sqcap \exists(h_2)(h_3).TPP$$

A (reflexive and transitive) partial order, $\geq$, on the defined concepts in the above TBox can be defined, which verifies $B_i \geq B_A$, $B_i \geq B_B$, $B_i \geq B_C$, $B_i \geq B_{DE}$, $B_{DE} \geq B_D$ and $B_{DE} \geq B_E$ (see Figure 3(Right)). The TBox verifies the property that, given any two defined concepts, C and D, if C "uses" D then $C \geq D$. The TBox is thus weakly cyclic.

The concept B_i describes the initial state of the physical system, which either performs the submotion of Subscene 1 before repeating itself, or skips to Subscene 2 which it repeats indefinitely. Again, because we want Subscene 2 to be reachable, the concept B_i describes an eventuality, and should be marked as an eventuality concept -this allows rejecting those potential models which repeat indefinitely Subscene 1 without reaching Subscene 2.

Example 4 (illustration of $\mathcal{MTALC}_{0,1}(\mathcal{D}_{\mathcal{CYC}_t})$). We consider an environment with four landmarks, L1, L2, L3 and L4, as depicted in Figure 4(Left). The lines through the different pairs of landmarks partition the plane into a tessellation of two-, one- and zero-dimensional convex regions. Nine of these regions are numbered R1, ..., R9 in Figure 4(Left). A robot R has to navigate all the way through from some point Pi in Region R1 to some point Pf in Rgion R9, traversing in between Regions R2, ..., R8, in that order. With each region Ri, $i = 1 \dots 9$, we associate a concept B_i describing the panorama of the robot while in Region Ri, and giving the region the robot will be in next. We make use of four concrete features $g_1, \dots, g_4$, which "perceive" at each time instant the orientations o_1, o_2, o_3 and o_4 of the directed lines joining the robot to Landmarks L1, L2, L3 and L4, respectively —Figure 4(Right); and of one abstract feature f representing the linear time immediate-successor function. The panorama of the robot at a specific time point consists in the conjunction of $\mathcal{CYC}_t$ constraints associating with each triple of the four orientations the $\mathcal{CYC}_t$ relation it satisfies. Within the same region, the panorama is constant. The navigation of the robot can thus be seen as a chronological evolution of the changing panorama. The TBox with the following axioms provides a plan describing a path the robot has to follow to reach the goal.

$$B_1 \doteq \exists(g_1)(g_2)(g_3).rrr \sqcap \exists(g_1)(g_2)(g_4).rrr \sqcap \exists(g_1)(g_3)(g_4).rrr \sqcap \exists(g_2)(g_3)(g_4).rrr \sqcap \exists f.B_2$$
$$B_2 \doteq \exists(g_1)(g_2)(g_3).rrr \sqcap \exists(g_1)(g_2)(g_4).rro \sqcap \exists(g_1)(g_3)(g_4).rro \sqcap \exists(g_2)(g_3)(g_4).rrr \sqcap \exists f.B_3$$
$$B_3 \doteq \exists(g_1)(g_2)(g_3).rrr \sqcap \exists(g_1)(g_2)(g_4).rrl \sqcap \exists(g_1)(g_3)(g_4).rrl \sqcap \exists(g_2)(g_3)(g_4).rrr \sqcap \exists f.B_4$$
$$B_4 \doteq \exists(g_1)(g_2)(g_3).rro \sqcap \exists(g_1)(g_2)(g_4).rol \sqcap \exists(g_1)(g_3)(g_4).orl \sqcap \exists(g_2)(g_3)(g_4).rro \sqcap \exists f.B_5$$
$$B_5 \doteq \exists(g_1)(g_2)(g_3).rrl \sqcap \exists(g_1)(g_2)(g_4).rll \sqcap \exists(g_1)(g_3)(g_4).lrl \sqcap \exists(g_2)(g_3)(g_4).rrl \sqcap \exists f.B_6$$
$$B_6 \doteq \exists(g_1)(g_2)(g_3).rol \sqcap \exists(g_1)(g_2)(g_4).rll \sqcap \exists(g_1)(g_3)(g_4).lrl \sqcap \exists(g_2)(g_3)(g_4).orl \sqcap \exists f.B_7$$
$$B_7 \doteq \exists(g_1)(g_2)(g_3).rll \sqcap \exists(g_1)(g_2)(g_4).rll \sqcap \exists(g_1)(g_3)(g_4).lrl \sqcap \exists(g_2)(g_3)(g_4).lrl \sqcap \exists f.B_8$$
$$B_8 \doteq \exists(g_1)(g_2)(g_3).rll \sqcap \exists(g_1)(g_2)(g_4).rll \sqcap \exists(g_1)(g_3)(g_4).lel \sqcap \exists(g_2)(g_3)(g_4).lel \sqcap \exists f.B_9$$
$$B_9 \doteq \exists(g_1)(g_2)(g_3).rll \sqcap \exists(g_1)(g_2)(g_4).rll \sqcap \exists(g_1)(g_3)(g_4).lll \sqcap \exists(g_2)(g_3)(g_4).lll$$

The defined concept B_4, for instance, provides the information that the orientations $o_1, \dots, o_4$ should satisfy the constraints that the $\mathcal{CYC}_t$ relation on the triple (o_1, o_2, o_3)

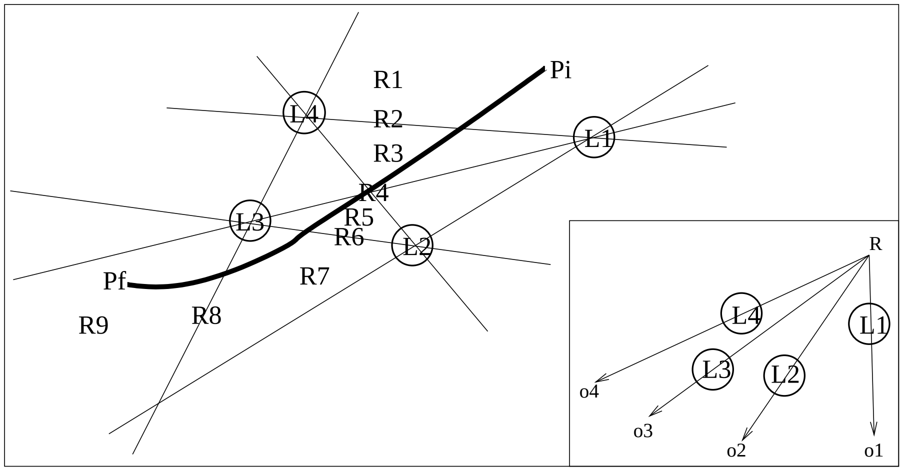

Fig. 4. Illustration of $\mathcal{MTALC}_{0,1}(\mathcal{D}_{\mathcal{CYC}_t})$.

is *rro*, the one on the triple (o_1, o_2, o_4) is *rol*, the one on the triple (o_1, o_3, o_4) is *orl*, and the one on the triple (o_2, o_3, o_4) is *rro* —which is a description of the panorama of the robot while in Region R_4. Concept B_4 also tells which submotion should take place next $(\exists f.B_5)$.

As with Example 1, we can use feature chains of length greater than one (i.e., not reducing to concrete features) to relate, for instance, the value of the line joining the robot to Landmark L_3 while the robot is in Region R_1, to the value of the same line while the robot will be in Region R_9. We might want to constrain the motion of the robot, so that it does not expand beyond the part of Region R_9 which, from Region R_1, appears to the robot's visual system to be to the left hand side of Landmark L_3. The reason for forcing such a constraint could be that, from Region R_1, the part of Region R_9 within the right hand side of Landmark R_3 is hidden to the robot's vision system, which makes its reaching a potential danger. This can be done by forcing the value of orientation 0_3 while the robot is in Region R_1, to be to the left of the value of the same orientation while the robot is in Region R_9. This constraint can be injected into the TBox by modifying the concept B_1 as follows, where $f^8 g_3$ stands for the feature chain $ffffffffg_3$:

$$B_1 \doteq \exists(g_1)(g_2)(g_3).rrr \sqcap \exists(g_1)(g_2)(g_4).rrr \sqcap \exists(g_1)(g_3)(g_4).rrr \sqcap \exists(g_2)(g_3)(g_4).rrr \sqcap \exists(g_3)(g_3)(f^8 g_3).err \sqcap \exists f.B_2$$

Example 5 (another illustration of $\mathcal{MTALC}(\mathcal{D}_{\mathcal{RCC}8})$). In [10], the authors describe a system answering queries on the RCC-8 [67] relation between two input (polygonal) regions of a (quantitative) geographic database. The system also includes the computation of the qualitative abstraction of a quantitative geographic database, which is done by transforming the quantitative database into a qualitative one, which records the $\mathcal{RCC}8$ relation on each pair of the regions in the quantitative database. The importance of the system is obvious: most applications querying geographic databases only need the query-answering system to provide them with the topological relation on pairs of regions in the database. If we think of the database as representing the World's geographic map, then the queries could be of the form "Is Hamburg a German city?", "What are the Mediterranean countries of Africa?", "Are France and Germany neighbouring countries?", or "Does the Sahara Desert just partially overlap, or is it part of, Algeria?". Because computing such a relation directly from a quantitative database is time-consuming, it is worth, especially in situations of repetition of such queries, to compute once and for all the topological relation between every pair of regions in the database, and to store them in a qualitative database; the next time a similar topological query reaches the system, the latter would then only need to access (in constant time) the qualitative database, and to retrieve the relation from there. Because of phenomena such as erosion and (unfortunately) wars, the boundaries of the regions in a geographic database may change with time. It should be clear that $\mathcal{MTALC}(\mathcal{D}_{\mathcal{RCC}8})$ can be used to represent the history of the qualitative abstraction of such a geographic database.

5 Semantics of $\mathcal{MTALC}(\mathcal{D}_x)$, with $x \in \{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$

Definition 5 (k-ary Σ-tree). *Let Σ and $K = \{d_1, \ldots, d_k\}$, $k \geq 1$, be two disjoint alphabets: Σ is a labelling alphabet and K an alphabet of directions. A (full) k-ary tree is an infinite tree whose nodes $\alpha \in K^*$ have exactly k immediate successors each, $\alpha d_1, \ldots, \alpha d_k$. A Σ-tree is a tree whose nodes are labelled with elements of Σ. A (full) k-ary Σ-tree is a k-ary tree t which is also a Σ-tree, which we consider as a mapping $t : K^* \rightarrow \Sigma$ associating with each node $\alpha \in K^*$ an element $t(\alpha) \in \Sigma$. The empty word, ϵ, denotes the root of t. Given a node $\alpha \in K^*$ and a direction $d \in K$, the concatenation of α and d, αd, denotes the d-successor of α. The level $|\alpha|$ of a node α is the length of α as a word. We can thus think of the edges of t as being labelled with directions from K, and of the nodes of t as being labelled with letters from Σ. A partial k-ary Σ-tree (over the set K of directions) is a Σ-tree with the property that a node may not have a d-successor for each direction d; in other terms, a partial k-ary Σ-tree is a Σ-tree which is a prefix-closed[4] partial function $t : K^* \rightarrow \Sigma$.*

$\mathcal{MTALC}(\mathcal{D}_x)$ is equipped with a Tarski-style, possible worlds semantics. $\mathcal{MTALC}(\mathcal{D}_x)$ interpretations are spatio-temporal structures consisting of k-ary trees t, representing k-immediate-successor branching time, together with an interpretation function associating with each primitive concept A the nodes of t at which A is true, and, additionally, associating with each concrete feature g and each node u of t, the value at u (seen as a time instant) of the spatial concrete object referred to by g. Formally:

Definition 6 (interpretation). *Let $x \in \{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$ and $K = \{d_1, \ldots, d_k\}$ a set of k directions. An interpretation $\mathcal{I}$ of $\mathcal{MTALC}(\mathcal{D}_x)$ consists of a pair $\mathcal{I} = (t_{\mathcal{I}}, .^{\mathcal{I}})$, where $t_{\mathcal{I}}$ is a k-ary tree and $.^{\mathcal{I}}$ is an interpretation function mapping each primitive concept A to a subset $A^{\mathcal{I}}$ of K^*, each role R to a subset $R^{\mathcal{I}}$ of $\{(u, ud) \in K^* \times K^* : d \in K\}$, so that $R^{\mathcal{I}}$ is functional if R is an an abstract feature, and each concrete feature g to a total function $g^{\mathcal{I}}$:*

1. *from K^* onto the set $\mathcal{RTS}$ of regions of a topological space $\mathcal{TS}$, if $x = \mathcal{RCC}8$;*
2. *from K^* onto the set $2\mathcal{DP}$ of points of the 2-dimensional space, if $x = \mathcal{CDA}$; and*
3. *from K^* onto the set $2\mathcal{DO}$ of orientations of the 2-dimensional space, if $x = \mathcal{CYC}_t$.*

[4] t is prefix-closed if, for all nodes α, if t is defined for α then it defined for all nodes α' consisting of prefixes of α.

Given an $\mathcal{MTALC}(\mathcal{D}_x)$ interpretation $\mathcal{I} = (t_{\mathcal{I}}, .^{\mathcal{I}})$, a feature chain $u = f_1 \dots f_n g$, and a node v_1, we denote by $u^{\mathcal{I}}(v_1)$ the value $g^{\mathcal{I}}(v_2)$, where v_2 is the $f_1^{\mathcal{I}} \dots f_n^{\mathcal{I}}$-successor of v_1; i.e., v_2 is so that there exists a sequence $v_1 = w_0, w_1, \dots, w_n = v_2$ verifying $(w_i, w_{i+1}) \in f_{i+1}^{\mathcal{I}}$, for all $i \in \{1, \dots, n-1\}$.

Definition 7 (satisfiability w.r.t. a TBox). *Let $x \in \{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$ be a spatial RA, $K = \{d_1, \dots, d_k\}$ a set of k directions, C an $\mathcal{MTALC}(\mathcal{D}_x)$ concept, $\mathcal{T}$ an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox, and $\mathcal{I} = (t_{\mathcal{I}}, .^{\mathcal{I}})$ an $\mathcal{MTALC}(\mathcal{D}_x)$ interpretation. The satisfiability, by a node s of $t_{\mathcal{I}}$, of C w.r.t. to $\mathcal{T}$, denoted $\mathcal{I}, s \models \langle C, \mathcal{T} \rangle$, is defined inductively as follows:*

1. $\mathcal{I}, s \models \langle \top, \mathcal{T} \rangle$
2. $\mathcal{I}, s \not\models \langle \bot, \mathcal{T} \rangle$
3. $\mathcal{I}, s \models \langle A, \mathcal{T} \rangle$ *iff* $s \in A^{\mathcal{I}}$, *for all primitive concepts* A
4. $\mathcal{I}, s \models \langle B, \mathcal{T} \rangle$ *iff* $\mathcal{I}, s \models \langle C, \mathcal{T} \rangle$, *for all defined concepts B defined by the axiom* $B \doteq C$ *of* $\mathcal{T}$
5. $\mathcal{I}, s \models \langle \neg C, \mathcal{T} \rangle$ *iff* $\mathcal{I}, s \not\models \langle C, \mathcal{T} \rangle$
6. $\mathcal{I}, s \models \langle C \sqcap D, \mathcal{T} \rangle$ *iff* $\mathcal{I}, s \models \langle C, \mathcal{T} \rangle$ *and* $\mathcal{I}, s \models \langle D, \mathcal{T} \rangle$
7. $\mathcal{I}, s \models \langle C \sqcup D, \mathcal{T} \rangle$ *iff* $\mathcal{I}, s \models \langle C, \mathcal{T} \rangle$ *or* $\mathcal{I}, s \models \langle D, \mathcal{T} \rangle$
8. $\mathcal{I}, s \models \langle \exists R.C, \mathcal{T} \rangle$ *iff* $\mathcal{I}, s' \models \langle C, \mathcal{T} \rangle$, *for some s' such that* $(s, s') \in R^{\mathcal{I}}$
9. $\mathcal{I}, s \models \langle \forall R.C, \mathcal{T} \rangle$ *iff* $\mathcal{I}, s' \models \langle C, \mathcal{T} \rangle$, *for all s' such that* $(s, s') \in R^{\mathcal{I}}$
10. $\mathcal{I}, s \models \langle \exists (u_1)(u_2).P, \mathcal{T} \rangle$ *iff* $P(u_1^{\mathcal{I}}(s), u_2^{\mathcal{I}}(s))$
11. $\mathcal{I}, s \models \langle \exists (u_1)(u_2)(u_3).P, \mathcal{T} \rangle$ *iff* $P(u_1^{\mathcal{I}}(s), u_2^{\mathcal{I}}(s), u_3^{\mathcal{I}}(s))$

A concept C is satisfiable w.r.t. a TBox $\mathcal{T}$ iff $\mathcal{I}, s \models \langle C, \mathcal{T} \rangle$, for some $\mathcal{MTALC}(\mathcal{D}_x)$ interpretation $\mathcal{I}$, and some state $s \in t_{\mathcal{I}}$, in which case the pair $(\mathcal{I}, s)$ is a model of C w.r.t. $\mathcal{T}$; C is insatisfiable (has no models) w.r.t. $\mathcal{T}$, otherwise. C is valid w.r.t. $\mathcal{T}$ iff the negation, $\neg C$, of C is insatisfiable w.r.t. $\mathcal{T}$. The satisfiability problem and the subsumption problem are defined as follows:

The satisfiability problem:
- **Input:** *a concept C and a TBox $\mathcal{T}$*
- **Problem:** *is C satisfiable w.r.t. $\mathcal{T}$?*

The subsumption problem:
- **Input:** *two concepts C and D and a TBox $\mathcal{T}$*
- **Problem:** *does C subsume D w.r.t. $\mathcal{T}$ (notation: $D \sqsubseteq_{\mathcal{T}} C$)? in other words, are all models of C w.r.t. $\mathcal{T}$ also models of D w.r.t. $\mathcal{T}$?*

The satisfiability problem and the subsumption problem are related to each other, as follows: $D \sqsubseteq_{\mathcal{T}} C$ iff $D \sqcap \neg C$ is insatisfiable w.r.t. $\mathcal{T}$.

6 The satisfiability of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept w.r.t. a weakly cyclic TBox

Let C be an $\mathcal{MTALC}(\mathcal{D}_x)$ concept and $\mathcal{T}$ an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox. We define $\mathcal{T} \oplus C$ as the TBox $\mathcal{T}$ augmented with the axiom $B_i \doteq C$, where B_i is a fresh defined concept (not occurring in $\mathcal{T}$):

$$\mathcal{T} \oplus C = \langle \mathcal{T} \cup \{B_i \doteq C\}, B_i \rangle$$

In the sequel, we refer to $\mathcal{T} \oplus C$ as the TBox $\mathcal{T}$ augmented with C, and to B_i as the initial state of $\mathcal{T} \oplus C$. The idea now is that, satisfiability of C w.r.t. $\mathcal{T}$ has (almost) been reduced to the emptiness problem of $\mathcal{T} \oplus C$, seen as a weak alternating automaton on k-ary Σ-trees, for some labelling alphabet Σ to be defined later, with the defined concepts as the states of the automaton, B_i as the initial state of the automaton, the axioms as defining the transition function, with the accepting condition derived from those defined concepts which are not eventuality concepts, and with k standing for the number of concepts of the form $\exists R.D$ in a certain closure, to be defined later, of $\mathcal{T} \oplus C$.

6.1 The Disjunctive Normal Form

The notion of Disjunctive Normal Form (DNF) of a concept C w.r.t. to a TBox $\mathcal{T}$, *dnf1*$(C, \mathcal{T})$, is crucial for the rest of the paper. Such a form results, among other things, from the use of De Morgan's Law to decompose a concept so that, in the final form, the negation symbol outside the scope of a (existential or universal) quantifier occurs only in front of primitive concepts.

Definition 8 (first DNF). *The first Disjunctive Normal Form (*dnf1*) of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept C w.r.t. an $\mathcal{MTALC}(\mathcal{D}_x)$ TBox $\mathcal{T}$,* $\text{dnf1}(C, \mathcal{T})$*, is defined recursively as follows:*

1. *for all primitive concepts A:* $\text{dnf1}(A, \mathcal{T}) = \{\{A\}\}$, $\text{dnf1}(\neg A, \mathcal{T}) = \{\{\neg A\}\}$
2. $\text{dnf1}(\top, \mathcal{T}) = \{\emptyset\}$, $\text{dnf1}(\bot, \mathcal{T}) = \emptyset$
3. *for all defined concepts B:* $\text{dnf1}(B, \mathcal{T}) = \text{dnf1}(E, \mathcal{T})$, $\text{dnf1}(\neg B, \mathcal{T}) = \text{dnf1}(\neg E, \mathcal{T})$, *where E is the right hand side of the axiom $B \doteq E$ defining B*
4. $\text{dnf1}(C \sqcap D, \mathcal{T}) = \prod(\text{dnf1}(C, \mathcal{T}), \text{dnf1}(D, \mathcal{T}))$
5. $\text{dnf1}(C \sqcup D, \mathcal{T}) = \text{dnf1}(C, \mathcal{T}) \cup \text{dnf1}(D, \mathcal{T})$
6. $\text{dnf1}(\exists R.C, \mathcal{T}) = \{\{\exists R.C\}\}$
7. $\text{dnf1}(\forall R.C, \mathcal{T}) = \{\{\forall R.C\}\}$
8. $\text{dnf1}(\exists(u_1)(u_2).P, \mathcal{T}) = \{\{\exists(u_1)(u_2).P\}\}$
9. $\text{dnf1}(\exists(u_1)(u_2)(u_3).P, \mathcal{T}) = \{\{\exists(u_1)(u_2)(u_3).P\}\}$
10. $\text{dnf1}(\neg(C \sqcap D), \mathcal{T}) = \text{dnf1}(\neg C, \mathcal{T}) \cup \text{dnf1}(\neg D, \mathcal{T})$
11. $\text{dnf1}(\neg(C \sqcup D), \mathcal{T}) = \prod(\text{dnf1}(\neg C, \mathcal{T}), \text{dnf1}(\neg D, \mathcal{T}))$
12. $\text{dnf1}(\neg\exists R.C, \mathcal{T}) = \{\{\forall R.\neg C\}\}$
13. $\text{dnf1}(\neg\forall R.C, \mathcal{T}) = \{\{\exists R.\neg C\}\}$

where $\prod$ is defined as follows:

1. $\prod(\{S\}, \{T\}) = \begin{cases} \emptyset & \textit{if } \{A, \neg A\} \subseteq S \cup T \textit{ for some primitive concept } A, \\ \{S \cup T\} & \textit{otherwise} \end{cases}$
2. $\prod(\{S_1, \ldots, S_n\}, \{T_1, \ldots, T_m\}) = \bigcup_{i \in \{1,\ldots,n\}, j \in \{1,\ldots,m\}} \prod(\{S_i\}, \{T_j\})$

Note that the *dnf1* function checks satisfiability at the propositional level, in the sense that, given a concept C, *dnf1*$(C, \mathcal{T})$ is either empty, or is such that for all $S \in$ *dnf1*$(C, \mathcal{T})$, S does not contain both A and $\neg A$, A being a primitive concept. Furthermore, given a set $S \in$ *dnf1*$(C, \mathcal{T})$, all elements of S are concepts of either of the following forms:

1. A or $\neg A$, where A is a primitive concept;
2. $\exists R.D$;
3. $\forall R.D$; or
4. $\exists(u_1)(u_2).P$ if x binary, $\exists(u_1)(u_2)(u_3).P$ if x ternary.

Definition 9. *Let C be an $\mathcal{MTALC}(\mathcal{D}_x)$ concept, $\mathcal{T}$ and $\mathcal{MTALC}(\mathcal{D}_x)$ TBox and $S \in \mathrm{dnf1}(C, \mathcal{T})$. The set of concrete features of S, $\mathrm{cFeatures}(S)$, is defined as the set of concrete features, g, for which there exists a feature chain u suffixed by g, such that S contains a predicate concept $\exists(u_1)(u_2).P$, with $u \in \{u_1, u_2\}$, if x binary; or S contains a predicate concept $\exists(u_1)(u_2)(u_3).P$, with $u \in \{u_1, u_2, u_3\}$, if x ternary.*

Definition 10 (the *pc*∃∀ partition). *Let C be an $\mathcal{MTALC}(\mathcal{D}_x)$ concept, $\mathcal{T}$ an $\mathcal{MTALC}(\mathcal{D}_x)$ TBox, $S \in \mathrm{dnf1}(C, \mathcal{T})$ and N_{aF}^* the language of all finite words over the alphabet N_{aF}. The* pc∃∀ *partition of S, $\mathrm{pc}\exists\forall(S)$, is defined as $\mathrm{pc}\exists\forall(S) = S_{prop} \cup S_{csp} \cup S_\exists \cup S_\forall$, where:*

$$\begin{aligned}
S_{prop} &= \{A : A \in S \textit{ and } A \textit{ primitive concept}\} \\
&\quad \cup \{\neg A : \neg A \in S \textit{ and } A \textit{ primitive concept}\} \\
S_{csp} &= \begin{cases} \{\exists(u_1)(u_2).P : \exists(u_1)(u_2).P \in S\}, & \textit{if } x \textit{ binary} \\ \{\exists(u_1)(u_2)(u_3).P : \exists(u_1)(u_2)(u_3).P \in S\}, & \textit{if } x \textit{ ternary} \end{cases} \\
S_\exists &= \{\exists R.C : \exists R.C \in S\} \\
S_\forall &= \{\forall R.C : \forall R.C \in S\}
\end{aligned}$$

If *dnf1*$(C, \mathcal{T}) = \{S_1, \ldots, S_n\}$ then C is satisfiable w.r.t. $\mathcal{T}$ iff for some $i = 1 \ldots n$, S_i is satisfiable w.r.t. $\mathcal{T}$. On the other hand, the following conditions are necessary for the satisfiability of an element S of *dnf1*$(C, \mathcal{T})$:

1. S_{prop} does not contain A and $\neg A$, where A is a primitive concept;
2. The CSP induced by S (see the definition below) is consistent;
3. for all concepts $\exists R.D$ in $S_\exists$, where R is a general, not necessarily functional role, the conjunction $D \sqcap \sqcap_{\forall R.D' \in S_\forall} D'$ is a consistent concept (recursive call of concept consistency). This point is expected to clarify the reader the idea of distributing all $\forall R$-prefixed concepts over each $\exists R$-prefixed concept; and
4. for all abstract features $f \in N_{aF}$, such that there exists a concept $\exists f.D$ in $S_\exists$, the conjunction $\sqcap_{\exists f.D \in S_\exists} D \sqcap \sqcap_{\forall f.D \in S_\forall} D$ is a consistent concept (again, recursive call of concept consistency).

With the help of the just-above explanation, given a set $S \in$ *dnf1*$(C, \mathcal{T})$, we can replace S with the equivalent set S^f computed as follows:

1. The semantics suggests that, for all general, not necessarily functional roles R, whenever S contains a concept of the form $\exists R.D$, the tableaux method would create an R-successor S' containing the concept D and all concepts E such that $\forall R.E$ belongs to S:

— initialise T to S: $T \leftarrow S$

— for all elements of S of the form $\exists R.D$:
$T \leftarrow (T \setminus \{\exists R.D\}) \cup \{\exists R.(D \sqcap \sqcap_{\forall R.E \in S_\forall} E)\}$

2. A similar work has to be done for abstract features f such that S contains elements of the form $\exists f.D$, bearing in mind that abstract features are functional. For all such f, we replace the subset $\{\exists f.D : \exists f.D \in S\} \cup \{\forall f.D : \forall f.D \in S\}$ by the singleton set $\{\exists f.(\sqcap_{\exists f.D \in S} D \sqcap \sqcap_{\forall f.D \in S} D)\}$. The motivation, again, comes straight from the semantics: because abstract features are functional, only one f-successor to S has to be created, which has to satisfy all concepts D such that $\exists f.D \in S$, as well as all concepts D such that $\forall f.D \in S$:

— for all abstract features $f \in N_{aF}$, such that S contains elements of the form $\exists f.D$:
$T \leftarrow T \setminus \{\exists f.D : \exists f.D \in S\}$
$T \leftarrow T \cup \{\exists f.(\sqcap_{\exists f.D \in S} D \sqcap \sqcap_{\forall f.D \in S} D)\}$

3. remove from T all elements of the form $\forall R.D$: $T \leftarrow T \setminus S_\forall$
4. $S^f \leftarrow T$

The second *dnf* of a concept C w.r.t. a TBox $\mathcal{T}$, *dnf2*$(C, \mathcal{T})$, is now introduced. This consists of the *dnf1* of C w.r.t. $\mathcal{T}$, *dnf1*$(C, \mathcal{T})$, as given by Definition 8, in which each element S is replaced with S^f, computed as shown just above. Formally:

Definition 11 (second DNF). *Let $x \in \{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$, C be an $\mathcal{MTALC}(\mathcal{D}_x)$ concept, and $\mathcal{T}$ an $\mathcal{MTALC}(\mathcal{D}_x)$ TBox. The second Disjunctive Normal Form (*dnf2*) of C w.r.t. $\mathcal{T}$, $\mathrm{dnf2}(C, \mathcal{T})$, is defined as $\mathrm{dnf2}(C, \mathcal{T}) = \{S^f : S \in \mathrm{dnf1}(C, \mathcal{T})\}$.*

Given an $\mathcal{MTALC}(\mathcal{D}_x)$ concept C and an $\mathcal{MTALC}(\mathcal{D}_x)$ TBox $\mathcal{T}$, we can now use the second DNF, *dnf2*, to define the closure $(T \oplus C)^*$ of $T \oplus C$, the TBox T augmented with C. Initially, $(T \oplus C)^* = T \oplus C$, and no defined concept in $(T \oplus C)^*$ is marked. Then we repeat the following process until all defined concepts in $(T \oplus C)^*$ are marked. We consider an axiom $B_1 \doteq E$ of $(T \oplus C)^*$ such that B_1 is not marked. We mark B_1. We compute *dnf2*$(E, (T \oplus C)^*)$. For all $S \in$ *dnf2*$(E, (T \oplus C)^*)$, S is of the form $S_{prop} \cup S_{csp} \cup S_\exists$. For all such S, we do the following. We consider in turn the elements $\exists R.D$ in $S_\exists$. If D is a defined concept of $(T \oplus C)^*$ then we do nothing. Otherwise, if $(T \oplus C)^*$ has an axiom of the form $B_2 \doteq D$, then we replace D with B_2 in $\exists R.D$. Otherwise, we add to $(T \oplus C)^*$ the axiom $B_2 \doteq D$, and we replace, in S, $\exists R.D$ with $\exists R.B_2$. Formally, $(T \oplus C)^*$ is defined as follows.

Definition 12 (closure of $T \oplus C$). *Let C be an $\mathcal{MTALC}(\mathcal{D}_x)$ concept and T an $\mathcal{MTALC}(\mathcal{D}_x)$ TBox. The closure $(T \oplus C)^*$ of $T \oplus C$ is defined by the procedure of Figure 5. The initial defined concept of $(T \oplus C)^*$ is the same as the initial defined concept of $T \oplus C$.*

We also need the closure of a concept C w.r.t. a TBox $\mathcal{T}$, *cl*$(C, \mathcal{T})$, which is defined recursively as the union of *dnf2*$(C, \mathcal{T})$, and the closures, w.r.t. $\mathcal{T}$, of all concepts C' such that for some $S \in$ *dnf2*$(C, \mathcal{T})$ and some general (possibly functional) role R, $\exists R.C' \in S$. Formally:

Definition 13 (closure of a concept w.r.t. a TBox). *The closure of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept C w.r.t. an $\mathcal{MTALC}(\mathcal{D}_x)$ TBox $\mathcal{T}$, $\mathrm{cl}(C, \mathcal{T})$, is defined recursively as follows:*

$$\mathrm{cl}(C, \mathcal{T}) = \mathrm{dnf2}(C, \mathcal{T}) \cup \bigcup_{\exists R.C' \in S \in \mathrm{dnf2}(C, \mathcal{T})} \mathrm{cl}(C', \mathcal{T})$$

Input: an $\mathcal{MTALC}(\mathcal{D}_x)$ concept C and an $\mathcal{MTALC}(\mathcal{D}_x)$ TBox T
Output: the closure $(T \oplus C)^*$ of $T \oplus C$
Initialise $(T \oplus C)^*$ to $T \oplus C$: $(T \oplus C)^* \leftarrow T \oplus C$;
Initially, no defined concept of $(T \oplus C)^*$ is marked;
while($(T \oplus C)^*$ contains defined concepts that are not marked){
  consider a non marked defined concept B_1 from $(T \oplus C)^*$;
  let $B_1 \doteq E$ be the axiom from $(T \oplus C)^*$ defining B_1;
  mark B_1;
  compute *dnf2*$(E, (T \oplus C)^*)$;
  for all $\exists R.D \in S \in$ *dnf2*$(E, (T \oplus C)^*)$
    if D is not a defined concept of $(T \oplus C)^*$ *then*
      if($(T \oplus C)^*$ contains an axiom of the form $B_2 \doteq D$) *then*
        replace $\exists R.D$ with $\exists R.B_2$ in S;
      else{
        add the axiom $B_2 \doteq D$ to $(T \oplus C)^*$, where B_2 is a fresh defined concept:
            $(T \oplus C)^* \leftarrow (T \oplus C)^* \cup \{B_2 \doteq D\}$;
        replace $\exists R.D$ with $\exists R.B_2$ in S;
        }
  }

Fig. 5. Closure $(T \oplus C)^*$ of a TBox T augmented with a concept C, $T \oplus C$.

Definition 14. *Let C be an $\mathcal{MTALC}(\mathcal{D}_x)$ concept and $\mathcal{T}$ an $\mathcal{MTALC}(\mathcal{D}_x)$ TBox. We denote by:*

1. $\text{cFeatures}(C, \mathcal{T}) = \bigcup_{S \in \text{cl}(C,\mathcal{T})} \text{cFeatures}(S)$, *the set of concrete features of C w.r.t. $\mathcal{T}$;*
2. $\text{ncf}(C, \mathcal{T}) = |\text{cFeatures}(C, \mathcal{T})|$, *the number of concrete features of C w.r.t. $\mathcal{T}$;*
3. $\text{aFeatures}(C, \mathcal{T}) = \{f \in N_{aF} : \exists D$ *s. t.* $\exists f.D \in S \in \text{cl}(C, \mathcal{T})\}$, *the set of abstract features of C w.r.t. $\mathcal{T}$;*
4. $\text{naf}(C, \mathcal{T}) = |\text{aFeatures}(C, \mathcal{T})|$, *the number of abstract features of C w.r.t. $\mathcal{T}$;*
5. $\text{pConcepts}(C, \mathcal{T}) = \{A : \exists S \in \text{cl}(C, \mathcal{T})$ *s. t.* $\{A, \neg A\} \cap S_{prop} \neq \emptyset\}$, *the set of primitive concepts of C w.r.t. $\mathcal{T}$;*
6. $\text{dConcepts}(C, \mathcal{T})$ *is the set of defined concepts in $(T \oplus C)^*$;*
7. $\text{eConcepts}(C, \mathcal{T})$, *the set of existential (sub)concepts of C w.r.t. $\mathcal{T}$, is the union of all $\exists R.D$ such that there exists an axiom $B \doteq E$ in $(T \oplus C)^*$ and S in E, so that $\exists R.D \in S$;*
8. $\text{feConcepts}(C, \mathcal{T}) = \{\exists f.D \in \text{eConcepts}(C, \mathcal{T}) :$ *f abstract feature*$\}$, *the set of functional existential concepts of C w.r.t. $\mathcal{T}$;*
9. $\text{reConcepts}(C, \mathcal{T}) = \text{eConcepts}(C, \mathcal{T}) \setminus \text{feConcepts}(C, \mathcal{T})$, *the set of relational existential concepts of C w.r.t. $\mathcal{T}$;*
10. $\text{fbf}(C, \mathcal{T}) = \text{naf}(C, \mathcal{T})$, *the functional branching factor of C w.r.t. $\mathcal{T}$;*
11. $\text{rbf}(C, \mathcal{T}) = |\text{reConcepts}(C, \mathcal{T})|$, *the relational branching factor of C w.r.t. $\mathcal{T}$;*
12. $\text{bf}(C, \mathcal{T}) = \text{fbf}(C, \mathcal{T}) + \text{rbf}(C, \mathcal{T})$, *the branching factor of C w.r.t. $\mathcal{T}$.*

We suppose that the relational existential concepts in *reConcepts*$(C, \mathcal{T})$ are ordered, and refer to the i-th element of *reConcepts*$(C, \mathcal{T})$, $i = 1 \ldots$ *rbf*$(C, \mathcal{T})$, as $rec_i(C, \mathcal{T})$. Similarly, we suppose that the abstract features in *aFeatures*$(C, \mathcal{T})$ are ordered, and refer to the i-th element of $aFeatures(C,\mathcal{T})$, $i = 1 \dots fbf(C,\mathcal{T})$, as $af_i(C,\mathcal{T})$. Together, they constitute the directions of the weak alternating automaton to be associated with he satisfiability of C w.r.t. $\mathcal{T}$.

Definition 15. *Let C be an $\mathcal{MTALC}(\mathcal{D}_x)$ concept and $\mathcal{T}$ an $\mathcal{MTALC}(\mathcal{D}_x)$ TBox. The branching tuple of C is given by the ordered* $\mathrm{bf}(C,\mathcal{T})$*-tuple* $\mathrm{bt}(C,\mathcal{T}) = (\mathrm{rec}_1(C,\mathcal{T}),\dots,\mathrm{rec}_{\mathrm{rbf}(C,\mathcal{T})}(C,\mathcal{T}),\mathrm{af}_1(C,\mathcal{T}),\dots,\mathrm{af}_{\mathrm{fbf}(C,\mathcal{T})}(C,\mathcal{T}))$ *of the* $\mathrm{rbf}(C,\mathcal{T})$ *relational existential concepts in* $\mathrm{reConcepts}(C,\mathcal{T})$ *and the* $\mathrm{fbf}(C,\mathcal{T})$ *abstract features in* $\mathrm{aFeatures}(C,\mathcal{T})$.

Given an $\mathcal{MTALC}(\mathcal{D}_x)$ concept C and an $\mathcal{MTALC}(\mathcal{D}_x)$ TBox $\mathcal{T}$, we will be interested in k-ary Σ-trees (see Definition 5), t, verifying the following:

1. $k = bf(C,\mathcal{T})$; and
2. $M = 2^{pConcepts(C,\mathcal{T})} \times \Theta(cFeatures(C,\mathcal{T}),\Delta_{\mathcal{D}_x})$, where $\Theta(cFeatures(C,\mathcal{T}),\Delta_{\mathcal{D}_x})$ is the set of total functions $\theta : cFeatures(C,\mathcal{T}) \rightarrow \Delta_{\mathcal{D}_x}$ associating with each concrete feature g in $cFeatures(C,\mathcal{T})$ a concrete value $\theta(g)$ from the spatial concrete domain $\Delta_{\mathcal{D}_x}$.

Such a tree will be seen as representing a class of interpretations of the satisfiability of C w.r.t. $\mathcal{T}$: the label (X,θ) of a node $\alpha \in \{1,\dots,bf(C,\mathcal{T})\}^*$, with $X \subseteq pConcepts(C,\mathcal{T})$ and $\theta \in \Theta(cFeatures(C,\mathcal{T}),\Delta_{\mathcal{D}_x})$, is to be interpreted as follows:

1. X records the information on the primitive concepts that are true at α, in all interpretations of the class; and
2. $\theta : cFeatures(C,\mathcal{T}) \rightarrow \Delta_{\mathcal{D}_x}$ records the values, at the abstract object represented by node α, of the concrete features $g_1,\dots,g_{ncf(C,\mathcal{T})}$ in $cFeatures(C,\mathcal{T})$.

The crucial question is when we can say that an interpretation of the class is a model of C w.r.t. $\mathcal{T}$. To answer the question, we consider (weak) alternating automata on k-ary Σ-trees, with $k = bf(C,\mathcal{T})$ and $\Sigma = 2^{pConcepts(C,\mathcal{T})} \times \Theta(cFeatures(C,\mathcal{T}),\Delta_{\mathcal{D}_x})$. We then show how to associate such an automaton with the satisfiability of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept C w.r.t. a weakly cyclic TBox $\mathcal{T}$, in such a way that the models of C w.r.t. $\mathcal{T}$ coincide with the k-ary Σ-trees accepted by the automaton. The background on alternating automata has been adapted from [58].

7 Weak alternating automata and $\mathcal{MTALC}(\mathcal{D}_x)$ with weakly cyclic Tboxes

We now provide the required background on weak alternating automata, adapted from [58] (see also [39, 40, 59, 60]). We then show how to associate such an automaton with the satisfiability problem of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept w.r.t. an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox, so that the language accepted by the automaton coincides with the set of models of the concept w.r.t. to the TBox.

7.1 Weak alternating automata

Definition 16 (free distributive lattice). *Let S be a set of generators. $\mathcal{L}(S)$ denotes the free distributive lattice generated by S. $\mathcal{L}(S)$ can be thought of as the set of logical formulas built from variables taken from S using the disjunction and conjunction operators $\vee$ and $\wedge$ (but not the negation operator $\neg$). In other words, $\mathcal{L}(S)$ is the smallest set such that:*

1. *for all $s \in S$, $s \in \mathcal{L}(S)$; and*
2. *if e_1 and e_2 belong to $\mathcal{L}(S)$, then so do $e_1 \wedge e_2$ and $e_1 \vee e_2$.*

Each element $e \in \mathcal{L}(S)$ has, up to isomorphism, a unique representation in *DNF* (Disjunctive Normal Form), $e = \bigvee_i C_i$ (each C_i is a conjunction of generators from S, and no C_i subsumes C_k, with $k \neq i$). We suppose, without loss of generality, that each element of $\mathcal{L}(S)$ is written in such a form. If $e = \bigvee_i \bigwedge_j s_{ij}$ is an element of $\mathcal{L}(S)$, the dual of e is the element $\tilde{e} = \bigwedge_i \bigvee_j s_{ij}$ obtained by interchanging $\vee$ and $\wedge$ ($\bigwedge_i \bigvee_j s_{ij}$ is not necessarily in *DNF*).

Definition 17 (set representation). *Let S be a set of generators, $\mathcal{L}(S)$ the free distributive lattice generated by S, and e an element of $\mathcal{L}(S)$. Write e in DNF as $\bigvee_{i=1}^{n} \bigwedge_{j=1}^{n_i} s_{ij}$. The set representation of e,* set-rep(e), *is the subset of 2^S defined as $\{S_1, \ldots, S_n\}$, with $S_i = \{s_{i1}, \ldots, s_{in_i}\}$.*

In the following, we denote by K a set of k directions $d_1, \ldots, d_k$; by N_P a set of primitive concepts; by x a spatial RA from the set $\{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$; by N_{cF} a finite set of concrete features referring to objects in $\Delta_{\mathcal{D}_x}$; by $\Sigma(x, N_P, N_{cF})$ the alphabet $2^{N_P} \times \Theta(N_{cF}, \Delta_{\mathcal{D}_x})$, $\Theta(N_{cF}, \Delta_{\mathcal{D}_x})$ being the set of total functions $\theta : N_{cF} \rightarrow \Delta_{\mathcal{D}_x}$, associating with each concrete feature g a concrete value $\theta(g)$ from the spatial concrete domain $\Delta_{\mathcal{D}_x}$; by $\mathcal{L}it(N_P)$ the set of literals derived from N_P (viewed as a set of atomic propositions): $\mathcal{L}it(N_P) = N_P \cup \{\neg A : A \in N_P\}$; by $c(2^{\mathcal{L}it(N_P)})$ the set of subsets of $\mathcal{L}it(N_P)$ which do not contain a primitive concept and its negation: $c(2^{\mathcal{L}it(N_P)}) = \{S \subset \mathcal{L}it(N_P) : (\forall A \in N_P)(\{A, \neg A\} \not\subseteq S)\}$; by $constr(x, K, N_{cF})$ the set of constraints of the form $P(u_1, u_2)$, if x binary, and $P(u_1, u_2, u_3)$, if x ternary, with P being an x relation, u_1, u_2 and u_3 *K^*N_{cF}-chains* (i.e., each of u_1, u_2 and u_3 is of the form g or $d_{i_1} \ldots d_{i_n} g$, $n \geq 1$ and n finite, the d_{i_j}'s being directions in K, and g a concrete feature).

Definition 18 (alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees). *Let $k \geq 1$ be an integer and $K = \{d_1, \ldots, d_k\}$ a set of directions. An alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees is a tuple $\mathcal{A} = (\mathcal{L}(\mathcal{L}\text{it}(N_P) \cup \text{constr}(x, K, N_{cF}) \cup K \times Q), \Sigma(x, N_P, N_{cF}), \delta, q_0, \mathcal{F})$, where Q is a finite set of states; $\Sigma(x, N_P, N_{cF})$ is the input alphabet (labelling the nodes of the input trees); $\delta : Q \rightarrow \mathcal{L}(\mathcal{L}\text{it}(N_P) \cup \text{constr}(x, K, N_{cF}) \cup K \times Q)$ is the transition function; $q_0 \in Q$ is the initial state; and $\mathcal{F}$ defines the acceptance condition:*

1. *$\mathcal{F} \subseteq Q$ in case of a Büchi alternating automaton; and*
2. *$\mathcal{F} \subseteq 2^Q$ in case of a Muller alternating automaton.*

A weak alternating automaton is a special case of a Muller alternating automaton.

Definition 19 (Weak alternating automaton [58]). *Let $\mathcal{A}$ be a Muller alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees, as defined in Definition 18. $\mathcal{A}$ is said to be a weak alternating automaton if there exists a partition $Q = \bigcup_{i=1}^{n} Q_i$ of the set Q of states, and a partial order $\geq$ on the collection of the Q_i's, so that:*

1. *the transition function δ has the property that, given two states $q \in Q_i$ and $q' \in Q_j$, if q' occurs in $\delta(q)$ then $Q_i \geq Q_j$; and*
2. *the set $\mathcal{F}$ giving the acceptance condition is a subset of $\{Q_1, \ldots, Q_n\}$.*

Let $\mathcal{A}$ be an alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees, as defined in Definition 18, and t a k-ary $\Sigma(x, N_P, N_{cF})$-tree. Given two alphabets Σ_1 and Σ_2, we denote by $\Sigma_1\Sigma_2$ the concatenation of Σ_1 and Σ_2, consisting of all words ab, with $a \in \Sigma_1$ and $b \in \Sigma_2$. In a run $r(\mathcal{A}, t)$ of $\mathcal{A}$ on t (see below), which can be seen as an unfolding of a branch of the computation tree $T(\mathcal{A}, t)$ of $\mathcal{A}$ on t, as defined in [59, 58, 60], the nodes of level n will represent one possibility for choices of $\mathcal{A}$ up to level n in t. For each $n \geq 0$, we define the set of n-histories to be the set $H_n = \{q_0\}(KQ)^n$ of all $2n+1$-length words consisting of q_0 as the first letter, followed by a $2n$-length word $d_{i_1}q_{i_1}\ldots d_{i_n}q_{i_n}$, with $d_{i_j} \in K$ and $q_{i_j} \in Q$, for all $j = 1\ldots n$. If $h \in H_n$ and $g \in KQ$ then hg, the concatenation of h and g, belongs to H_{n+1}. More generally, if $h \in H_n$ and $e \in \mathcal{L}(KQ)$, the concatenation he of h and e will denote the element of $\mathcal{L}(H_{n+1})$ obtained by prefixing h to each generator in KQ which occurs in e. Additionally, given an n-history $h = q_0d_{i_1}q_{i_1}\ldots d_{i_n}q_{i_n}$, with $n \geq 0$, we denote

1. by *Last*(h) the initial state q_0 if h consists of the 0-history q_0 ($n = 0$), and the state q_{i_n} if $n \geq 1$;
2. by K-*proj*(h) (the K-projection of h) the empty word ϵ if $n = 0$, and the n-length word $d_{i_1}\ldots d_{i_n}$ otherwise; and
3. by Q-*proj*(h) (the Q-projection of h) the state q_0 if $n = 0$, and the $n+1$-length word $q_0q_{i_1}\ldots q_{i_n} \in Q^{n+1}$ otherwise.

The union of all H_n, with n finite, will be referred to as the set of finite histories of $\mathcal{A}$, and denoted by $H_{<\infty}$. We denote by $\Sigma(2^{H_{<\infty}}, N_P, x, K, N_{cF})$ the alphabet $2^{H_{<\infty}} \times c(2^{\mathcal{L}it(N_P)}) \times 2^{constr(x,K,N_{cF})}$, by $\Sigma(2^Q, N_P, x, K, N_{cF})$ the alphabet $2^Q \times c(2^{\mathcal{L}it(N_P)}) \times 2^{constr(x,K,N_{cF})}$, and, in general, by $\Sigma(S, N_P, x, K, N_{cF})$ the alphabet $S \times c(2^{\mathcal{L}it(N_P)}) \times 2^{constr(x,K,N_{cF})}$.

A run of the alternating automaton $\mathcal{A}$ on t is now introduced.

Definition 20 (Run). *Let $\mathcal{A}$ be an alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees, as defined in Definition 18, and t a k-ary $\Sigma(x, N_P, N_{cF})$-tree. A run, $r(\mathcal{A}, t)$, of $\mathcal{A}$ on t is a partial k-ary $\Sigma(2^{H_{<\infty}}, N_P, x, K, N_{cF})$-tree defined inductively as follows. For all directions $d \in K$, and for all nodes $u \in K^*$ of $r(\mathcal{A}, t)$, u has at most one outgoing edge labelled with d, and leading to the d-successor ud of u. The label $(Y_\epsilon, L_\epsilon, X_\epsilon)$ of the root belongs to $2^{H_0} \times c(2^{\mathcal{L}\mathrm{it}(N_P)}) \times 2^{\mathrm{constr}(x,K,N_{cF})}$ —in other words, $Y_\epsilon = \{q_0\}$. If u is a node of $r(\mathcal{A}, t)$ of level $n \geq 0$, with label (Y_u, L_u, X_u), then calculate $e = \bigwedge_{h \in Y_u} \mathrm{dist}(h, \delta(\mathrm{Last}(h)))$, where* dist *is a function associating with each pair (h_1, e_1) of $H_{<\infty} \times \mathcal{L}(\mathcal{L}\mathrm{it}(N_P) \cup \mathrm{constr}(x, K, N_{cF}) \cup K \times Q)$ an element of $\mathcal{L}(\mathcal{L}\mathrm{it}(N_P) \cup \mathrm{constr}(x, K, N_{cF}) \cup H_{<\infty})$ defined inductively in the following way:*

$$\mathrm{dist}(h_1,e_1)=\begin{cases} e_1 & \text{if } e_1\in \mathcal{L}\mathrm{it}(N_P)\cup \mathrm{constr}(x,K,N_{cF}),\\ h_1dq & \text{if } e_1=(d,q), \text{ for some } (d,q)\in K\times Q,\\ \mathrm{dist}(h_1,e_2)\vee \mathrm{dist}(h_1,e_3) & \text{if } e_1=e_2\vee e_3,\\ \mathrm{dist}(h_1,e_2)\wedge \mathrm{dist}(h_1,e_3) & \text{if } e_1=e_2\wedge e_3\end{cases}$$

Write e in dnf *as $e=\bigvee_{i=1}^{r}(L_i\wedge X_i\wedge Y_i)$, where the L_i's are conjunctions of literals from $\mathcal{L}\mathrm{it}(N_P)$, the X_i's are conjunctions of constraints from $\mathrm{constr}(x,K,N_{cF})$, and the Y_i's are conjunctions of $n+1$-histories. Then there exists $i=1\ldots r$ such that $L_u=\{\ell\in\mathcal{L}\mathrm{it}(N_P):\ell \text{ occurs in } L_i\}$; $X_u=\{x\in\mathrm{constr}(x,K,N_{cF}):x \text{ occurs in } X_i\}$; for all $d\in K$, such that the set $Y=\{hdq\in H_{n+1}:h\in H_n\wedge q\in Q\wedge(hdq \text{ occurs in } Y_i)\}$ is nonempty, and only for those d, u has a d-successor, ud, whose label (Y_{ud},X_{ud},L_{ud}) is such that $Y_{ud}=Y$; and the label $t(u)=(\mathcal{P}_u,\theta_u)\in 2^{N_P}\times\Theta(N_{cF},\Delta_{\mathcal{D}_x})$ of the node u of the input tree t verifies the following, where, given a node v in t, the notation θ_v consists of the function $\theta_v:N_{cF}\to\Delta_{\mathcal{D}_x}$ which is the second argument of $t(v)$:*

- *for all $A\in N_P$: if $A\in L_u$ then $A\in\mathcal{P}_u$; and if $\neg A\in L_u$ then $A\notin\mathcal{P}_u$ (the elements A of N_P such that, neither A nor $\neg A$ occur in L_u, may or may not occur in $\mathcal{P}_u$);*
- *if x binary, for all $P(d_{i_1}\ldots d_{i_n}g_1,d_{j_1}\ldots d_{j_m}g_2)$ appearing in X_u, $P(\theta_{ud_{i_1}\ldots d_{i_n}}(g_1),\theta_{ud_{j_1}\ldots d_{j_m}}(g_2))$ holds. In other words, the value of the concrete feature g_1 at the $d_{i_1}\ldots d_{i_n}$-successor of u in t, on the one hand, and the value of the concrete feature g_2 at the $d_{j_1}\ldots d_{j_m}$-successor of u in t, on the other hand, are related by the x relation P.*
- *similarly, if x ternary, for all $P(d_{i_1}\ldots d_{i_n}g_1,d_{j_1}\ldots d_{j_m}g_2,d_{l_1}\ldots d_{l_p}g_3)$ appearing in X_u, $P(\theta_{ud_{i_1}\ldots d_{i_n}}(g_1),\theta_{ud_{j_1}\ldots d_{j_m}}(g_2),\theta_{ud_{l_1}\ldots d_{l_p}}(g_3))$ holds.*

A partial k-ary $\Sigma(2^{H_{<\infty}},N_P,x,K,N_{cF})$-tree σ is a run of $\mathcal{A}$ if there exists a k-ary $\Sigma(x,N_P,N_{cF})$-tree t such that σ is a run of $\mathcal{A}$ on t.

Definition 21 (CSP of a run). *Let $\mathcal{A}$ be an alternating automaton on k-ary $\Sigma(x,N_P,N_{cF})$-trees, as defined in Definition 18, and σ a run of $\mathcal{A}$:*

1. *for all nodes v of σ, of label $\sigma(v)=(Y_v,L_v,X_v)\in 2^{H_{<\infty}}\times c(2^{\mathcal{L}\mathrm{it}(N_P)})\times 2^{\mathrm{constr}(x,K,N_{cF})}$, the argument X_v gives rise to the CSP of σ at v, $\mathrm{CSP}_v(\sigma)$, whose set of variables, $V_v(\sigma)$, and set of constraints, $C_v(\sigma)$, are defined as follows:*
   (a) *Initially, $V_v(\sigma)=\emptyset$ and $C_v(\sigma)=\emptyset$*
   (b) *for all K^*N_{cF}-*chains *$d_{i_1}\ldots d_{i_n}g$ appearing in X_v, create, and add to $V_v(\sigma)$, a variable $\langle vd_{i_1}\ldots d_{i_n},g\rangle$*
   (c) *if x binary, for all $P(d_{i_1}\ldots d_{i_n}g_1,d_{j_1}\ldots d_{j_m}g_2)$ in X_v, add the constraint $P(\langle vd_{i_1}\ldots d_{i_n},g_1\rangle,\langle vd_{j_1}\ldots d_{j_m},g_2\rangle)$ to $C_v(\sigma)$*
   (d) *similarly, if x ternary, for all $P(d_{i_1}\ldots d_{i_n}g_1,d_{j_1}\ldots d_{j_m}g_2,d_{l_1}\ldots d_{l_p}g_3)$ in X_v, add the constraint $P(\langle vd_{i_1}\ldots d_{i_n},g_1\rangle,\langle vd_{j_1}\ldots d_{j_m},g_2\rangle,\langle vd_{l_1}\ldots d_{l_p},g_3\rangle)$ to $C_v(\sigma)$*
2. *the CSP of σ, $\mathrm{CSP}(\sigma)$, is the CSP whose set of variables, $\mathcal{V}(\sigma)$, and set of constraints, $\mathcal{C}(\sigma)$, are defined as $\mathcal{V}(\sigma)=\bigcup_{v \text{ node of } \sigma}V_v(\sigma)$ and $\mathcal{C}(\sigma)=\bigcup_{v \text{ node of } \sigma}C_v(\sigma)$.*

An n-branch of a run $\sigma=r(\mathcal{A},t)$ is a path of length (number of edges) n beginning at the root of σ. A branch is an infinite path. If u is the terminal node of an n-branch β, then the argument Y_u of the label (Y_u,L_u,X_u) of u is a set of n-histories.

Following [58], we say that each n-history in Y_u lies along β. An n-history h lies along σ if there exists an n-branch β of σ such that h lies along β. An (infinite) history is a sequence $h = q_0 d_{i_1} q_{i_1} \dots d_{i_n} q_{i_n} \dots \in \{q_0\}(KQ)^\omega$. Given such a history, $h = q_0 d_{i_1} q_{i_1} \dots d_{i_n} q_{i_n} \dots \in \{q_0\}(KQ)^\omega$:

1. h lies along a branch β if, for every $n \geq 1$, the prefix of h consisting of the n-history $q_0 d_{i_1} q_{i_1} \dots d_{i_n} q_{i_n}$ lies along the n-branch β_n consisting of the first n edges of β;
2. h lies along σ if there exists a branch β of σ such that h lies along β;
3. Q-*proj*(h) (the Q-projection of h) is the infinite word $q_0 q_{i_1} \dots q_{i_n} \dots \in Q^\omega$ such that, for all $n \geq 1$, the $n+1$-length prefix $q_0 q_{i_1} \dots q_{i_n}$ is the Q-projection of h_n, the n-history which is the $2n+1$-prefix of h.
4. we denote by *Inf*(h) the set of states appearing infinitely often in Q-*proj*(h)

The acceptance condition is now defined as follows. In the Büchi case, a history h is accepting if *Inf*$(h) \cap \mathcal{F} \neq \emptyset$. In the case of a weak alternating automaton, h is accepting if *Inf*$(h) \subseteq Q_i$, for some $Q_i \in \mathcal{F}$.[5] A branch β of $r(\mathcal{A}, t)$ is accepting if every history lying along β is accepting.

The condition for a run σ to be accepting splits into two subconditions. The first subcondition is the standard one, and is related to (the histories lying along) the branches of σ, all of which should be accepting. The second subcondition is new and is the same for both kinds of automata: the CSP of σ, *CSP*(σ), should be consistent. $\mathcal{A}$ accepts a k-ary $\Sigma(x, N_P, N_{cF})$-tree t if there exists an accepting run of $\mathcal{A}$ on t. The language $\mathcal{L}(\mathcal{A})$ accepted by $\mathcal{A}$ is the set of all k-ary $\Sigma(x, N_P, N_{cF})$-trees accepted by $\mathcal{A}$.

Informally, a run σ is uniform if, for all $n \geq 0$, any two n-histories lying along σ, and suffixed (i.e., terminated) by the same state, make the same transition. To define it formally, we suppose that the transition function δ is given as a disjunction of conjunctions, in *dnf*.

Definition 22 (Uniform run). *Let $\mathcal{A}$ be an alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees, as defined in Definition 18, and σ a run of $\mathcal{A}$. σ is said to be a uniform run iff it satisfies the following. For all $n \geq 0$, select for each state q in Q, one conjunct from $\delta(q)$, and refer to it as $\delta(q, \sigma, n)$. If u is a node of σ of level $n \geq 0$, with label (Y_u, L_u, X_u), then calculate $e = \bigwedge_{h \in Y_u} \text{dist}(h, \delta(\text{Last}(h), \sigma, n))$, where* dist *is defined as in Definition 20. Write e as $e = L \wedge X \wedge Y$, where L is a conjunction of literals from $\mathcal{L}\text{it}(N_P)$, X is a conjunction of constraints from $\text{constr}(x, K, N_{cF})$, and Y is a conjunction of $n+1$-histories. Then $L_u = \{\ell \in \mathcal{L}\text{it}(N_P) : \ell \text{ occurs in } L\}$; $X_u = \{x \in \text{constr}(x, K, N_{cF}) : x \text{ occurs in } X\}$; for all $d \in K$, such that the set $Z = \{hdq \in H_{n+1} : h \in H_n \wedge q \in Q \wedge (hdq \text{ occurs in } Y)\}$ is nonempty, and only for those d, u has a d-successor, ud, whose label (Y_{ud}, X_{ud}, L_{ud}) is such that $Y_{ud} = Z$.*

The uniformisation theorem for alternating automata, as defined in [58, 60], states that the existence of an accepting run of $\mathcal{A}$ is equivalent to the existence of an accepting uniform run of $\mathcal{A}$. However, the accepting condition in [58, 60] involves only

[5] In the case of a weak alternating automaton, if h is an (infinite) history, then from some point onwards, all the states occurring in h belong to the same element Q_i of the partition associated with the set of states. In [58], the element Q_i is referred to as the finality of h, and is denoted by $f(h)$. Q_i is the finality of h is equivalent to *Inf*$(h) \subseteq Q_i$. This observation will be made use of in the proof of Theorem 2.

the states repeated infinitely often in the branches of the run, and this is mainly due to the fact that the input alphabet is a simple set of symbols. In our case, as already explained, the input alphabet is the set $2^{N_P} \times \Theta(N_{cF}, \Delta_{\mathcal{D}_x})$, $\Theta(N_{cF}, \Delta_{\mathcal{D}_x})$ being the set of total functions $\theta : N_{cF} \rightarrow \Delta_{\mathcal{D}_x}$, associating with each concrete feature g, at each node of a run, a concrete value $\theta(g)$ from the spatial concrete domain $\Delta_{\mathcal{D}_x}$. In addition to the condition on the states infinitely often repeated on each branch of a run, one has also to consider the constraints on the values of the different concrete features at the different nodes of the run. The set of all such constraints, over the nodes of a run, gives birth to what we have named "CSP of the run" (Definition 21), which is a potentially infinite CSP. The uniformisation theorem was used in [58] to show that, a weak alternating automaton M of size (number of states) $|M|$ can be simulated by a (standard) nondeterministic Büchi automaton of size $|M|4^{|M|}$. We will define a "forgetful run" of a (weak) alternating automaton, or *f-run* for short, which, intuitively, is a 0-memory run, in the sense that it does not keep track of the histories of a branch leading to a node, but just of the states ending such histories —i.e., the states q such that, if the node is of level n, there exists an n-history h_n lying along the branch leading to the node, and such that $Last(h_n) = q$. We will then show that the existence of an accepting run of weak alternating automaton $\mathcal{A}$ on a tree t, is equivalent to the existence of an *f-run* of $\mathcal{A}$ on t. In particular, this will improve by an exponential factor the bound on the size of the nondeterministic Büchi automaton simulating a weak alternating automaton, which will be shown to be $2^{|M|}$, instead of the $|M|4^{|M|}$ bound in [58].

Definition 23 (Forgetful run). *Let $\mathcal{A}$ be an alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees, as defined in Definition 18, and t a k-ary $\Sigma(x, N_P, N_{cF})$-tree. A forgetful run, $f\text{-}r(\mathcal{A}, t)$, of $\mathcal{A}$ on t is a partial k-ary $\Sigma(2^Q, N_P, x, K, N_{cF})$-tree defined inductively as follows. For all directions $d \in K$, and for all nodes $u \in K^*$ of $f\text{-}r(\mathcal{A}, t)$, u has at most one outgoing edge labelled with d, and leading to the d-successor ud of u. The label $(Y_\epsilon, L_\epsilon, X_\epsilon)$ of the root belongs to $2^{\{q_0\}} \times c(2^{\mathcal{L}\text{it}(N_P)}) \times 2^{\text{constr}(x,K,N_{cF})}$ —in other words, $Y_\epsilon = \{q_0\}$. If u is a node of $f\text{-}r(\mathcal{A}, t)$ of level $n \geq 0$, with label (Y_u, L_u, X_u), then calculate $e = \bigwedge_{q \in Y_u} \delta(q)$. Write e in* dnf *as $e = \bigvee_{i=1}^{r} (L_i \wedge X_i \wedge Y_i)$, where the L_i's are conjunctions of literals from $\mathcal{L}\text{it}(N_P)$, the X_i's are conjunctions of constraints from* $\text{constr}(x, K, N_{cF})$, *and the Y_i's are conjunctions of direction-state pairs from $K \times Q$. Then there exists $i = 1 \ldots r$ such that $L_u = \{\ell \in \mathcal{L}\text{it}(N_P) : \ell \text{ occurs in } L_i\}$; $X_u = \{x \in \text{constr}(x, K, N_{cF}) : x \text{ occurs in } X_i\}$; for all $d \in K$, such that the set $Y = \{q \in Q : (d, q) \text{ occurs in } Y_i)\}$ is nonempty, and only for those d, u has a d-successor, ud, whose label (Y_{ud}, X_{ud}, L_{ud}) is such that $Y_{ud} = Y$; and the label $t(u) = (\mathcal{P}_u, \theta_u) \in 2^{N_P} \times \Theta(N_{cF}, \Delta_{\mathcal{D}_x})$ of the node u of the input tree t verifies the following, where, given a node v in t, the notation θ_v consists of the function $\theta_v : N_{cF} \rightarrow \Delta_{\mathcal{D}_x}$ which is the second argument of $t(v)$:*

- *for all $A \in N_P$: if $A \in L_u$ then $A \in \mathcal{P}_u$; and if $\neg A \in L_u$ then $A \notin \mathcal{P}_u$ (the elements A of N_P such that, neither A nor $\neg A$ occur in L_u, may or may not occur in $\mathcal{P}_u$);*
- *if x binary, for all $P(d_{i_1} \ldots d_{i_n} g_1, d_{j_1} \ldots d_{j_m} g_2)$ appearing in X_u, $P(\theta_{ud_{i_1} \ldots d_{i_n}}(g_1), \theta_{ud_{j_1} \ldots d_{j_m}}(g_2))$ holds. In other words, the value of the concrete feature g_1 at the $d_{i_1} \ldots d_{i_n}$-successor of u in t, on the one hand, and the value of the concrete feature g_2 at the $d_{j_1} \ldots d_{j_m}$-successor of u in t, on the other hand, are related by the x relation P.*

- *similarly, if x ternary, for all $P(d_{i_1}\dots d_{i_n}g_1, d_{j_1}\dots d_{j_m}g_2, d_{l_1}\dots d_{l_p}g_3)$ appearing in X_u, $P(\theta_{ud_{i_1}\dots d_{i_n}}(g_1), \theta_{ud_{j_1}\dots d_{j_m}}(g_2), \theta_{ud_{l_1}\dots d_{l_p}}(g_3))$ holds.*

A partial k-ary $\Sigma(2^Q, N_P, x, K, N_{cF})$-tree σ is an f-run *of $\mathcal{A}$ if there exists a k-ary $\Sigma(x, N_P, N_{cF})$-tree t such that σ is an* f-run *of $\mathcal{A}$ on t.*

A run σ can give rise to one and only one *f-run*, σ', which is obtained by replacing, in the argument $Y_u \subset H_{<\infty}$ of the label of a node u of σ, of, say, level n, each n-history h_n by $Last(h_n)$. We say that the run σ is the generator of the *f-run* σ', and denote this by $\sigma = gen(\sigma')$. The label $(Y_u^\sigma, L_u^\sigma, X_u^\sigma)$ of a node u in σ, and the label $(Y_u^{\sigma'}, L_u^{\sigma'}, X_u^{\sigma'})$ of the same node but in σ', verify $Y_u^{\sigma'} = \{Last(h) : h \in Y_u^\sigma\}$, $L_u^{\sigma'} = L_u^\sigma$, and $X_u^{\sigma'} = X_u^\sigma$.

Let σ be an *f-run*. We define a forgetful n-history, $n \geq 1$, of σ, or f-n-history of σ for short, as an $n+1$-length word $u = \{q_0\}v$ over the alphabet 2^Q —$v \in (2^Q)^n$; and a forgetful history, or f-history for short, as an ω-word from $\{q_0\}(2^Q)^\omega$. The f-0-history of σ is simply $\{q_0\}$. The f-0-history $\{q_0\}$ lies along the 0-branch reducing to the root of σ. An f-n-history $\Gamma_0 \dots \Gamma_n \in (2^Q)^{n+1}$ lies along an n-branch β_n of σ if, for all nodes u of β_n, of level i, the first argument Y_u of the label $\sigma(u)$ of u verifies $Y_u = \Gamma_i$. An f-history h lies along a branch β of σ if, for all $n \geq 0$, the f-n-history consisting of the $n+1$-length prefix of h lies along the n-branch consisting of the first n edges of β. *Inf*(h) is the set of subsets of Q infinitely often repeated in h. For all (infinite) branch β of σ, there is one and only one f-history lying along β, which we refer to as *fh*(β). A branch β of σ is accepting if the union of the elements of *Inf*(*fh*(β)) is a subset of the union of elements of $\mathcal{F}$; i.e., if $\bigcup_{S \in Inf(fh(\beta))} S \subseteq \bigcup_{F \in \mathcal{F}} F$. The *f-run* σ is accepting iff all its branches are accepting. We also refer to *Inf*(*fh*(β)) as *Inf*(β).

Theorem 2. *Let $\mathcal{A}$ be a weak alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees, and t a k-ary $\Sigma(x, N_P, N_{cF})$-tree. There exists an accepting run of $\mathcal{A}$ on t iff there exists an accepting* f-run *of $\mathcal{A}$ on t.*

Proof: We show the following. Given an *f-run* σ of $\mathcal{A}$ on t, σ is accepting iff the run *gen*(σ) is accepting. The CSP of σ and the CSP of *gen*(σ) are the same. So we only need look at the accepting subcondition related to the states infinitely often repeated.

Suppose, to start with, that *gen*(σ) is accepting. Consider a branch β_g in *gen*(σ), and the correponding branch β of σ. β_g is accepting: the states infinitely often repeated in a history h lying along β_g are elements of a set $Q_h \in \mathcal{F}$. The union of the subsets of Q infinitely often repeated in the history *fh*(β) of σ, is a subset of the union of all such Q_h over the histories lying along β_g: $\bigcup_{S \in Inf(fh(\beta))} S \subseteq \bigcup_{h \text{ lies along } \beta_g} Q_h$. According to our definition of an accepting branch of an *f-run*, β is clearly accepting, since, for all h lying along β_g, Q_h belongs to $\mathcal{F}$. It follows that σ is accepting.

Conversely, suppose that σ is accepting. We need to show that *gen*(σ) is accepting. Consider a branch β_g of *gen*(σ), and the corresponding branch β in σ. Let Q^f be the union of the subsets of Q infinitely often repeated in *Inf*(*fh*(β)). β being accepting, Q^f is a subset of the union of all elements of $\mathcal{F}$:

$$Q^f \subseteq \bigcup_{F \in \mathcal{F}} F \tag{2}$$

The key point now is that, the set Q^f can also be seen as the union of the *Inf*(h)'s over the histories h lying along β_g:

$$Q^f = \bigcup_{h \text{ lies along } \beta_g} \mathit{Inf}(h) \tag{3}$$

From (2) and (3), we get:

$$\bigcup_{h \text{ lies along } \beta_g} \mathit{Inf}(h) \subseteq \bigcup_{F \in \mathcal{F}} F \tag{4}$$

Suppose now that there exists a history h_* lying along β_g, which is not accepting. In concrete terms, this would mean that:

$$\forall F \in \mathcal{F}, \mathit{Inf}(h_*) \nsubseteq F \tag{5}$$

Given the partial order $\geq$ associated with the partition $Q = \bigcup_{i=1}^{n} Q_i$ of the set of states Q of $\mathcal{A}$, and the decreasing property of the transition function δ, that, given $q \in Q_i$ and $q' \in Q_j$, if $q' \in \delta(q)$ then $Q_i \geq Q_j$, it follows that the set *Inf*(h) of states infinitely often repeated in a history h, should be a subset of some element Q_i of the partition, $i = 1 \ldots n$. For h_*, in particular, we should have:

$$\exists i = 1 \ldots n, \mathit{Inf}(h_*) \subseteq Q_i \tag{6}$$

The conjunction of (6) and (5) implies that, *Inf*(h_*) is disjoint from each of the elements in $\mathcal{F}$:

$$\forall F \in \mathcal{F}, \mathit{Inf}(h_*) \cap F = \emptyset \tag{7}$$

(7) clearly contredicts (4). All histories lying along β_g are thus accepting, and the run $gen(\sigma)$ is accepting. ■

A Büchi nondeterministic automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees can be thought of as a special case of a Büchi alternating automaton: as one that sends at most one copy per direction in a run. In other words, as a Büchi alternating automaton with the property that, there is one and only one history lying on any branch of any run of the automaton.

Definition 24 (Büchi nondeterministic automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees). *A Büchi nondeterministic automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees is a tuple $\mathcal{B} = (Q, K, \mathcal{L}\text{it}(N_P), \text{constr}(x, K, N_{cF}), \Sigma(x, N_P, N_{cF}), \delta, q_0, q_\#, \mathcal{F})$, where Q is a finite set of states; $K = \{d_1, \ldots, d_k\}$ ($k \geq 1$) is a set of directions; $\mathcal{L}\text{it}(N_P)$, $\text{constr}(x, K, N_{cF})$ and $\Sigma(x, N_P, N_{cF})$ are as in Definition 18; $q_0 \in Q$ is the initial state; $q_\# \in Q$ is a state indicating that no copy has to be sent in the corresponding direction; $\mathcal{F}$ defines the acceptance condition; and $\delta : Q \to \mathcal{P}(2^{\mathcal{L}\text{it}(N_P)} \times 2^{\text{constr}(x,K,N_{cF})} \times (Q \cup \{q_\#\})^k)$ is the transition function.*

Definition 25 (Run of a Büchi automaton). *Let $\mathcal{B}$ be a Büchi nondeterministic automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees, as defined in Definition 24, and t a k-ary*

$\Sigma(x, N_P, N_{cF})$-tree. A run, $r(\mathcal{B}, t)$, of $\mathcal{B}$ on t is a partial k-ary $\Sigma(Q, N_P, x, K, N_{cF})$-tree[6] *defined inductively as follows. For all directions $d \in K$, and for all nodes $u \in K^*$ of $r(\mathcal{B}, t)$, u has at most one outgoing edge labelled with d, and leading to the d-successor ud of u. The label $(Y_\epsilon, L_\epsilon, X_\epsilon)$ of the root belongs to $\{q_0\} \times c(2^{\mathcal{L}\text{it}(N_P)}) \times 2^{\text{constr}(x,K,N_{cF})}$ —in other words, $Y_\epsilon = q_0$. If u is a node of $r(\mathcal{B}, t)$ of level $n \geq 0$, with label (Y_u, L_u, X_u), then let $e = \delta(Y_u) \subseteq 2^{\mathcal{L}\text{it}(N_P)} \times 2^{\text{constr}(x,K,N_{cF})} \times (Q \cup \{q_\#\})^k$. Then there exists $(L, X, (q_{i_1}, \ldots, q_{i_k})) \in \delta(Y_u)$ such that $L_u = L$; $X_u = X$; for all $j = 1 \ldots k$, such that $q_{i_j} \neq q_\#$, and only for those j, u has a d_j-successor, ud_j, whose label $(Y_{ud_j}, X_{ud_j}, L_{ud_j})$ is such that $Y_{ud_j} = q_{i_j}$; and the label $t(u) = (\mathcal{P}_u, \theta_u) \in 2^{N_P} \times \Theta(N_{cF}, \Delta_{\mathcal{D}_x})$ of the node u of the input tree t verifies the following, where, given a node v in t, the notation θ_v consists of the function $\theta_v : N_{cF} \rightarrow \Delta_{\mathcal{D}_x}$ which is the second argument of $t(v)$:*

- *for all $A \in N_P$: if $A \in L_u$ then $A \in \mathcal{P}_u$; and if $\neg A \in L_u$ then $A \notin \mathcal{P}_u$ (the elements A of N_P such that, neither A nor $\neg A$ occur in L_u, may or may not occur in $\mathcal{P}_u$);*
- *if x binary, for all $P(d_{i_1} \ldots d_{i_n} g_1, d_{j_1} \ldots d_{j_m} g_2)$ appearing in X_u, $P(\theta_{ud_{i_1} \ldots d_{i_n}}(g_1), \theta_{ud_{j_1} \ldots d_{j_m}}(g_2))$ holds. In other words, the value of the concrete feature g_1 at the $d_{i_1} \ldots d_{i_n}$-successor of u in t, on the one hand, and the value of the concrete feature g_2 at the $d_{j_1} \ldots d_{j_m}$-successor of u in t, on the other hand, are related by the x relation P.*
- *similarly, if x ternary, for all $P(d_{i_1} \ldots d_{i_n} g_1, d_{j_1} \ldots d_{j_m} g_2, d_{l_1} \ldots d_{l_p} g_3)$ appearing in X_u, $P(\theta_{ud_{i_1} \ldots d_{i_n}}(g_1), \theta_{ud_{j_1} \ldots d_{j_m}}(g_2), \theta_{ud_{l_1} \ldots d_{l_p}}(g_3))$ holds.*

A partial k-ary $\Sigma(Q, N_P, x, K, N_{cF})$-tree σ is a run of $\mathcal{B}$ if there exists a k-ary $\Sigma(x, N_P, N_{cF})$-tree t such that σ is a run of $\mathcal{B}$ on t.

An n-branch and a branch of a run of a Büchi nondeterministic automaton are defined as in the alternating case. Given an n-branch β, one and only one n-history lies along β, which is $h = q_0 d_{i_1} q_{i_1} \ldots d_{i_n} q_{i_n} \in \{q_0\}(KQ)^n$, such that, the node given by K-*proj*(h) is the terminal node of the n-branch, and q_{i_j}, $j = 1 \ldots n$, is the first argument of the label of the j-th node of the n-branch. An (infinite) history $h = q_0 d_{i_1} q_{i_1} \ldots d_{i_n} q_{i_n} \ldots \in \{q_0\}(KQ)^\omega$ lies along a branch β if, for every $n \geq 1$, the prefix of h consisting of the n-history $q_0 d_{i_1} q_{i_1} \ldots d_{i_n} q_{i_n}$ lies along the n-branch β_n consisting of the first n edges of β. A history h is accepting if $\mathit{Inf}(h) \cap \mathcal{F} \neq \emptyset$. A branch is accepting if the history lying along it is accepting. A run is accepting if all its branches are accepting. The following corollary is a direct consequence of Theorem 2.

Corollary 1. *Let $\mathcal{A}$ be a weak alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees, and Q the set of states of $\mathcal{A}$. There exists a Büchi nondeterministic automaton simulating $\mathcal{A}$, with a number of states bounded by $2^{|Q|}$, $|Q|$ being the size (number of states) of $\mathcal{A}$.*

Proof: Let $\mathcal{A} = (\mathcal{L}(\mathcal{L}it(N_P) \cup constr(x, K, N_{cF}) \cup K \times Q), \Sigma(x, N_P, N_{cF}), \delta, q_0, \mathcal{F})$ be an alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees, as defined in Definition 18, and suppose that $\mathcal{A}$ is weak (Definition 19). From the proof of Theorem 2, the following

[6] $\Sigma(Q, N_P, x, K, N_{cF}) = Q \times c(2^{\mathcal{L}it(N_P)}) \times 2^{constr(x,K,N_{cF})}$.

Büchi nondeterministic automaton, $\mathcal{B} = (2^Q, K, \mathcal{L}it(N_P), constr(x, K, N_{cF}), \Sigma(x, N_P, N_{cF}), \delta_\mathcal{B}, \{q_0\}, \{q_\#\}, \mathcal{F}_\mathcal{B})$, simulates $\mathcal{A}$. In particular, the set of states of $\mathcal{B}$ is the set of subsets of Q, and the initial state of $\mathcal{B}$ is the singleton subset $\{q_0\}$ of Q. The only parameters that are not obvious are the transition function $\delta_\mathcal{B}$ and the set $\mathcal{F}_\mathcal{B}$ providing the acceptance condition. For all $Q_1 \in 2^Q$, $\delta_\mathcal{B}(Q_1)$ is obtained as follows. An element $(L, X, (Q_{i_1}, \ldots, Q_{i_k}))$ of $2^{\mathcal{L}it(N_P)} \times 2^{constr(x,K,N_{cF})} \times (2^Q \cup \{\{q_\#\}\})^k$ belongs to $\delta_\mathcal{B}(Q_1)$ iff there exists an *f-run* σ of $\mathcal{A}$, and a node u of σ, so that, the label (Y_u, X_u, L_u) satisfies $Y_u = Q_1$, $X_u = X$ and $L_u = L$, and for all $j = 1 \ldots k$, such that $Q_{i_j} \neq \{q_\#\}$, and only for those j, u has a d_j-successor, ud_j, whose label $(Y_{ud_j}, X_{ud_j}, L_{ud_j})$ is such that $Y_{ud_j} = Q_{i_j}$. The set $\mathcal{F}_\mathcal{B}$ is $\mathcal{F}_\mathcal{B} = \bigcup_{F \in \mathcal{F}} F$. The condition for a history h to be accepting is not $Inf(h) \cap \mathcal{F}_\mathcal{B} \neq \emptyset$, rather $\bigcup_{Q_1 \in Inf(h)} Q_1 \subseteq \mathcal{F}_\mathcal{B}$ (the explanation lies in the proof of Theorem 2). ■

Let σ be an *f-run*. Given a branch β of σ, $Inf(\beta)$ denotes, as we have seen, the set of subsets of Q infinitely often repeated in β: in other words, the set of $Q_1 \in 2^Q$ such that, there exist infinitely many nodes u of β so that, the label $\sigma(u) = (Y_u, X_u, L_u)$ of u verifies $Y_u = Q_1$. By $Inf(\sigma)$, we denote the union of all $Inf(\beta)$ along the branches β of σ: $Inf(\sigma) = \bigcup_{\beta \text{ branch of } \sigma} Inf(\beta)$. The following corollary is also a direct consequence of Theorem 2.

Corollary 2. *Let $\mathcal{A}$ be a weak alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees, and t a k-ary $\Sigma(x, N_P, N_{cF})$-tree. An* f-run *σ of $\mathcal{A}$ on t is accepting iff the union of the elements of* Inf(σ) *is a subset of the union of the elements in $\mathcal{F}$; in other words, iff* $\bigcup_{Q_1 \in \text{Inf}(\sigma)} Q_1 \subseteq \bigcup_{F \in \mathcal{F}} F$.

Proof: Let σ be an *f-run* as described in the theorem. Suppose, to start with, that σ is accepting. As a consequence, for all branches β of σ, we have $\bigcup_{Q_1 \in Inf(\beta)} Q_1 \subseteq \bigcup_{F \in \mathcal{F}} F$. This straightforwardly leads to $\bigcup_{Q_1 \in Inf(\sigma)} Q_1 \subseteq \bigcup_{F \in \mathcal{F}} F$. To show the other direction of the corollary, suppose that $\bigcup_{Q_1 \in Inf(\sigma)} Q_1 \subseteq \bigcup_{F \in \mathcal{F}} F$. As an immediate consequence, for all branches β of σ, we have $\bigcup_{Q_1 \in Inf(\beta)} Q_1 \subseteq \bigcup_{F \in \mathcal{F}} F$, which clearly means that the *f-run* σ is accepting. ■

Deciding whether a standard Büchi nondeterministic automaton on k-ary Σ-trees (see, for instance, [65, 66]) accepts a nonempty language is trivial. The intuitive idea is to build a partial run, whose size is linear in the number of states, and with the property that no state appears more than once in the label of an internal node, though it may appear more than once at the level of leaves. The kind of Büchi automata we are dealing with is more complicated, due mainly to the use of feature chains to relate values of different concrete features at different nodes of the run, which gives rise to what we have named "CSP of a run", which is potentially infinite. The rest of the section will show how to extend the method, so that it can handle the emptiness problem of this new kind of Büchi automata. Thanks to Theorem 2 and Corollary 1, we transform the problem into how to check whether an *f-run* of a weak alternating automaton is accepting. The method is constructive and can easily be used to derive effective tableaux methods for the problem of deciding satisfiability of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept w.r.t. an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox. Some additional vocabulary is needed.

Definition 26 (prefix and lexicographic order). *Let $\Sigma = \{a_1, \dots, a_n\}$ be an ordered alphabet, with $a_1 < a_2 < \cdots < a_n$, and $u, v \in \Sigma^*$. The relations "u is prefix of v", denoted by* $\text{pfx}(u,v)$, *and "u is lexicographically smaller than v", denoted by $u \leq_\ell v$, are defined in the following obvious manner:*

1. $\text{pfx}(u,v)$ *iff $v = uw$, for some $w \in \Sigma^*$*
2. $u \leq_\ell v$ *iff, either* $\text{pfx}(u,v)$*; or $u = w_1 a w_2$ and $v = w_1 b w_3$, for some $w_1, w_2, w_3 \in \Sigma^*$ and $a, b \in \Sigma$, with $a < b$.*

We will also need the derived relations "u is a strict prefix of v", "u is lexicographically strictly smaller than v", and "u and v are incomparable", which we denote, respectively, by *s-pfx*(u,v), $u <_\ell v$ and *incp*(u,v):

1. *s-pfx*(u,v) iff *pfx*(u,v) and $u \neq v$
2. $u <_\ell v$ iff $u \leq_\ell v$ and $u \neq v$
3. *incp*(u,v) iff $\neg$*pfx*(u,v) and $\neg$*pfx*(v,u)

Definition 27 (subtree). *Let $K = \{d_1, \dots, d_k\}$ be a set of k directions, t a partial k-ary Σ-tree, and $u \in K^*$ a node of t. The subtree of t at u, denoted t/u, is the partial k-ary Σ-tree t', whose nodes are of the form v, so that uv is a node of t, and, for all such nodes, $t'(v) = t(uv)$ —i.e., the label of v in t', is the same as the one of uv in t.*

Definition 28 (substitution). *Let $K = \{d_1, \dots, d_k\}$ be a set of k directions, t and t' two partial k-ary Σ-trees, and $u \in K^*$ a node of t. The substitution of t' to the subtree of t at u, or u-substitution of t' in t, denoted $t(u \leftarrow t')$, is the partial k-ary Σ-tree t'' such that, the nodes are of the form v, with v node of t of which u is not a prefix, or of the form uv, with v a node of t'. The label $t''(v)$ of v in t'' is defined as follows:* $t''(v) = \begin{cases} t'(w) & \text{if } v = uw, \text{ for some node } w \text{ of } t', \\ t(v) & \text{otherwise} \end{cases}$

Definition 29 (cut). *Let $K = \{d_1, \dots, d_k\}$ be a set of k directions, t a partial k-ary Σ-tree, and $u \in K^*$ a node of t. The cut in t of the subtree at u, or u-cut in t, denoted $c(u,t)$, is the partial k-ary Σ-tree t' whose nodes are those nodes v of t of which u is not a strict prefix —i.e., such that* $\neg\text{s-pfx}(u,v)$*. The label $t'(v)$ of any node v in t' is the same as $t(v)$, the label of the same node in t.*

The last step of our walk towards decidability of the satisfiability of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept w.r.t. an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox, is to show how to handle the CSP of an *f-run*, which is potentially infinite. For the purpose, we need another kind of *f-run*, regular *f-run*, which is based on a function *back*: given an *f-run* σ and a node u of σ, *back*(σ,u) consists, intuitively, of those constraints that are still unfulfilled at

Input: an accepting *f-run* σ of a weak alternating automaton $\mathcal{A}$.
Output: a finite representation, t, of a regular *f-run* generated from σ.
Initialise t to σ: $t \leftarrow \sigma$;
Initially, no node of t is marked;
repeat while possible{
  Let u be the smallest node of t such that there exists a non marked node v, so that $<_\ell(u,v)$ <u>and</u> $Y_u = Y_v$ <u>and</u> $back(\sigma,u) = back(\sigma,v)$;
  choose v as small as possible, w.r.t. to the lexicographic order $\leq_\ell$;
  if $\neg pfx(u,v)${
    $t \leftarrow c(v,t)$;
    $back\text{-}node(v) \leftarrow u$;
    mark v;
    }
  else <u>% $pfx(u,v)$ %</u>
    if nodes w between u and v (i.e., so that $\leq_\ell(u,w) \wedge \leq_\ell(w,v)$) all verify $Y_w \subseteq \bigcup_{F\in\mathcal{F}} F$ then{
      $t \leftarrow c(v,t)$;
      $back\text{-}node(v) \leftarrow u$;
      mark v;
      }
    else{
      $t' \leftarrow t/v$;
      $t \leftarrow t(u \leftarrow t')$;
      }
  } <u>% end **repeat** %</u>

Fig. 6. The order $d_1 < \ldots < d_k$ is assumed on the directions in K.

u, and which were solicited at nodes v that are prefixes of u. Formally, the function is defined as follows for the case of x being binary: $back(\sigma,u) = back_l(\sigma,u) \cup back_r(\sigma,u)$, with

$$\begin{aligned}
back_l(\sigma,u) = \{(n, P(d_{i_1}\ldots d_{i_n}v_1g_1, v_2g_2)) : (\exists u_1 \in K^*)(u = u_1d_{i_1}\ldots d_{i_n} \wedge \\
P(d_{i_1}\ldots d_{i_n}v_1g_1, v_2g_2) \in X_{u_1})\} \\
back_r(\sigma,u) = \{(n, P(v_1g_1, d_{i_1}\ldots d_{i_n}v_2g_2)) : (\exists u_1 \in K^*)(u = u_1d_{i_1}\ldots d_{i_n} \wedge \\
P(v_1g_1, d_{i_1}\ldots d_{i_n}v_2g_2) \in X_{u_1})\}
\end{aligned}$$

Definition 30 (regular *f-run*). *Let $\mathcal{A}$ be a weak alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees, as defined in Definition 18, and σ an* f-run *of $\mathcal{A}$. σ is regular if, for all nodes u and v of σ verifying* back$(\sigma,u) =$ back(σ,v)*, and whose labels $\sigma(u) = (Y_u, L_u, X_u)$ and $\sigma(v) = (Y_v, L_v, X_v)$ verify $Y_u = Y_v$, the following holds:*

1. *$L_u = L_v$;*
2. *$X_u = X_v$;*
3. *for all $d \in K$, u has a d-successor iff v has a d-successor; and*
4. *for all $d \in K$ such that, each of u and d has a d-successor, it is the case that $Y_{ud} = Y_{vd}$.*

Theorem 3. *Let $\mathcal{A}$ be a weak alternating automaton on k-ary $\Sigma(x, N_P, N_{cF})$-trees, and t a k-ary $\Sigma(x, N_P, N_{cF})$-tree. There exists an accepting* f-run *of $\mathcal{A}$ on t iff there exists an accepting regular* f-run *of $\mathcal{A}$ on t.*

Proof: A regular *f-run* is a particular *f-run*, which means that the existence of an accepting regular *f-run* implies the existence of an accepting *f-run*. To show the other direction of the equivalence, suppose the existence of an accepting *f-run*, say σ. From σ, we first build a finite partial k-ary $\Sigma(2^{H_{<\infty}}, N_P, x, K, N_{cF})$-tree, t. We then show how to use t to get an accepting regular *f-run* of $\mathcal{A}$. The tree t is built by the procedure of Figure 6. The details of the procedure are as follows:

- u and v are chosen so that $<_\ell(u, v)$ and $Y_u = Y_v$ and $back(\sigma, u) = back(\sigma, v)$ (line (6))
- if u is not prefix of v: given that $back(\sigma, u) = back(\sigma, v)$, we can substitute the subtree of t at u to the subtree of t at v, and get a run with all branches accepting, and with a global CSP consistent. The procedure, however, does not do the substitution. Instead, it cuts the subtree at v, and marks u as the successor of v, information which will be used in the building of the accepting regular run (lines (9)-(10)-(11))
- if u is a (strict) prefix of v then there are two possibilities:
  - \# if all nodes w between u and v are so that $Y_w \subseteq \bigcup_{F \in \mathcal{F}} F$ (line (14)) then cutting t at v, and then repeating the subtree at u of the resulting tree, will lead to an accepting *f-run*, again thanks to $back(\sigma, u) = back(\sigma, v)$. What the procedure does in this case: it cuts the subtree at v, and sets v as a repetition of internal node u (lines (15)-(16)-(17))
  - \# the other possibility corresponds to the case when the segment $[u, v]$ does contain nodes w which do not have the property $Y_w \subseteq \bigcup_{F \in \mathcal{F}} F$. The procedure shortens the distance to segments $[u, v]$ with all nodes w verifying the property $Y_w \subseteq \bigcup_{F \in \mathcal{F}} F$ (lines (20)-(21)).

The output tree t of the procedure of Figure 6 is so that, each marked node, v, is a leaf and is so that, there is one and only one internal node, u, of t such that $Y_u = Y_v$ and $back(\sigma, u) = back(\sigma, v)$. For each such node v, we refer to the corresponding internal node u as i_v, and to the subtree of t at u as t/i_v. From t, we now build an accepting regular *f-run*, σ, which, intuitively, consists of "pasting" infinitely many times such subtrees at the matching leaves.

Step 0: $\sigma_0 \leftarrow t$
Step 1:
  initialise σ_1 to σ_0: $\sigma_1 \leftarrow \sigma_0$
  repeat while possible{
    consider a marked node v_1 of t
    if σ_0 and σ_1 have a (same) leaf node v_2 which is marked and so that $Y_{v_2} = Y_{v_1}${
      $\sigma_1 \leftarrow \sigma_1(v_2 \leftarrow t/i_{v_1})$
      if a leaf $i_{v_1}v_3$ of t is marked then mark the corresponding leaf v_2v_3 of σ_1
      }

```
    }
Step n (n ≥ 2):
  initialise σ_n to σ_{n−1}: σ_n ← σ_{n−1}
  repeat while possible{
    consider a marked node v_1 of t
    if σ_{n−1} and σ_n have a (same) leaf node v_2 which is marked and so that
  Y_{v_2} = Y_{v_1}{
      σ_n ← σ_n(v_2 ← t/i_{v_1})
      if a leaf i_{v_1}v_3 of t is marked then mark the corresponding leaf v_2v_3 of σ_n
      }
    }
```

The accepting regular *f-run* σ we are looking for now, is nothing else than the partial k-ary Σ-tree σ_n when n tends to $+\infty$. By construction, σ is an *f-run* of $\mathcal{A}$. Given that, in t, each marked node v verifying $\leq_\ell(i_v, v)$ is so that, the union of all Y_w, over the nodes w between i_v and v, is a subset of $\bigcup_{F\in\mathcal{F}} F$, all branches of σ are accepting. And, finally, given that, in t, each marked node v verifies $back(t,v) = back(t,i_v)$, the CSP of σ, $CSP(\sigma)$, is consistent. ■

The following corollary is a direct consequence of Theorem 3.

Corollary 3. *There exists a nondeterministic exponential-time algorithm deciding whether an $\mathcal{MTALC}(\mathcal{D}_x)$ concept is satisfiable w.r.t. an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox.*

Proof: The number of nodes of the output tree t of the procedure of Figure 6, that are not marked[7], is bounded by $2^{|Q|} \times \ell_{fc} \times 2^{n_c}$, where Q is the set of states of $\mathcal{A}$, and ℓ_{fc} and n_c are, respectively, the length of the longest K^*N_{cF}*-chain* and the number of constraints from $constr(x, K, N_{cF})$ appearing in the transition function σ of $\mathcal{A}$. We can thus in nondeterministic exponential-time build such a tree, if it exists, or report its inexistence, otherwise. ■

7.2 Associating a weak alternating automaton with the satisfiability of a concept w.r.t. a weakly cyclic TBox

We are now ready to describe how to effectively associate with the satisfiability, w.r.t. an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox $\mathcal{T}$, of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept C, a weak alternating automaton $\mathcal{A}_{C,\mathcal{T}}$, so that the set of models of C w.r.t. $\mathcal{T}$ coincides with the language accepted by $\mathcal{A}_{C,\mathcal{T}}$ —in particular, C is insatisfiable w.r.t. $\mathcal{T}$ iff the language accepted by $\mathcal{A}_{C,\mathcal{T}}$ is empty.

Definition 31. *Let $x \in \{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$, C an $\mathcal{MTALC}(\mathcal{D}_x)$ concept, $\mathcal{T}$ an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox, $T \oplus C$ the TBox T augmented with C, and B_i the initial defined concept of $T \oplus C$. With the satisfiability of C w.r.t. $\mathcal{T}$, we associate the weak alternating automaton $\mathcal{A}_{C,\mathcal{T}} = (\mathcal{L}(\mathcal{L}\text{it}(N_P) \cup \text{constr}(x, K, N_{cF}) \cup K \times Q), \Sigma(x, N_P, N_{cF}), \delta, q_0, \mathcal{F})$ on k-ary $\Sigma(x, N_P, N_{cF})$-trees such that C is satisfiable w.r.t. $\mathcal{T}$ iff the language $\mathcal{L}(\mathcal{A}_{C,\mathcal{T}})$ accepted by $\mathcal{A}_{C,\mathcal{T}}$ is nonempty. The parameters of the automaton are as follows:*

[7] The others are leaf nodes, and are repetitions of unmarked, internal nodes.

1. *$N_P = \text{pConcepts}(C, \mathcal{T})$, $N_{cF} = \text{cFeatures}(C, \mathcal{T})$, $Q = \text{dConcepts}(C, \mathcal{T})$, $q_0 = B_i$*
2. *K is the set of concepts appearing as arguments in the branching tuple of C w.r.t. $\mathcal{T}$: $K = \{d_1, \ldots, d_n : (d_1, \ldots, d_n) = \text{bt}(C, \mathcal{T})\}$ (Definition 15)*
3. *$\delta(B)$ is obtained from the axiom $B \doteq E$ in $(T \oplus C)^*$ defining B, as follows. E is of the form $\{S_1, \ldots, S_n\}$, with $S = S_{prop} \cup S_{csp} \cup S_\exists$, for all $S \in \{S_1, \ldots, S_n\}$. We transform $S_\exists$ into $S'_\exists = \{(\exists R.D, D) : \exists R.D \in S_\exists \cap \text{reConcepts}(C, \mathcal{T})\} \cup \{(f, D) : \exists f.D \in S_\exists \cap \text{feConcepts}(C, \mathcal{T})\}$. We transform S_{csp} into $S'_{csp} = \{P(u_1, u_2) : u_1, u_2 \in K^* N_{cF}$ and $\exists(u_1)(u_2).P \in S_{csp}\}$, if x binary, and into $S'_{csp} = \{P(u_1, u_2, u_3) : u_1, u_2, u_3 \in K^* N_{cF}$ and $\exists(u_1)(u_2)(u_3).P \in S_{csp}\}$, if x ternary. We get $S' = S_{prop} \cup S'_{csp} \cup S'_\exists$. Finally, $\delta(B) = \bigvee_{S \in E} \bigwedge_{X \in S'} X$.*
4. *The remaining part is to determine the right partition of Q; the right partial order $\geq$ on the elements of the partition; and those elements of the partition that are accepting, i.e., constituting the set $\mathcal{F}$ (see Definition 19). Without loss of generality, we assume that the axioms of $\mathcal{T}$ are given in the form $B \doteq \{S_1, \ldots, S_n\}$, with $S = S_{prop} \cup S_{csp} \cup S_\exists \cup S_\forall$, for all $S \in \{S_1, \ldots, S_n\}$. In other words, we suppose that in $T \oplus C$, each of the axioms, $B \doteq E$, has gone through the process of computing the first DNF,* dnf1, *of the right-hand side, E. Going from* dnf1 *to* dnf2 *involves transforming $S = S_{prop} \cup S_{csp} \cup S_\exists \cup S_\forall$ into $S^f = S_{prop} \cup S_{csp} \cup S^f_\exists$, with each $\exists R.C$ in $S^f_\exists$ verifying $C = C_1 \sqcap \ldots \sqcap C_{m_1} \sqcap C_{m_1+1} \sqcap \ldots \sqcap C_{m_2}$, such that $\{\exists R.C_1, \ldots, \exists R.C_{m_1}, \forall R.C_{m_1+1}, \ldots, \forall R.C_{m_2}\} \subseteq S_\exists \cup S_\forall$ ($m_1 \geq 1$, if R is functional, and $m_1 = 1$, otherwise). This decreasing property implies that no defined concept in $(T \oplus C)^*$, other than the ones in $T \oplus C$, "directly uses" itself. However, if in $T \oplus C$, a defined concept B "uses" (specifically, "directly uses") itself, it might give birth in $(T \oplus C)^*$ to a new defined concept which (directly) "uses" B, leading thus to a cycle of length strictly greater than one —all defined concepts of such a cycle will constitute one element of the partition. The right partition is computed by the procedure of Figure 7. The right partial order, $\geq$, is as follows: given two elements Q_i and Q_j of the partition, $Q_i \geq Q_j$ if there exist $B_1 \in Q_i$ and $B_2 \in Q_j$, such that B_1 "uses" B_2 in $(T \oplus C)^*$. The set $\mathcal{F}$ of accepting subsets of Q is the set of all elements Q_i of the partition, such that Q_i contains no state consisting of an eventuality defined concept of $(T \oplus C)^*$ —in other words, all states in such a Q_i are noneventuality defined concepts of $(T \oplus C)^*$.*

8 Discussion 1

Theorem 3 and Corollary 3 provide a tableaux-like procedure for the satisfiability of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept w.r.t. an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox. To understand how such a procedure will work in practice, suppose that the $\mathcal{RCC}8$-like spatial constraints at the different nodes of the output tree t of the procedure of Figure 6, are not processed while the tree is being built. In other words, we suppose that we first build the tree, and then process the (global) CSP of the tree, which is given by the sets S_{csp}, over the sets S labelling the nodes of t. The building of t is clear from Theorem 3 and Corollary 3, which, among other things, offer a bound on the size of t, as well as what is referred to in standard tableaux-like algorithms as a blocking condition. If such a tree is successfully built, according to Theorem 3, it can be extended to a full *f-run* which satisfies the accepting subcondition related to the states infinitely often

Input: the closure $(T \oplus C)^*$ of a TBox $\mathcal{T}$ augmented with a concept C, $T \oplus C$
Output: partition of the set of defined concepts in $(T \oplus C)^*$
Initially, no defined concept of $(T \oplus C)^*$ is marked;
$k = 1$;
while($(T \oplus C)^*$ contains defined concepts that are not marked){
 consider a non marked defined concept B_1 from $(T \oplus C)^*$;
 mark B_1;
 $USES\_B_1 \leftarrow \bigcup_{B_2 \text{ "uses" } B_1} \{B_2\}$;
 if $B_1 \in USES\_B_1${
  $PARTITION[k] \leftarrow USES\_B_1$;
  mark all defined concepts of $(T \oplus C)^*$ occurring in $USES\_B_1$;
  }
 else $PARTITION[k] \leftarrow \{B_1\}$;
 $k \leftarrow k + 1$;
 }

Fig. 7. Partition of the set of defined concepts in the closure $(T \oplus C)^*$ of a TBox $\mathcal{T}$ augmented with a concept C, $T \oplus C$.

repeated in the branches. We now need to check the other accepting subcondition, which is the consistency of the CSP of t, whose definition can be derived from that of the (global) CSP of a run, as follows:

1. For all nodes of t that are not marked, and for all directions $d \in K$ such that u has a d-successor v in t, the non marked d-successor of u in t is v, if v is not a marked node, and i_v, otherwise. If u has a $d_{i_1} \dots d_{i_n}$-successor in t, $n \geq 2$, then the non marked $d_{i_1} \dots d_{i_n}$-successor of u in t is the non marked $d_{i_2} \dots d_{i_n}$-successor of v in t, where v is the non marked d_{i_1}-successor of u in t.
2. for all nodes v of t, of label $t(v) = (Y_v, L_v, X_v) \in 2^{H_{<\infty}} \times c(2^{\mathcal{L}it(N_P)}) \times 2^{constr(x,K,N_{cF})}$, the argument X_v gives rise to the CSP of t at v, $CSP_v(t)$, whose set of variables, $V_v(t)$, and set of constraints, $C_v(t)$, are defined as follows:
   (a) Initially, $V_v(t) = \emptyset$ and $C_v(t) = \emptyset$
   (b) for all K^*N_{cF}*-chains* $d_{i_1} \dots d_{i_n} g$ appearing in X_v, create, and add to $V_v(t)$, a variable $\langle w, g \rangle$, where w is the non marked $d_{i_1} \dots d_{i_n}$-successor of v in t
   (c) if x binary, for all $P(d_{i_1} \dots d_{i_n} g_1, d_{j_1} \dots d_{j_m} g_2)$ in X_v, add the constraint $P(\langle w_1, g_1 \rangle, \langle w_2, g_2 \rangle)$ to $C_v(t)$, where w_1 is the non marked $d_{i_1} \dots d_{i_n}$-successor of v in t, and w_2 the non marked $d_{j_1} \dots d_{j_m}$-successor of v in t
   (d) similarly, if x ternary, for all $P(d_{i_1} \dots d_{i_n} g_1, d_{j_1} \dots d_{j_m} g_2, d_{l_1} \dots d_{l_p} g_3)$ in X_v, add the constraint $P(\langle w_1, g_1 \rangle, \langle w_2, g_2 \rangle, \langle w_3, g_3 \rangle)$ to $C_v(t)$, where w_1, w_2 and w_3 are, respectively, the non marked $d_{i_1} \dots d_{i_n}$-successor, the non marked $d_{j_1} \dots d_{j_m}$-successor, and the non marked $d_{l_1} \dots d_{l_p}$-successor of v in t
3. the CSP of t, $CSP(t)$, is the CSP whose set of variables, $\mathcal{V}(t)$, and set of constraints, $\mathcal{C}(t)$, are defined as $\mathcal{V}(t) = \bigcup_{v \text{ node of } t} V_v(t)$ and $\mathcal{C}(t) = \bigcup_{v \text{ node of } t} C_v(t)$.

Contrary to the CSP of a run, which is potentially infinite, the CSP of t is finite, and can thus be checked for consistency using existing algorithms, as explained in

Subsection 3.3. In particular, if the constraints, of the form $P(u_1, u_2)$, if x binary, or $P(u_1, u_2, u_3)$, if x ternary, appearing in the sets X_v, where v is a node of t, are x atomic relations, then the CSP of t can be solved in deterministic polynomial time in the number of its variables. In general, however, one has to use a search algorithm, which needs nondeterministic polynomial time in the number of variables (again, the reader is referred to Subsection 3.3, and to the pointers to the appropriate litterature).

The described tableaux-like procedure can be made much more efficient by pruning the search space with constraint propagation during the construction of t, as sketched at the end of Subsection 1.2. The basic idea is that, we do not wait until the completion of the construction of t, to process the CSP. Instead, whenever new constraints arise, resulting from the adding of a new node to the tree being built, we propagate them to the constraints already existing. In particular, this will potentially detect more dead-ends, and consequently shorten the search space. The reader is referred to Subsection 1.2 for details.

9 Discussion 2

9.1 The abstract objects as time intervals and the roles as the *meets* relation

Instead of interpreting the nodes of the k-ary structures as time points, and the roles as discrete-time immediate-successor relations, we can interpret the former as time intervals, the latter as the *meets* relation of Allen's time interval Relation Algebra (RA) [1]. In particular, the satisfiability of a concept w.r.t. a weakly cyclic TBox will remain decidable. We get then a new family of languages for (continuous) spatial change. This new spatio-temporalisation of $\mathcal{ALC}(\mathcal{D})$ can be summarised as follows:

1. Temporalisation of the roles, so that they consist of $n + l$ Allen's *meets* relations $m_1, \ldots, m_n, m_{n+1}, \ldots, m_{n+l}$, of which the m_i's, with $i \leq n$, are general, not necessarily functional relations, and the m_i's, with $i \geq n + 1$, functional relations.
2. Spatialisation of the concrete domain $\mathcal{D}$, in a similar way as we did for the first family: the concrete domain is is generated by a spatial RA such as the Region-Connection Calculus RCC8 [67].

In Examples 1, 2, 3 and 4, instead of interpreting the roles (including the abstract features) as discrete-time accessibility relations, we can, and do in the rest of the section, interpret them as durative-time meets relations [1]. In Example 2, for instance, the abstract features f_1 and f_2 can be so interpreted. The motion of the corresponding spatial scene (Figure 2(Right)), when it reaches, for instance, Submotion B, remains in that configuration for a (durative) while, before reaching, without discontinuing, Submotion C in which it remains another while: in this respect, Submotion B "meets" Submotion C, which is indicated with the abstract feature f_1.

9.2 The properties of durativeness, continuity and density of (spatial) change

The discussion is aimed at clarifying cognitively plausible assumptions on spatial change in the physical world: *durativeness*, *continuity* and *density*. In the particular case of motion of a spatial scene, for instance, this intuitively means that, on the one hand, once the scene has reached a certain configuration, it remains in that configuration for a (durative) while, before eventually reaching a distinct configuration; and, on the other hand, the transition from a configuration c to the very first future configuration c' distinct from c, respects some continuity condition, so that c' is a neighbour of c, in a sense to be explained shortly. The transition also fulfils a *density* criterion, in the sense that, there is no temporal gap in the spatial scene, between the end of configuration c and the beginning of configuration c'. The discussion is related to continuity as discussed in [28].

The theory of conceptual neighbourhoods is well-known in qualitative spatial and temporal reasoning (QSTR) —see, for instance, [24]. QSTR constraint-based languages consist mainly of RAs. In the spatial case, for instance, the atoms of such an RA, in finite number, are built by defining an appropriate partition of the spatial domain at hand, on which the RA is supposed to represent knowledge, as constraints on n-tuples of objects, where n is the arity of the relations. We say appropriate partition, in the sense that the partition has to fulfil some requirements, such as cognitive adequacy criteria, so that the obtained RA reflects, for instance, the common-sense reasoning, or the reasoning required by the task the RA is meant to be used for, as much as possible. The regions of the partition are generally continuous, and each groups together elements of the universe which do not need to be distinguished, because, for instance, the task at hand does not need, or Humans do not make, such a distinction. Given two atoms r_1 and r_2 of such an RA, r_2 is said to be a conceptual neighbour of r_1, if the union of the corresponding regions in the partition is continuous, so that one can move from one to the other without traversing a third region of the partition. The conceptual neighbourhood of r_1 is nothing else than the set of all its conceptual neighbours, including r_1 itself. The conceptual neighbourhood of a general relation, which is a set of atoms, is the union of the conceptual neighbourhoods of its atoms. The meets relation in Allen's RA [1], for instance, has two conceptual neighbours other than itself, which are before ($<$) and overlaps (o); the $\mathcal{RCC}8$ relation TPP has PO, EQ and $NTPP$ as conceptual neighbours. The conceptual neighbourhoods of the atoms, e, l, o and r, of the $\mathcal{CYC}_b$ binary RA of 2-dimensional orientations in [43] are, respectively, $\{e, l, r\}$, $\{e, l, o\}$, $\{l, o, r\}$ and $\{e, o, r\}$. Concerning the ternary RA $\mathcal{CYC}_t$ in [43], an atom $b'_1b'_2b'_3$ is a conceptual neighbour of an atom $b_1b_2b_3$, where the b_i's and the b'_i's are $\mathcal{CYC}_b$ atoms, if and only if the $\mathcal{CYC}_b$ atoms b_i and b'_i, $i = 1 \ldots 3$, are conceptual neighbourhoods of each other. In Example 2, for instance, it is the case that the relation on any pair of the involved objects (the objects o1, o2 and o3 in Subscene 1; and the objects q1, q2 and q3 in Subscene 2), when moving from the current atomic submotion to the next, either remains the same, or changes to a relation that is a conceptual neighbour. The transition from Submotion A to Submotion B involves only the change of the $\mathcal{RCC}8$ relation on the pair (o2,o3) from TPP to its conceptual neighbour NTPP; one might then argue that, because the distinction between the TPP and NTPP relations involves only a 0- or 1-dimensional region, it might happen that the transition from the TPP configuration to the NTPP configuration of the pair (o2,o3) is not durative. Nevertheless, even in such extreme situations, the time required for the scene's motion to achieve the transition is considered as an interval.

Without loss of generality, we restrict the remainder of the discussion to one member of our $\mathcal{MTALC}(\mathcal{D}_x)$ family of theories for continuous spatial change, which is $\mathcal{MTALC}(\mathcal{D}_{\mathcal{RCC}8})$, whose concrete domain is generated by $\mathcal{RCC}8$. We denote by M a motion of a spatial scene $\mathcal{S}$ composed on n objects, $O_1, \ldots, O_n$.[8] For all $i, j \in \{1, \ldots, n\}$, $i < j$, we denote by $\mathcal{S}_{ij}$ the subscene of $\mathcal{S}$ composed of objects O_i and O_j; by M_{ij} the restriction of motion M to subscene $\mathcal{S}_{ij}$; by $M_{ij}^1, \ldots, M_{ij}^{n_{ij}}$ the $n_{ij} \geq 1$ atomic submotions of M_{ij}; by I_{ij}^k, $k \in \{1, \ldots, n_{ij}\}$, the interval during which atomic submotion M_{ij}^k takes place; and by r_{ij}^k, $k \in \{1, \ldots, n_{ij}\}$, the $\mathcal{RCC}8$ relation of the pair (O_i, O_j) during Submotion M_{ij}^k of Subscene $\mathcal{S}_{ij}$. Submotions M_{ij}^k and M_{ij}^{k+1}, $k \in \{1, \ldots, n_{ij} - 1\}$, of M_{ij} are such that M_{ij}^{k+1} immediately follows M_{ij}^k; in other words, the intervals during which they hold are related by the *meets* relation: $m(I_{ij}^k, I_{ij}^{k+1})$.

Definition 32 (maximal atomic submotion). *The atomic submotion M_{ij}^k, $k \in \{1, \ldots, n_{ij}\}$, of subscene $\mathcal{S}_{ij}$ is said to be maximal if M_{ij} has no atomic submotion that stricly subsumes M_{ij}^k.*

Definition 33 (continuous motion of a 2-object subscene). *The motion of Subscene $\mathcal{S}_{ij}$ is said to be continuous if it can be decomposed into n_{ij} maximal atomic submotions $M_{ij}^1, \ldots, M_{ij}^{n_{ij}}$ such that r_{ij}^{k+1} is a conceptual neighbourhood of r_{ij}^k, for all $k = 1, \ldots, n_{ij} - 1$.*

Definition 34 (continuous motion of scene $\mathcal{S}$). *The motion of Scene $\mathcal{S}$ is said to be continuous if its restriction to any of its 2-object subscenes is continuous.*

9.3 Qualitative probabilistic decision making

The previous discussion clearly suggests that one can design a qualitative probablistic decision maker which, given a physical system such as, for instance, the one in Example 2, would decide which Submotion the system should enter next. The continuity assumption implies that the submotion to enter next should fulfil the conceptual neighbourhood condition. If we assume a uniform probability distribution, then we can easily give the conditional probabilities governing the motion restricted to any pair (o1,o2) of the involved objects. For the purpose, we denote by $p(r'(o1, o2)|r(o1, o2))$ the probability for the relation on the pair (o1,02), to be r' at the next submotion of the scene, given that it is currently r. If n is the number of conceptual neighbours of r, then clearly:

$$p(r'(o1, o2)|r(o1, o2)) = \begin{cases} \frac{1}{n}, & \text{if } r' \text{ is a conceptual neighbour of } r \\ 0, & \text{otherwise} \end{cases}$$

10 Discussion 3

As explained in Section 8, the construction of the tree t, ouput of the procedure of Figure 6, can be done in such a way that, the solving of the CSP of t is entirely left until the end of the construction of the tree itself —a two-step construction, with no constraint propagation during the first step. The CSP of t is a finite conjunction of constraints of the form $P(x_1, \ldots, x_n)$, P being a predicate of the concrete domain, and

[8] The objects $O_1, \ldots, O_n$ are regions of a topological space.

$x_1, \ldots, x_n$ variables. Consistency of the CSP is decidable thanks to the admissibility of the concrete domain. We restricted the work to admissible concrete domains generated by $\mathcal{RCC}8$-like qualitative spatial languages [67, 18], for we wanted such languages to be combined with modal temporal logics in a way leading to flexible domain-specific languages for spatial change in general, and for motion of spatial scenes in particular. The reader should easily see that our decidability results (Theorem 3 and Corollary 3) extend to any admissible concrete domain, in the sense of "concrete domain" and "admissibility" which we have been using [3] (Definition 1 and 2).

11 Discussion 4: adding "atemporal" roles to $\mathcal{MTALC}(\mathcal{D}_x)$

Another important point to mention is that, the roles of $\mathcal{MTALC}(\mathcal{D}_x)$ are temporal. The *dnf*2 of a concept C is of the form $dnf2(C) = S_{prop} \cup S_{csp} \cup S_{\exists}$, where, in particular, S_{prop} is a set of primitive concepts and negated primitive concepts (literals), representing a conjunction of propositional knowledge. It is known that $\mathcal{ALC}(\mathcal{D})$ with an admissible concrete domain and an acyclic TBox is decidable [3]; i.e., if the concrete domain D is admissible then, satisfiability of an $\mathcal{ALC}(\mathcal{D})$ concept w.r.t. an acyclic TBox is decidable. If we denote by $\mathcal{ALCF}$ the DL $\mathcal{ALC}$ [73] augmented with abstract features, then clearly $\mathcal{ALCF}$ is a sublanguage of $\mathcal{ALC}(\mathcal{D})$, and thus satisfiability of an $\mathcal{ALCF}$ concept with respect to an $\mathcal{ALCF}$ acyclic TBox is decidable. $\mathcal{ALCF}$ is particularly important for the representation of structured data, such as data in *XML* documents (see, e.g., [12, 21]), thanks, among other things, to its abstract features, which allow it to access specific paths. We can add "atemporal" roles to $\mathcal{MTALC}(\mathcal{D}_x)$, so that $\mathcal{ALCF}$ gets subsumed, again without compromising our decidability results. Such an extension would offer a representational tool for the history of structured data in general, and of data in *XML* documents in particular. It would also offer a representational tool for event models in high-level computer vision (see, e.g., [4, 62]), which are "a representation of classes of events and a tool to recognise events in a given scene [62]". The concepts of such a language are the atemporal concepts and the temporal concepts, as given by the definition below:

Definition 35. *Let x be an RA from the set $\{\mathcal{RCC}8, \mathcal{CDA}, \mathcal{CYC}_t\}$. Let N_C, N_R^a, N_R^t and N_{cF} be mutually disjoint and countably infinite sets of concept names, atemporal role names, temporal role names, and concrete features, respectively; N_{aF}^a a countably infinite subset of N_R^a whose elements are atemporal abstract features; and N_{aF}^t a countably infinite subset of N_R^t whose elements are temporal abstract features. A temporal (concrete) feature chain is any finite composition $f_1^t \ldots f_n^t g$ of $n \geq 0$ temporal abstract features $f_1^t, \ldots, f_n^t$ and one concrete feature g. The set of atemporal concepts and the set of temporal concepts are the smallest sets such that:*

1. *$\top$ and $\bot$ are atemporal concepts*
2. *a concept name is an atemporal concept*
3. *if C^a and D^a are atemporal concepts; C^t and D^t are temporal concepts; R^a is an atemporal role (in general, and an atemporal abstract feature in particular); R^t is a temporal role (in general, and a temporal abstract feature in particular); g is a concrete feature; u_1^t, u_2^t and u_3^t are temporal feature chains; and P is a predicate, then:*
   (a) $\neg C^a$, $C^a \sqcap D^a$, $C^a \sqcup D^a$, $\exists R^a.C^a$, $\forall R^a.C^a$ are atemporal concepts;

(b) $\neg C^t$, $C^t \sqcap D^t$, $C^t \sqcup D^t$, $\exists R^t.C^t$, $\forall R^t.C^t$ *are temporal concepts;*
(c) $C^a \sqcap C^t$, $C^a \sqcup C^t$, $\exists R^t.C^a$, $\forall R^t.C^a$ *are temporal concepts; and*
(d) $\exists(u_1^t)(u_2^t).P$, *if* x *binary,* $\exists(u_1^t)(u_2^t)(u_3^t).P$, *if* x *ternary, are temporal concepts.*

A TBox T is now weakly cyclic if it satisfies the following two conditions:

1. Whenever A uses B and B uses A, we have $B = A$.
2. All possible occurrences of a defined concept B in the right hand side of the axiom defining B itself, are within subconcepts of C of the form $\exists R.D$ or $\forall R.D$, C being the right hand side of the axiom, $B \doteq C$, defining B, with R being a temporal role —in other words, we do not allow any cyclicity in any atemporal part of the TBox.

12 Conclusion

We have described how to enhance the expressiveness of modal temporal logics with qualitative spatial constraints. The theoretical framework consists of a spatio-temporalisation of the $\mathcal{ALC}(\mathcal{D})$ family of description logics with a concrete domain [3], obtained by temporalising the roles, so that they consist of $m+n$ immediate-successor (accessibility) relations, the first m being general, not necessarily functional roles, the other n abstract features; and spatialising the concrete domain, which is generated by an $\mathcal{RCC}8$-like qualitative spatial language [67, 18]. The result is a family $\mathcal{MTALC}(\mathcal{D}_x)$ of languages for qualitative spatial change in general, and for motion of spatial scenes in particular. We considered $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBoxes, expressive enough to capture most of existing modal temporal logics -which was shown for Propositional Linear Temporal Logic $\mathcal{PLTL}$, and for the $\mathcal{CTL}$ version of the full branching modal temporal logic $\mathcal{CTL}^*$ [19]. We proved that satisfiability of a concept w.r.t. such a TBox is decidable, by reducing it to the emptiness problem of a weak alternating automaton [58] augmented with qualitative spatial constraints. The accepting condition of a run of such an augmented weak alternating automaton involves, additionally to the states infinitely often repeated, consistency of a CSP (Constraint Satisfaction Problem) potentially infinite. Nevertheless, the emptiness problem was shown to remain decidable. A tableaux-like procedure for the satisfiability of an $\mathcal{MTALC}(\mathcal{D}_x)$ concept w.r.t. an $\mathcal{MTALC}(\mathcal{D}_x)$ weakly cyclic TBox, which we have discussed, is straightforwardly obtainable from our results.

We also discussed various extensions of the work. In particular, we discussed that, if, instead of interpreting the nodes of the k-ary structures as time points, and the roles as immediate-successor relations, we interpreted the former as time intervals, the latter as the *meets* relation of Allen's time interval Relation Algebra (RA) [1], the satisfiability of a concept w.r.t. a weakly cyclic TBox remained decidable. This led to a new family of languages for continuous spatial change.

Acknowledgement

I had discussions with Bernd Neumann on issues related to the presented work.

References

1. J F Allen. Maintaining knowledge about temporal intervals. *Communications of the Association for Computing Machinery*, 26(11):832–843, 1983.
2. A Artale and E Franconi. A Survey of Temporal Extensions of Description Logics. *Annals of Mathematics and Artificial Intelligence*, 30(1-4):171–210, 2000.
3. F Baader and P Hanschke. A scheme for integrating concrete domains into concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 452–457, Sydney, 1991.
4. N I Badler. *Temporal Scene Analysis: Conceptual Descriptions of Object Movements.* PhD thesis, University of Toronto, Toronto, Ontario, Canada, February 1975.
5. P Balbiani and J-F Condotta. Computational Complexity of Propositional Linear Temporal Logics Based on Qualitative Spatial or Temporal Reasoning. In *Proceedings FroCos*, pages 162–176, 2002.
6. P Balbiani, J-F Condotta, and L F del Cerro. A model for reasoning about bidimensional temporal relations. In *Proceedings of Principles of Knowledge Representation and Reasoning (KR)*, pages 124–130. Morgan Kaufmann, 1998.
7. F Benhamou and A Colmerauer, editors. *Constraint Logic Programming: Selected Research.* MIT Press, 1993.
8. B Bennett, A G Cohn, F Wolter, and M Zakharyaschev. Multi-Dimensional Modal Logic as a Framework for Spatio-Temporal Reasoning. *Applied Intelligence*, 2002.
9. B Bennett, C Dixon, M Fischer, U Hustadt, E Franconi, I Horrocks, and M Rijke. Combinations of Modal Logics. *AI Review*, 17(1):1–20, 2002.
10. B Bennett, A Isli, and A G Cohn. A system handling RCC-8 Queries on 2D Regions representable in the Closure Algebra of Half-Planes. In D Abel and B C Ooi, editors, *Proceedings of the 11th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA-AIE)*, Lecture Notes in Artificial Intelligence, pages 281–290. Springer-Verlag, 1998.
11. C Bettini. Time-dependent concepts: representation and reasoning using temporal description logics. *Data & Knowledge Engineering*, 22:1–38, 1997.
12. P Buneman, W Fan, J Siméon, and S Weinstein. Constraints for Semi-structured Data and XML. *SIGMOD Record*, 30(1):47–54, 2001.
13. B L Clarke. A Calculus of Individuals based on 'Connection'. *Notre Dame Journal of Formal Logic*, 23(3):204–218, July 1981.
14. A G Cohn. Qualitative spatial representation and reasoning techniques. In *Proceedings KI: German Annual Conference on Artificial Intelligence*, volume 1303 of *Lecture Notes in Artificial Intelligence*, pages 1–30. Springer-Verlag, 1997.
15. A Colmerauer. An Introduction to Prolog III. *Communications of the ACM*, 33(7):69–90, 1990.
16. P Dague (MQ&D coordinated by). Qualitative Reasoning: A Survey of Techniques and Applications. *AI Communications*, 8(3/4):119–192, 1995.
17. I Düntsch, H Wang, and S McCloskey. Relation algebras in qualitative spatial reasoning. *Fundamenta Informatica*, 39:229–248, 1999.
18. M Egenhofer. Reasoning about binary topological relations. In *Proceedings SSD*, LNCS 525, pages 143–160, 1991.
19. E A Emerson. Temporal and modal logic. In J van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, pages 995–1072. Elsevier and MIT Press, 1990.
20. M T Escrig and F Toledo. *Qualitative Spatial Reasoning: Theory and Practice (Application to Robot Navigation).* Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, 1998.
21. W Fan, G M Kuper, and J Siméon. A unified constraint model for XML. *Computer Networks*, 39(5):489–505, 2002.

22. K D Forbus, P Nielsen, and B Faltings. Qualitative spatial reasoning: The clock project. *Artificial Intelligence*, 51:417–471, 1991.
23. A U Frank. Qualitative spatial reasoning about distances and directions in geographic space. *Journal of Visual Languages and Computing*, 3:343–371, 1992.
24. C Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54:199–227, 1992.
25. C Freksa. Using Orientation Information for Qualitative Spatial Reasoning. In A U Frank, I Campari, and U Formentini, editors, *Proceedings of GIS —from Space to Territory: Theories and Methods of Spatio-Temporal Reasoning*, Berlin, 1992. Springer.
26. E C Freuder. A sufficient condition for backtrack-free search. *Journal of the Association for Computing Machinery*, 29:24–32, 1982.
27. D Gabbay, A Kurucz, F Wolter, and M Zakharyaschev. *Many-Dimensional Modal Logics: Teory and Applications.* 2002. Forthcoming.
28. A Galton. Space, time, and movement. In O Stock, editor, *Spatial and Temporal Reasoning*, chapter 10, pages 321–352. Kluwer Academic Publishers, Dordrecht/Boston/London, 1997.
29. M R Garey and D S Johnson. *Computers and intractability.* Freeman, New York, 1979.
30. H Güsgen. Spatial reasoning based on Allen's temporal logic. Technical Report TR-89-049, ICSI, Berkley, CA, 1989.
31. V Haarslev, C Lutz, and R Möller. A description logic with concrete domains and a role-forming predicate operator. *Journal of Logic and Computation*, 9(3):351–384, 1999.
32. C Habel. Representing space and time: Discrete, dense or continuous? is that the question? In C Eschenbach and W Heydrich, editors, *Parts and Wholes — Integrity and Granularity*, pages 97–107. 1995.
33. J Y Halpern and Y Moses. A guide to the modal logics of knowledge and belief. In *Proceedings IJCAI*, pages 480–490, Los Angeles, CA, 1985.
34. P J Hayes. Naive physics 1: Ontology for liquids. In J R Hobbs and R C Moore, editors, *Formal Theories of the Commonsense World*, pages 71–107. Ablex Publishing Corporation, Norwood, NJ, 1985.
35. P J Hayes. The second naive physics manifesto. In J R Hobbs and R C Moore, editors, *Formal Theories of the Commonsense World*, pages 1–36. Ablex, 1985.
36. S M Hazarika and A G Cohn. Qualitative Spatio-Temporal Continuity. In D R Montello, editor, *Spatial Information Theory: Foundations of GIS*, number 2205 in LNCS, pages 92–107, Morro Bay, CA, 2001. Springer.
37. S M Hazarika and A G Cohn. Abducing Qualitative Spatio-Temporal Histories from Partial Observations. In *Proceedings KR*, 2002. To appear.
38. J E Hopcroft and J D Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Publishing Company, 1979.
39. A Isli. *Automates alternants et logiques temporelles, satisfaction de contraintes temporelles.* PhD thesis, Université Paris XIII, November 1993.
40. A Isli. Converting a Büchi alternating automaton to a usual nondeterministic one. *SADHANA*, 21(2):213–228, 1996. Indian Academy of Sciences.
41. A Isli. A family of qualitative theories for continuous spatio-temporal change as a spatio-temporalisation of ALC(D) —first results. In R V Rodríguez, editor, *Proceedings of the ECAI Workshop on Spatial and Temporal Reasoning*, pages 81–86, Lyon, France, 2002.
42. A Isli and A G Cohn. An Algebra for cyclic Ordering of 2D Orientations. In *Proceedings AAAI*, pages 643–649, Madison, WI, 1998. AAAI Press.
43. A Isli and A G Cohn. A new Approach to cyclic Ordering of 2D Orientations using ternary Relation Algebras. *Artificial Intelligence*, 122(1-2):137–187, 2000.
44. J Jaffar and J-L Lassez. Constraint Logic Programming. In *Proceedings 14th ACM Symposium on Principles of Programming Languages (POPL)*, pages 111–119, Munich, 1987.
45. J Jaffar and M J Maher. Constraint Logic Programming: A Survey. *Journal of Logic Programming*, 19-20:503–581, 1994.

46. P Ladkin and R Maddux. On binary Constraint Problems. *Journal of the Association for Computing Machinery*, 41(3):435–469, 1994.
47. P Ladkin and A Reinefeld. Effective Solution of qualitative Constraint Problems. *Artificial Intelligence*, 57:105–124, 1992.
48. J-C Latombe. *Robot Motion Planning.* Kluwer, Dordrecht, Holland, 1991.
49. H S Leonard and N Goodman. The Calculus of Individuals and its Uses. *Journal of Symbolic Logic*, 5:45–55, 1940.
50. S Leśniewski. O Podstawach Ogolnej Teoryi Mnogosci. *Przeglad Filozoficzny*, 31:261–291, 1928.
51. G Ligozat. Reasoning about cardinal Directions. *Journal of Visual Languages and Computing*, 9(1):23–44, 1998.
52. C Lutz. *The Complexity of Description Logics with Concrete Domains.* PhD thesis, LuFG Theoretical Computer Science, RWTH, Aachen, 2001.
53. A K Mackworth. Consistency in Networks of Relations. *Artificial Intelligence*, 8:99–118, 1977.
54. R D Maddux. *Relation Algebras.* 2002. Book in preparation.
55. M Minsky. A framework for representing knowledge. In P H Winston, editor, *The Psychology of Computer Vision*, pages 211–277. McGraw-Hill, 1975.
56. U Montanari. Networks of Constraints: fundamental Properties and Applications to Picture Processing. *Information Sciences*, 7:95–132, 1974.
57. A Mukerjee and G Joe. A qualitative Model for Space. In *Proceedings AAAI-90*, pages 721–727, Los Altos, 1990. Morgan Kaufmann.
58. D E Muller, A Saoudi, and P E Schupp. Alternating automata, the weak monadic theory of trees and its complexity. *Theoretical Computer Science*, 97:233–244, 1992.
59. D E Muller and P E Schupp. Alternating automata on infinite trees. *Theoretical Computer Science*, 54:267–276, 1987.
60. D E Muller and P E Schupp. Simulating alternating Tree Automata by nondeterministic Automata: New Results and new Proofs of the Theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science*, 141:69–107, 1995.
61. P Muller. A qualitative Theory of Motion based on spatio-temporal Primitives. In *Proceedings of Principles of Knowledge Representation and Reasoning (KR)*, pages 131–141. Morgan Kaufmann, 1998.
62. B Neumann and H-J Novak. Event models for recognition and natural language description of events in real-world image sequences. In *Proceedings of the Eighth IJCAI Conference*, pages 724–726, August 1983.
63. J Pearl. *CAUSALITY Models, Reasoning, and Inference.* Cambridge University Press, 2000.
64. M R Quillian. Semantic memory. In M Minsky, editor, *Semantic Information Processing*, pages 227–270. MIT Press, 1968.
65. M O Rabin. Decidability of the second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.
66. M O Rabin. Weakly definable Relations and special Automata. In Y Bar-Hillel, editor, *Proceedings of the Symposium Mathematical Logic and Foundations of Set Theory*, pages 1–23, Amsterdam, 1970.
67. D Randell, Z Cui, and A Cohn. A spatial Logic based on Regions and Connection. In *Proceedings KR-92*, pages 165–176, San Mateo, 1992. Morgan Kaufmann.
68. J Renz and B Nebel. On the Complexity of Qualitative Spatial Reasoning: A maximal tractable Fragment of the Region Connection Calculus. *Artificial Intelligence*, 108:69–123, 1999.
69. J Renz, R Rauh, and M Knauff. Towards Cognitive Adequacy of Topological Spatial Relations. In C Freksa, W Brauer, C Habel, and K F Wender, editors, *Spatial Cognition II - Integrating abstract theories, empirical studies, formal models, and practical applications*, volume 1849 of *Lecture notes in Artificial Intelligence*, pages 184–197, Berlin, 2000. Springer-Verlag.

70. R D Rimey and C M Brown. Where to look next using a bayes net: Incorporating geometric relations. In G Sandini, editor, *Computer Vision — ECCV '92*, Lecture Notes in Computer Science No. 588, pages 542–550, Santa Margherita Ligure, Italy, may 1992. ECCV, Springer-Verlag.
71. K Schild. A Correspondence Theory for Terminological Logics: Preliminary Report. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 466–471, Sydney, 1991.
72. C Schlieder. Representing Visible Locations for Qualitative Navigation. In *Qualitative Reasoning and Decision Technologies: CIMNE*, 1993.
73. M Schmidt-Schauss and G Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.
74. A Scivos and B Nebel. Double-Crossing: Decidability and Computational Complexity of a Qualitative Calculus for Navigation. In D R Montello, editor, *Spatial Information Theory: Foundations of GIS*, number 2205 in LNCS, pages 431–446, Morro Bay, CA, 2001. Springer.
75. A Tarski. On the Calculus of Relations. *Journal of Symbolic Logic*, 6:73–89, 1941.
76. J F A K van Benthem. *The Logic of Time*. D. Reidel Publishing Company, Dordrecht, Holland, 1983.
77. P van Hentenryck. *Constraint Satisfaction in Logic Programming*. MIT Press, Cambridge, MA, 1989.
78. M Y Vardi and P Wolper. Automata-theoretic Techniques for modal Logics of Programs. *Journal of Computer and System Science*, 32(2):183–221, 1986.
79. L Vila. A Survey on Temporal Reasoning in Artificial Intelligence. *AI Communications*, 7(1):4–28, 1994.
80. M B Vilain and H Kautz. Constraint Propagation Algorithms for Temporal Reasoning. In *Proceedings AAAI-86*, Philadelphia, August 1986.
81. A N Whitehead. *Process and Reality*. The MacMillan Company, New York, 1929.
82. F Wolter and M Zakharyaschev. Spatio-temporal Representation and Reasoning based on RCC-8. In A G Cohn, F Giunchiglia, and B Selman, editors, *Proceedings of Principles of Knowledge Representation and Reasoning (KR)*, pages 3–14, 2000.
83. K Zimmermann and C Freksa. Qualitative Spatial Reasoning using Orientation, Distance, and Path Knowledge. *Applied Intelligence*, 6:49–58, 1996.