ASPMT(QS): Non-Monotonic Spatial Reasoning with Answer Set Programming Modulo Theories

Przemysław Andrzej Wałęga[1]([⊠]), Mehul Bhatt[2], and Carl Schultz[2]

[1] Institute of Philosophy, University of Warsaw, Warsaw, Poland
`przemek.walega@wp.pl`
[2] Department of Computer Science, University of Bremen, Bremen, Germany

Abstract. The systematic modelling of *dynamic spatial systems* [9] is a key requirement in a wide range of application areas such as comonsense cognitive robotics, computer-aided architecture design, dynamic geographic information systems. We present ASPMT(QS), a novel approach and fully-implemented prototype for non-monotonic spatial reasoning — a crucial requirement within dynamic spatial systems– based on Answer Set Programming Modulo Theories (ASPMT). ASPMT(QS) consists of a (qualitative) spatial representation module (QS) and a method for turning tight ASPMT instances into Sat Modulo Theories (SMT) instances in order to compute stable models by means of SMT solvers. We formalise and implement concepts of default spatial reasoning and spatial frame axioms using choice formulas. Spatial reasoning is performed by encoding spatial relations as systems of polynomial constraints, and solving via SMT with the theory of real nonlinear arithmetic. We empirically evaluate ASPMT(QS) in comparison with other prominent contemporary spatial reasoning systems. Our results show that ASPMT(QS) is the only existing system that is capable of reasoning about indirect spatial effects (i.e. addressing the ramification problem), and integrating geometric and qualitative spatial information within a non-monotonic spatial reasoning context.

Keywords: Non-monotonic spatial reasoning · Answer set programming modulo theories · Declarative spatial reasoning

1 Introduction

Non-monotonicity is characteristic of commonsense reasoning patterns concerned with, for instance, making default assumptions (e.g., about spatial inertia), counterfactual reasoning with hypotheticals (e.g., what-if scenarios), knowledge interpolation, explanation & diagnosis (e.g., filling the gaps, causal links), belief revision. Such reasoning patterns, and therefore non-monotonicity, acquires a special significance in the context of *spatio-temporal dynamics*, or computational commonsense *reasoning about space, actions, and change* as applicable within areas as disparate as geospatial dynamics, computer-aided design, cognitive vision, commonsense cognitive robotics [6]. Dynamic spatial systems are characterised

by scenarios where spatial configurations of objects undergo a change as the result of interactions within a physical environment [9]; this requires explicitly identifying and formalising relevant actions and events at both an ontological and (qualitative and geometric) spatial level, e.g. formalising *desertification* and *population displacement* based on spatial theories about *appearance, disappearance, splitting, motion*, and *growth* of regions [10]. This calls for a deep integration of spatial reasoning within KR-based non-monotonic reasoning frameworks [7].

We select aspects of a theory of *dynamic spatial systems*—pertaining to *(spatial) inertia, ramifications, causal explanation*— that are inherent to a broad category of dynamic spatio-temporal phenomena, and require non-monotonic reasoning [5,9]. For these aspects, we provide an operational semantics and a computational framework for realising fundamental non-monotonic spatial reasoning capabilities based on Answer Set Programming Modulo Theories [3]; ASPMT is extended to the qualitative spatial (QS) domain resulting in the non-monotonic spatial reasoning system ASPMT(QS). Spatial reasoning is performed in an analytic manner (e.g. as with reasoners such as CLP(QS) [8]), where spatial relations are encoded as systems of polynomial constraints; the task of determining whether a spatial graph G is consistent is now equivalent to determining whether the system of polynomial constraints is satisfiable, i.e. Satisfiability Modulo Theories (SMT) with real nonlinear arithmetic, and can be accomplished in a sound and complete manner. Thus, ASPMT(QS) consists of a (qualitative) spatial representation module and a method for turning tight ASPMT instances into Sat Modulo Theories (SMT) instances in order to compute stable models by means of SMT solvers.

In the following sections we present the relevant foundations of stable model semantics and ASPMT, and then extend this to ASPMT(QS) by defining a (qualitative) spatial representations module, and formalising spatial default reasoning and spatial frame axioms using choice formulas. We empirically evaluate ASPMT(QS) in comparison with other existing spatial reasoning systems. We conclude that ASMPT(QS) is the only system, to the best of our knowledge, that operationalises dynamic spatial reasoning within a KR-based framework.

2   Preliminaries

2.1   Bartholomew – Lee Stable Models Semantics

We adopt a definition of stable models based on syntactic transformations [2] which is a generalization of the previous definitions from [13,14,19]. For predicate symbols (constants or variables) u and c, expression $u \leq c$ is defined as shorthand for $\forall \mathbf{x}(u(\mathbf{x}) \rightarrow c(\mathbf{x}))$. Expression $u = c$ is defined as $\forall \mathbf{x}(u(\mathbf{x}) \equiv c(\mathbf{x}))$ if u and c are predicate symbols, and $\forall \mathbf{x}(u(\mathbf{x}) = c(\mathbf{x}))$ if they are function symbols. For lists of symbols $\mathbf{u} = (u_1, \ldots, u_n)$ and $\mathbf{c} = (c_1, \ldots, c_n)$, expression $\mathbf{u} \leq \mathbf{c}$ is defined as $(u_1 \leq c_1) \wedge \cdots \wedge (u_n \leq c_n)$, and similarly, expression $\mathbf{u} = \mathbf{c}$ is defined as $(u_1 = c_1) \wedge \cdots \wedge (u_n = c_n)$. Let $\mathbf{c}$ be a list of distinct predicate and function constants, and let $\widehat{\mathbf{c}}$ be a list of distinct predicate and function variables corresponding to c. By $\mathbf{c}^{pred}$ ($\mathbf{c}^{func}$, respectively) we mean the list

of all predicate constants (function constants, respectively) in $\mathbf{c}$, and by $\widehat{\mathbf{c}}^{pred}$ ($\widehat{\mathbf{c}}^{func}$, respectively) the list of the corresponding predicate variables (function variables, respectively) in $\widehat{\mathbf{c}}$. In what follows, we refer to function constants and predicate constants of arity 0 as object constants and propositional constants, respectively.

Definition 1 (Stable model operator SM). *For any formula F and any list of predicate and function constants $\mathbf{c}$ (called intensional constants), $\mathbf{SM}[F;\mathbf{c}]$ is defined as*

$$F \wedge \neg\exists\widehat{\mathbf{c}}(\widehat{\mathbf{c}} < \mathbf{c} \wedge F^*(\widehat{\mathbf{c}})), \tag{1}$$

where $\widehat{\mathbf{c}} < \mathbf{c}$ is a shorthand for $(\widehat{\mathbf{c}}^{pred} \leq \mathbf{c}^{pred}) \wedge \neg(\widehat{\mathbf{c}} = \mathbf{c})$ and $F^*(\widehat{\mathbf{c}})$ is defined recursively as follows:

– *for atomic formula F, $F^* \equiv F' \wedge F$, where F' is obtained from F by replacing all intensional constants $\mathbf{c}$ with corresponding variables from $\widehat{\mathbf{c}}$,*
– *$(G \wedge H)^* = G^* \wedge H^*$, $(G \vee H)^* = G^* \vee H^*$,*
– *$(G \rightarrow H)^* = (G^* \rightarrow H^*) \wedge (G \rightarrow H)$,*
– *$(\forall x G)^* = \forall x G^*$, $(\exists x G)^* = \exists x G^*$.*

$\neg F$ is a shorthand for $F \rightarrow \bot$, $\top$ for $\neg\bot$ and $F \equiv G$ for $(F \rightarrow G) \wedge (G \rightarrow F)$.

Definition 2 (Stable model). *For any sentence F, a stable model of F on $\mathbf{c}$ is an interpretation I of underlying signature such that $I \models \mathbf{SM}[F;\mathbf{c}]$.*

2.2   Turning ASPMT into SMT

It is shown in [3] that a tight part of ASPMT instances can be turned into SMT instances and, as a result, off-the-shelf SMT solvers (e.g. Z3 for arithmetic over reals) may be used to compute stable models of ASP, based on the notions of Clark normal form, Clark completion.

Definition 3 (Clark normal form). *Formula F is in* Clark normal form *(relative to the list $\mathbf{c}$ of intensional constants) if it is a conjunction of sentences of the form (2) and (3).*

$$\forall\mathbf{x}(G \rightarrow p(\mathbf{x})) \qquad (2) \qquad\qquad \forall\mathbf{x}y(G \rightarrow f(\mathbf{x}) = y) \qquad (3)$$

one for each intensional predicate p and each intensional function f, where $\mathbf{x}$ is a list of distinct object variables, y is an object variable, and G is an arbitrary formula that has no free variables other than those in $\mathbf{x}$ and y.

Definition 4 (Clark completion). *The* completion *of a formula F in Clark normal form (relative to $\mathbf{c}$), denoted by $Comp_{\mathbf{c}}[F]$ is obtained from F by replacing each conjunctive term of the form (2) and (3) with (4) and (5) respectively*

$$\forall\mathbf{x}(G \equiv p(\mathbf{x})) \qquad (4) \qquad\qquad \forall\mathbf{x}y(G \equiv f(\mathbf{x}) = y). \qquad (5)$$

Definition 5 (Dependency graph). *The* dependency graph *of a formula F (relative to $\mathbf{c}$) is a directed graph $DG_{\mathbf{c}}[F] = (V, E)$ such that:*

1. V consists of members of $\boldsymbol{c}$,
2. for each $c, d \in V$, $(c, d) \in E$ whenever there exists a strictly positive occurrence of $G \rightarrow H$ in F, such that c has a strictly positive occurrence in H and d has a strictly positive occurrence in G,

where an occurrence of a symbol or a subformula in F is called strictly positive in F if that occurrence is not in the antecedent of any implication in F.

Definition 6 (**Tight formula**). *Formula F is* tight *(on $\boldsymbol{c}$) if $DG_{\boldsymbol{c}}[F]$ is acyclic.*

Theorem 1 (Bartholomew, Lee). *For any sentence F in Clark normal form that is tight on $\boldsymbol{c}$, an interpretation I that satisfies $\exists xy(x = y)$ is a model of $\boldsymbol{SM}[F; \boldsymbol{c}]$ iff I is a model of $Comp_{\boldsymbol{c}}[F]$ relative to $\boldsymbol{c}$.*

3   ASPMT with Qualitative Space – ASPMT(QS)

In this section we present our spatial extension of ASPMT, and formalise spatial default rules and spatial frame axioms.

3.1   The Qualitative Spatial Domain $\mathcal{QS}$

Qualitative spatial calculi can be classified into two groups: topological and positional calculi. With topological calculi such as the *Region Connection Calculus* (RCC) [25], the primitive entities are spatially extended regions of space, and could possibly even be 4D spatio-temporal histories, e.g., for *motion-pattern* analyses. Alternatively, within a dynamic domain involving translational motion, point-based abstractions with orientation calculi could suffice. Examples of orientation calculi include: the Oriented-Point Relation Algebra ($\mathcal{OPRA}_m$) [22], the Double-Cross Calculus [16]. The qualitative spatial domain ($\mathcal{QS}$) that we consider in the formal framework of this paper encompasses the following ontology:

QS1.   Domain Entities in $\mathcal{QS}$. Domain entities in $\mathcal{QS}$ include *circles, triangles, points* and *segments*. While our method is applicable to a wide range of 2D and 3D spatial objects and qualitative relations, for brevity and clarity we primarily focus on a 2D spatial domain. Our method is readily applicable to other 2D and 3D spatial domains and qualitative relations, for example, as defined in [8, 11, 12, 23, 24, 26, 27].

– a *point* is a pair of reals x, y
– a *line segment* is a pair of end points p_1, p_2 ($p_1 \neq p_2$)
– a *circle* is a centre point p and a real radius r ($0 < r$)
– a *triangle* is a triple of vertices (points) p_1, p_2, p_3 such that p_3 is *left of* segment p_1, p_2.

QS2.   Spatial Relations in $\mathcal{QS}$. We define a range of spatial relations with the corresponding polynomial encodings. Examples of spatial relations in $\mathcal{QS}$ include:

Relative Orientation.   Left, right, collinear orientation relations between *points* and *segments*, and *parallel, perpendicular* relations between *segments* [21].

Mereotopology.   Part-whole and *contact* relations between regions [25, 28].

3.2   Spatial Representations in ASPMT(QS)

Spatial representations in ASPMT(QS) are based on parametric functions and qualitative relations, defined as follows.

Definition 7 (Parametric function). *A* parametric function *is an n–ary function $f_n : D_1 \times D_2 \times \cdots \times D_n \to \mathbb{R}$ such that for any $i \in \{1 \ldots n\}$, D_i is a type of spatial object, e.g., Points, Circles, Polygons, etc.*

Example 1. Consider following parametric functions $x : Circles \to \mathbb{R}$, $y : Circles \to \mathbb{R}$, $r : Circles \to \mathbb{R}$ which return the position values x, y of a circle's centre and its radius r, respectively. Then, circle $c \in Cirlces$ may be described by means of parametric functions as follows: $x(c) = 1.23 \wedge y(c) = -0.13 \wedge r(c) = 2$.

Definition 8 (Qualitative spatial relation). *A* qualitative spatial relation *is an n-ary predicate $Q_n \subseteq D_1 \times D_2 \times \cdots \times D_n$ such that for any $i \in \{1 \ldots n\}$, D_i is a type of spatial object. For each Q_n there is a corresponding formula of the form*

$$\forall d_1 \in D_1 \ldots \forall d_n \in D_n \Big(Q_n(d_1, \ldots, d_n) \leftarrow p_1(d_1, \ldots, d_n) \wedge \cdots \wedge p_m(d_1, \ldots, d_n) \Big) \ (6)$$

where $m \in \mathbb{N}$ and for any $i \in \{1 \ldots n\}$, p_i is a polynomial equation or inequality.

Proposition 1. *Each qualitative spatial relation according to Definition 8 may be represented as a tight formula in Clark normal form.*

Proof. Follows directly from Definitions 3 and 8.

Thus, qualitative spatial relations belong to a part of ASPMT that may be turned into SMT instances by transforming the implications in the corresponding formulas into equivalences (Clark completion). The obtained equivalence between polynomial expressions and predicates enables us to compute relations whenever parametric information is given, and vice versa, i.e. computing possible parametric values when only the qualitative spatial relations are known.

Many relations from existing qualitative calculi may be represented in ASPMT(QS) according to Definition 8; our system can express the polynomial encodings presented in e.g. [8,11,12,23,24]. Here we give some illustrative examples.

Proposition 2. *Each relation of Interval Algebra (IA) [1] and Rectangle Algebra (RA) [20] may be defined in ASPMT(QS).*

Proof. Each IA relation may be described as a set of equalities and inequalities between interval endpoints (see Fig. 1 in [1]), which is a conjunction of polynomial expressions. RA makes use of IA relations in 2 and 3 dimensions. Hence, each relation is a conjunction of polynomial expressions [27].

Proposition 3. *Each relation of RCC–5 in the domain of convex polygons with a finite maximum number of vertices may be defined in ASPMT(QS).*

Proof. Each RCC–5 relation may be described by means of relations $P(a, b)$ and $O(a, b)$. In the domain of convex polygons, $P(a, b)$ is true whenever all vertices of a are in the interior (inside) or on the boundary of b, and $O(a, b)$ is true if there exists a point p that is inside both a and b. Relations of a point being inside, outside or on the boundary of a polygon can be described by polynomial expressions e.g. [8]. Hence, all RCC–5 relations may be described with polynomials, given a finite upper limit on the number of vertices a convex polygon can have.

Proposition 4. *Each relation of Cardinal Direction Calculus (CDC) [15] may be defined in ASPMT(QS).*

Proof. CDC relations are obtained by dividing space with 4 lines into 9 regions. Since halfplanes and their intersections may be described with polynomial expressions, then each of the 9 regions may be encoded with polynomials. A polygon object is in one or more of the 9 cardinal regions by the topological *overlaps* relation between polygons, which can be encoded with polynomials (i.e. by the existence of a shared point) [8].

3.3   Choice Formulas in ASPMT(QS)

A choice formula [14] is defined for a predicate constant p as $\texttt{Choice}(p) \equiv \forall \mathbf{x}(p(\mathbf{x}) \lor \neg p(\mathbf{x}))$ and for function constant f as $\texttt{Choice}(f) \equiv \forall \mathbf{x}(f(\mathbf{x}) = y \lor \neg f(\mathbf{x}) = y)$, where $\mathbf{x}$ is a list of distinct object variables and y is an object variable distinct from $\mathbf{x}$. We use the following notation: $\{F\}$ for $F \lor \neg F$, $\forall \mathbf{x} y \{f(\mathbf{x}) = y\}$ for $\texttt{Choice}(f)$ and $\forall \mathbf{x} \{p(\mathbf{x})\}$ for $\texttt{Choice}(p)$. Then, $\{\mathbf{t} = \mathbf{t}'\}$, where $\mathbf{t}$ contains an intentional function constant and $\mathbf{t}'$ does not, represents the default rule stating that $\mathbf{t}$ has a value of $\mathbf{t}'$ if there is no other rule requiring $\mathbf{t}$ to take some other value.

Definition 9 *(Spatial choice formula).* *The* spatial choice formula *is a rule of the form (8) or (7):*

$$\{f_n(d_1, \ldots, d_n) = x\} \leftarrow \alpha_1 \land \alpha_2 \land \cdots \land \alpha_k, \tag{7}$$

$$\{Q_n(d_1, \ldots, d_n)\} \leftarrow \alpha_1 \land \alpha_2 \land \cdots \land \alpha_k. \tag{8}$$

where f_n is a parametric function, $x \in \mathbb{R}$, Q_n is a qualitative spatial relation, and for each $i \in \{1, \ldots, k\}$, α_i is a qualitative spatial relation or expression of a form $\{f_r(d_k, \ldots, d_m) = y\}$ or a polynomial equation or inequality, whereas $d_i \in D_i$ is an object of spatial type D_i.

Definition 10 *(Spatial frame axiom).* *The* spatial frame axiom *is a special case of a spatial choice formula which states that, by default, a spatial property remains the same in the next step of a simulation. It takes the form (9) or (10):*

$$\{f_n(d_1, \ldots, d_{n-1}, s + 1) = x\} \leftarrow f_n(d_1, \ldots, d_{n-1}, s) = x, \tag{9}$$

$$\{Q_n(d_1, \ldots, d_{n-1}, s + 1)\} \leftarrow Q_n(d_1, \ldots, d_{n-1}, s). \tag{10}$$

where f_n is a parametric function, $x \in \mathbb{R}$, Q_n is a qualitative spatial relation, and $s \in \mathbb{N}$ represents a step in the simulation.

Corollary 1. *One spatial frame axiom for each parametric function and quali-*
tative spatial relation is enough to formalise the intuition that spatial properties,
by default, do not change over time.

The combination of spatial reasoning with stable model semantics and arithmetic
over the reals enables the operationalisation of a range of novel features within
the context of dynamic spatial reasoning. We present concrete examples of such
features in Sect. 5.

4   System Implementation

We present our implementation of ASPMT(QS) that builds on ASPMT2SMT [4] –
a compiler translating a tight fragment of ASPMT into SMT instances. Our
system consists of an additional module for spatial reasoning and Z3 as the SMT
solver. As our system operates on a tight fragment of ASPMT, input programs
need to fulfil certain requirements, described in the following section. As output,
our system either produces the stable models of the input programs, or states
that no such model exists.

4.1   Syntax of Input Programs

The input program to our system needs to be *f-plain* to use Theorem 1
from [2].

Definition 11 *(f-plain formula).* *Let f be a function constant. A first–order*
formula is called f-plain if each atomic formula:

– *does not contain f, or*
– *is of the form $f(\mathbf{t}) = u$, where $\mathbf{t}$ is a tuple of terms not containing f, and u*
  is a term not containing f.

Additionally, the input program needs to be *av-separated*, i.e. no variable occur-
ring in an argument of an uninterpreted function is related to the value variable
of another uninterpreted function via equality [4]. The input program is divided
into declarations of:

– `sorts` (data types);
– `objects` (particular elements of given types);
– `constants` (functions);
– `variables` (variables associated with declared types).

The second part of the program consists of clauses. ASPMT(QS) supports:

– connectives: `&`, `|`, `not`, `->`, `<-`, and
– arithmetic operators: `<`, `<=`, `>=`, `>`, `=`, `!=`, `+`, `=`, `*`, with their usual meaning.

Additionally, ASPMT(QS) supports the following as native / first-class entities:

- **sorts** for geometric objects types, e.g., `point`, `segment`, `circle`, `triangle`;
- parametric functions describing objects parameters e.g., $x(\texttt{point})$, $r(\texttt{circle})$;
- qualitative relations, e.g., `rccEC(circle, circle)`, `coincident(point, circle)`.

▷ **Example 1: combining topology and size.** Consider a program describing three circles a, b, c such that a is discrete from b, b is discrete from c, and a is a proper part of c, declared as follows:

```
:- sorts
   circle.
:- objects
   a, b, c      ::  circle.
:- constants
   .
:- variables
   C, C1, C2    ::  circle.

{x(C)=X}. {y(C)=X}. {r(C)=X}.
rccDR(a,b)=true. rccDR(b,c)=true. rccPP(a,c)=true.
```

ASPMT(QS) checks if the spatial relations are satisfiable. In the case of a positive answer, a parametric model and computation time are presented. The output of the above mentioned program is:

```
r(a) = 0.5        r(b) = 1.0        r(c) = 0.25
x(a) = 1.0        x(b) = 1.0        x(c) = 1.0
y(a) = 3.0        y(b) = 1.0        y(c) = 3.0
```

This example demonstrates that ASPMT(QS) is capable of computing *composition tables*, in this case the RCC–5 table for circles [25]. Now, consider the addition of a further constraint to the program stating that circles a, b, c have the same radius:

```
<- r(a)=R1 & r(b)=R2 & r(c)=R3 & (R1!=R2 | R2!=R3 | R1!=R3).
```

This new program is an example of combining different *types* of qualitative information, namely topology and size, which is a non-trivial research topic within the relation algebraic spatial reasoning community; relation algebraic-based solvers such as GQR [17,29] will not correctly determine inconsistencies in general for arbitrary combinations of different types of relations (orientation, shape, distance, etc.). In this case, ASPMT(QS) correctly determines that the spatial constraints are inconsistent:

```
UNSATISFIABLE; Z3 time in milliseconds: 10; Total time in milliseconds: 946
```

▷ **Example 2: combining topology and relative orientation.** Given three circles a, b, c let a be proper part of b, b discrete from c, and a in contact with c, declared as follows:

Fig. 1. Reasoning about consistent and refinement by combining topology and relative orientation.

```
:- sorts
   circle.
:- objects
   a, b, c     ::  circle.
:- constants
   .
:- variables
   C, C1, C2   ::  circle.

{x(C)=X}. {y(C)=X}. {r(C)=X}.
rccPP(a,b)=true. rccDR(b,c)=true. rccC(a,c)=true.
```

Given this basic qualitative information, ASPMT(QS) is able to refine the topological relations to infer that (Fig. 1a): (i) *a* must be a *tangential proper part* of *b* (ii) both *a* and *b* must be *externally connected* to *c*.

```
r(a)  = 1.0          r(b)  = 2.0          r(c)  = 1.0
x(a)  = 1.0          x(b)  = 0.0          x(c)  = 3.0
y(a)  = 0.0          y(b)  = 0.0          y(c)  = 0.0
rccTPP(a,b) = true   rccEC(a,c) = true    rccEC(b,c) = true
```

We then add an additional constraint that the centre of *a* is *left of* the segment between the centres *b* to *c*.

```
...
left_of(center(a),center(b),center(c)).
```

ASPMT(QS) determines that this is inconsistent, i.e., the centres must be *collinear* (Fig. 1b).

```
UNSATISFIABLE;
```

5   Empirical Evaluation and Examples

In this section we present an empirical evaluation of ASPMT(QS) in comparison with other existing spatial reasoning systems. The range of problems demonstrate the unique, non-monotonic spatial reasoning features that ASPMT(QS) provides beyond what is possible using other currently available systems. Table 1 presents run times obtained by Clingo – an ASP grounder and solver [18],

Table 1. Cumulative results of performed tests. "—" indicates that the problem can not be formalised, "^{I}" indicates that indirect effects can not be formalised, "^{D}" indicates that default rules can not be formalised.

Problem	Clingo	GQR	CLP($\mathcal{QS}$)	ASPMT(QS)
Growth	$0.004s^I$	$0.014s^{I,D}$	$1.623s^D$	0.396s
Motion	$0.004s^I$	$0.013s^{I,D}$	$0.449s^D$	15.386s
Attach I	$0.008s^I$	—	$3.139s^D$	0.395s
Attach II	—	—	$2.789s^D$	0.642s

GQR – a binary constraint calculi reasoner [17], CLP($\mathcal{QS}$) – a declarative spatial reasoning system [8] and our ASPMT(QS) implementation. Tests were performed on an Intel Core 2 Duo 2.00 GHZ CPU with 4 GB RAM running Ubuntu 14.04. The polynomial encodings of the topological relations have not been included here for space considerations.

5.1   Ramification Problem

The following two problems, *Growth* and *Motion*, were introduced in [5]. Consider the initial situation S_0 presented in Fig. 2, consisting of three cells: a, b, c, such that a is a non-tangential proper part of b: rccNTPP$(a, b, 0)$, and b is externally connected to c: rccEC$(b, c, 0)$.

Fig. 2. Indirect effects of $growth(a, 0)$ and $motion(a, 0)$ events.

▷ *Growth.* Let a grow in step S_0; the event growth$(a, 0)$ occurs and leads to a successor situation S_1. The direct effect of $growth(a, 0)$ is a change of a relation between a and b from rccNTPP$(a, b, 0)$ to rccEQ$(a, b, 1)$ (i.e. a is equal to b). No change of the relation between a and c is directly stated, and thus we must derive the relation rccEC$(a, c, 1)$ as an indirect effect.

▷ *Motion.* Let a move in step S_0; the event motion$(a, 0)$ leads to a successor situation S_1. The direct effect is a change of the relation rccNTPP$(a, b, 0)$ to rccTPP$(a, b, 1)$ (a is a tangential proper part of b). In the successor situation S_1 we must determine that the relation between a and c can only be either rccDC$(a, c, 1)$ or rccEC$(a, c, 1)$.

GQR provides no support for domain-specific reasoning, and thus we encoded the problem as two distinct qualitative constraint networks (one for each simulation step) and solved them independently i.e. with no definition of *growth* and *motion*. Thus, GQR is not able to produce any additional information about indirect effects. As Clingo lacks any mechanism for analytic geometry, we implemented the RCC8 composition table and thus it inherits the incompleteness of relation algebraic reasoning. While CLP(QS) facilitates the modelling of domain
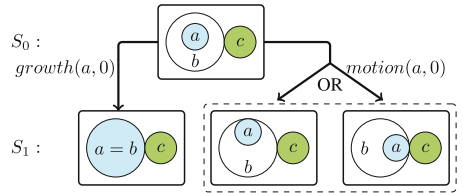
rules such as *growth*, there is no native support for default reasoning and thus we forced b and c to remain unchanged between simulation steps, otherwise all combinations of spatially consistent actions on b and c are produced without any preference (i.e. leading to the frame problem).

In contrast, ASPMT(QS) can express spatial inertia, and derives indirect effects directly from spatial reasoning: in the *Growth* problem ASPMT(QS) abduces that a has to be concentric with b in S_0 (otherwise a *move* event would also need to occur). Checking global consistency of scenarios that contain inter-dependent spatial relations is a crucial feature that is enabled by a support polynomial encodings and is provided only by CLP(QS) and ASPMT(QS).

5.2   Geometric Reasoning and the Frame Problem

In problems *Attachment I* and *Attachment II* the initial situation S_0 consists of three objects (circles), namely car, trailer and garage as presented in Fig. 3. Initially, the trailer is attached to the car: `rccEC(car, trailer, 0)`, `attached(car, trailer, 0)`. The successor situation S_1 is described by `rccTPP(car, garage, 1)`. The task is to infer the possible relations between the trailer and the garage, and the necessary actions that would need to occur in each scenario.

There are two domain-specific actions: the car can move, `move(car, X)`, and the trailer can be detached, `detach(car, trailer, X)` in simulation step X. Whenever the trailer is attached to the car, they remain `rccEC`. The car and the trailer may be either completely outside or completely inside the garage.

▷ *Attachment I.* Given the available topological information, we must infer that there are two possible solutions (Fig. 3); (a) the car was detached from the trailer and then moved into the garage: (b) the car, together with the *trailer* attached to it, moved into the garage:

▷ *Attachment II.* We are given additional geometric information about the objects' size: $r(car) = 2$, $r(trailer) = 2$ and $r(garage) = 3$. Case (b) is now inconsistent, and we must determine that the only possible solution is (a).

Fig. 3. Non-monotonic reasoning with additional geometric information.

These domain-specific rules require default reasoning: "*typically* the trailer remains in the same position" and "*typically* the trailer remains attached to the car". The later default rule is formalised in ASPMT(QS) by means of the spatial defaul.: The formalisation of such rules addresses the frame problem. GQR is not capable of expressing the domain-specific rules for detachment and attachment in *Attachment I* and *Attachment II*. Neither GQR nor Clingo are capable of reasoning with a combination of topological and numerical information, as required in *Attachment II*. As CLP(QS) cannot express default rules, we can not capture the notion that, for example, the trailer should typically
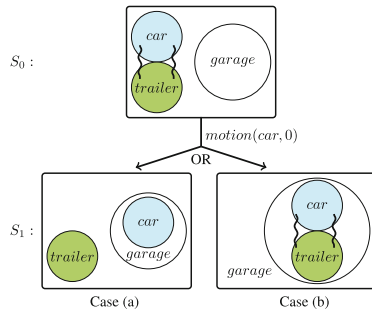
remain in the same position unless we have some explicit reason for determining that it moved; once again this leads to an exhaustive enumeration of all possible scenarios without being able to specify preferences, i.e. the frame problem, and thus CLP(QS) will not scale in larger scenarios.

The results of the empirical evaluation show that ASPMT(QS) is the only system that is capable of (a) non-monotonic spatial reasoning, (b) expressing domain-specific rules that also have spatial aspects, and (c) integrating both qualitative and numerical information. Regarding the greater execution times in comparison to CLP(QS), we have not yet implemented any optimisations with respect to spatial reasoning; this is one of the directions of future work.

6   Conclusions

We have presented ASPMT(QS), a novel approach for reasoning about spatial change within a KR paradigm. By integrating dynamic spatial reasoning within a KR framework, namely answer set programming (modulo theories), our system can be used to model behaviour patterns that characterise high-level processes, events, and activities as identifiable with respect to a general characterisation of commonsense *reasoning about space, actions, and change* [6,9]. ASPMT(QS) is capable of sound and complete spatial reasoning, and combining qualitative and quantitative spatial information when reasoning non-monotonically; this is due to the approach of encoding spatial relations as polynomial constraints, and solving using SMT solvers with the theory of real nonlinear arithmetic. We have demonstrated that no other existing spatial reasoning system is capable of supporting the key non-monotonic spatial reasoning features (e.g., spatial inertia, ramification) provided by ASPMT(QS) in the context of a mainstream knowledge representation and reasoning method, namely, answer set programming.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM **26**(11), 832–843 (1983)
2. Bartholomew, M., Lee, J.: Stable models of formulas with intensional functions. In: KR (2012)
3. Bartholomew, M., Lee, J.: Functional stable model semantics and answer set programming modulo theories. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 718–724. AAAI Press (2013)
4. Bartholomew, M., Lee, J.: System ASPMT2SMT: computing ASPMT theories by SMT solvers. In: Fermé, E., Leite, J. (eds.) JELIA 2014. LNCS, vol. 8761, pp. 529–542. Springer, Heidelberg (2014)

5. Bhatt, M.: (Some) Default and non-monotonic aspects of qualitative spatial reasoning. In: AAAI 2008 Technical reports, Workshop on Spatial and Temporal Reasoning, pp. 1–6 (2008)

6. Bhatt, M.: Reasoning about space, actions and change: a paradigm for applications of spatial reasoning. In: Qualitative Spatial Representation and Reasoning: Trends and Future Directions. IGI Global, USA (2012)

7. Bhatt, M., Guesgen, H., Wölfl, S., Hazarika, S.: Qualitative spatial and temporal reasoning: emerging applications, trends, and directions. Spat. Cogn. Comput. **11**(1), 1–14 (2011)

8. Bhatt, M., Lee, J.H., Schultz, C.: CLP(QS): a declarative spatial reasoning framework. In: Egenhofer, M., Giudice, N., Moratz, R., Worboys, M. (eds.) COSIT 2011. LNCS, vol. 6899, pp. 210–230. Springer, Heidelberg (2011)

9. Bhatt, M., Loke, S.: Modelling dynamic spatial systems in the situation calculus. Spat. Cogn. Comput. **8**(1), 86–130 (2008)

10. Bhatt, M., Wallgrün, J.O.: Geospatial narratives and their spatio-temporal dynamics: commonsense reasoning for high-level analyses in geographic information systems. ISPRS Int. J. Geo-Inf. **3**(1), 166–205 (2014)

11. Bouhineau, D.: Solving geometrical constraint systems using CLP based on linear constraint solver. In: Pfalzgraf, J., Calmet, J., Campbell, J. (eds.) AISMC 1996. LNCS, vol. 1138. Springer, Heidelberg (1996)

12. Bouhineau, D., Trilling, L., Cohen, J.: An application of CLP: checking the correctness of theorems in geometry. Constraints **4**(4), 383–405 (1999)

13. Ferraris, P.: Answer sets for propositional theories. In: Baral, C., Greco, G., Leone, N., Terracina, G. (eds.) LPNMR 2005. LNCS (LNAI), vol. 3662, pp. 119–131. Springer, Heidelberg (2005)

14. Ferraris, P., Lee, J., Lifschitz, V.: Stable models and circumscription. Artif. Intell. **175**(1), 236–263 (2011)

15. Frank, A.U.: Qualitative spatial reasoning with cardinal directions. In: Kaindl, H. (ed.) 7. Österreichische Artificial-Intelligence-Tagung/Seventh Austrian Conference on Artificial Intelligence. Informatik-Fachberichte, vol. 287, pp. 157–167. Springer, Heidelberg (1991)

16. Freksa, C.: Using orientation information for qualitative spatial reasoning. In: Frank, A.U., Formentini, U., Campari, I. (eds.) GIS 1992. LNCS, vol. 639, pp. 162–178. Springer, Heidelberg (1992)

17. Gantner, Z., Westphal, M., Wölfl, S.: GQR-A fast reasoner for binary qualitative constraint calculi. In: Proceedings of AAAI, vol. 8 (2008)

18. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Clingo= ASP+ control: Preliminary report. arXiv preprint arXiv:1405.3694 (2014)

19. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. ICLP/SLP, vol. 88, pp. 1070–1080 (1988)

20. Guesgen, H.W.: Spatial reasoning based on Allen's temporal logic. Technical report TR-89-049, International Computer Science Institute Berkeley (1989)

21. Lee, J.H.: The complexity of reasoning with relative directions. In: 21st European Conference on Artificial Intelligence (ECAI 2014) (2014)

22. Moratz, R.: Representing relative direction as a binary relation of oriented points. In: Brewka, G., Coradeschi, S., Perini, A., Traverso, P. (eds.) ECAI. Frontiers in Artificial Intelligence and Applications, vol. 141, pp. 407–411. IOS Press (2006)

23. Pesant, G., Boyer, M.: QUAD-CLP(R): adding the power of quadratic constraints. In: Borning, A. (ed.) PPCP 1994. LNCS, vol. 874, pp. 95–108. Springer, Heidelberg (1994)

24. Pesant, G., Boyer, M.: Reasoning about solids using constraint logic programming. J. Autom. Reasoning **22**(3), 241–262 (1999)
25. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: KR, vol. 92, pp. 165–176 (1992)
26. Schultz, C., Bhatt, M.: Towards a declarative spatial reasoning system. In: 20th European Conference on Artificial Intelligence (ECAI 2012) (2012)
27. Schultz, C., Bhatt, M.: Declarative spatial reasoning with boolean combinations of axis-aligned rectangular polytopes. In: ECAI 2014–21st European Conference on Artificial Intelligence, pp. 795–800 (2014)
28. Varzi, A.C.: Parts, wholes, and part-whole relations: the prospects of mereotopology. Data Knowl. Eng. **20**(3), 259–286 (1996)
29. Wölfl, S., Westphal, M.: On combinations of binary qualitative constraint calculi. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence. IJCAI 2009, Pasadena, California, USA, 11–17 July 2009, pp. 967–973 (2009)