

# Answer Set Programming Modulo ‘Space-Time’

Carl Schultz<sup>1</sup>, Mehul Bhatt<sup>2,3</sup>, Jakob Suchan<sup>3</sup>, and Przemysław Andrzej Wałęga<sup>4</sup>

Spatial Reasoning. [www.spatial-reasoning.com](http://www.spatial-reasoning.com)

<sup>1</sup> DIGIT, Aarhus University, Denmark, <sup>2</sup> Örebro University, Sweden

<sup>3</sup> University of Bremen, Germany, <sup>4</sup>University of Oxford, <sup>4</sup>University of Warsaw

**Abstract.** We present ASP Modulo ‘Space-Time’, a declarative representational and computational framework to perform commonsense reasoning about regions with both spatial and temporal components. Supported are capabilities for mixed qualitative-quantitative reasoning, consistency checking, and inferring compositions of space-time relations; these capabilities combine and synergise for applications in a range of AI application areas where the processing and interpretation of spatio-temporal data is crucial. The framework and resulting system is the only general KR-based method for declaratively reasoning about the dynamics of ‘space-time’ regions as first-class objects.

## 1 INTRODUCTION

Answer Set Programming (ASP) has emerged as a robust declarative problem solving methodology with tremendous application potential [9, 10, 18]. Most recently, there has been heightened interest to extend ASP in order to handle specialised domains and application-specific knowledge representation and reasoning (KR) capabilities. For instance, ASP Modulo Theories (ASPMT) go beyond the propositional setting of standard answer set programs by the integration of ASP with Satisfiability Modulo Theories (SMT) thereby facilitating reasoning about continuous domains [2, 9, 13]; using this approach, integrating knowledge sources of *heterogeneous semantics* (e.g., infinite domains) becomes possible. Similarly, systems such as CLINGCON [8] combine ASP with specialised constraint solvers to support arithmetic constraints over integers. Other most recent extensions include the ASPMT founded *non-monotonic spatial reasoning* extensions in ASPMT(QS) [19] and ASP modulo *acyclicity* [5]. Indeed, being rooted in KR, in particular non-monotonic reasoning, ASP can theoretically characterise —and promises to serve in practice as— a modern foundational language for several domain-specific AI formalisms, and offer a uniform computational platform for solving many of the classical AI problems involving planning, explanation, diagnosis, design, decision-making, control [6, 18]. In this line of research, this paper presents ASP Modulo ‘Space-Time’, a specialised formalism and computational backbone enabling generalised commonsense reasoning about ‘*space-time objects*’ and their spatio-temporal dynamics [4] directly within the answer set programming paradigm.

**Reasoning about ‘Space-Time’ (Motion).** Imagine a moving object within 3D space. Here, the complete trajectory of motion of the moving object within a space-time localisation framework constitutes a 4D space-time history consisting of both spatial and temporal components – i.e., it is a region in *space-time* (Fig. 1). Regions in *space*, *time*,

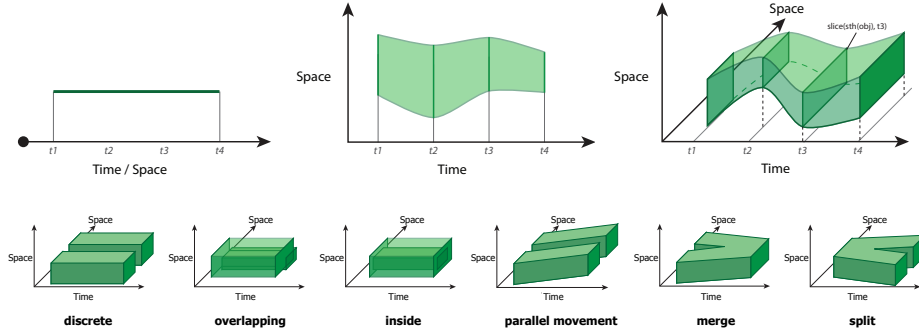


Fig. 1: Space-Time Histories in 1D and 2D; Spatio-temporal patterns and events, i.e., discrete, overlapping, inside, parallel movement, merge, and split.

and *space-time* have been an object of study across a range of disciplines such as ontology, cognitive linguistics, conceptual modeling, KR (particularly qualitative spatial reasoning), and spatial cognition and computation. Spatial knowledge representation and reasoning can be classified into two groups: topological and positional calculi [1, 14]. With topological calculi such as the Region Connection Calculus (RCC) [16], the primitive entities are spatially extended regions of space, and could be arbitrarily (but uniformly) dimensioned space-time histories. For the case of ‘space-time’ representations, the main focus in the state of the art has been on axiom systems (and the study of properties resulting therefrom) aimed at pure *qualitative reasoning*. In particular, axiomatic characterisations of mereotopologically founded theories with spatio-temporal regions as primitive entities are very well-studied [11, 15]. Furthermore, the dominant method and focus within the field of spatial representation and reasoning — be it for topological or positional calculi — has been primarily on relational-algebraically founded semantics [14] in the absence of (or by discarding available) quantitative information. Pure qualitative spatial reasoning is very valuable, but it is often counterintuitive to not utilise or discard quantitative data if it is available (numerical information is typically available in domains involving sensing, interaction, interpretation, and control).

## 2 ASP MODULO ‘SPACE-TIME’

**Spatial Domains.** Spatial entities in our spatio-temporal domain ( $ST$ ) include *points* and *simple polygons*: a  $2D$  point is a pair of reals  $x, y$ ; a *simple polygon*  $P$  is defined by a list of  $n$  vertices (points)  $p_0, \dots, p_{n-1}$  such that the boundary is non-self-intersecting, i.e., no two edges of the polygon intersect. We denote the number of vertices in  $P$  by  $|P|$ . A polygon is *ground* if all of its vertices are assigned real values. A translation vector  $t$  is a pair of reals  $(t_x, t_y)$ . Given a point  $p = (x, y)$  and a translation vector  $t$  let  $p + t := (x + t_x, y + t_y)$ . A *translation* is a ternary relation between two polygons  $P, Q$  and a translation vector  $t$  such that:  $|P| = |Q| = n$  and  $p_i = q_i + t$  where  $p_i$  is the  $i^{\text{th}}$  vertex in  $P$  and  $q_i$  is the  $i^{\text{th}}$  vertex in  $Q$ , for  $0 \leq i < n$ . A translation vector  $t$  is *ground* if  $t_x, t_y$  are assigned real values, otherwise it is *unground*.

**Temporal domain  $\mathcal{T}$ .** The temporal dimension is constituted by an infinite set of time points — each time point is a real number. The time-line is given by a linear ordering  $<$  of time-points.

Table 1: Relations between  $ST$  regions  $s_1, s_2$  over time interval  $I = [t_0, t_N]$ ;  $t, t'$  range over  $I$  with  $t \leq t'$ ;  $\text{reverse}(R)$  denotes the definition of relation  $R$  with reversed temporal ordering,  $t' \leq t$ ;  $p_i(t_j)$  is the centre point of  $s_i$  at  $t_j$ ;  $\Delta$  is the Euclidean distance between two points;  $\alpha$  is a user-specified temporal threshold.

Relation	Definition
<b>Topology</b>	
disconnects (DC)	$\forall t \text{ dc}(s_1(t), s_2(t))$
discrete from (DR)	$\forall t \text{ dr}(s_1(t), s_2(t))$
part of (P)	$\forall t \text{ p}(s_1(t), s_2(t))$
non-tangential proper part (NTPP)	$\forall t \text{ ntp}(s_1(t), s_2(t))$
equal (EQ)	$\forall t \text{ eq}(s_1(t), s_2(t))$
contacts (C)	$\exists t \text{ c}(s_1(t), s_2(t))$
overlaps (O)	$\exists t \text{ o}(s_1(t), s_2(t))$
partially overlaps (PO)	$\exists t \text{ po}(s_1(t), s_2(t))$
externally connects (EC)	$\text{dr}(s_1, s_2) \wedge \exists t \text{ ec}(s_1(t), s_2(t))$
proper part (PP)	$\text{p}(s_1, s_2) \wedge \exists t \text{ pp}(s_1(t), s_2(t))$
tangential proper part (TPP)	$\text{p}(s_1, s_2) \wedge \exists t \text{ tpp}(s_1(t), s_2(t))$
split	$\text{p}(s_1(t_0), s_2(t_0)) \wedge \text{dc}(s_1(t_N), s_2(t_N))$
merge	$\text{dc}(s_1(t_0), s_2(t_0)) \wedge \text{p}(s_1(t_N), s_2(t_N))$
<b>Size</b>	
fixed size	$\forall t \forall t' (\text{area}(s(t)) = \text{area}(s(t')))$
grows	$\neg \text{fixed\_size}(s) \wedge \forall t \forall t' (\text{area}(s(t)) \leq \text{area}(s(t')))$
shrinks	$\text{reverse}(\text{grows}(s_1, s_2))$
<b>Movement</b>	
moves	$\exists t \exists t' p(t) \neq p(t')$
move parallel	$\text{moves}(s_1) \wedge \forall t \forall t' (p_2(t) - p_1(t)) = (p_2(t') - p_1(t'))$
towards	$\text{moves}(s_1) \wedge \neg \text{moves\_parallel}(s_1, s_2) \wedge \forall t \forall t' \Delta(p_1(t), p_2(t)) \geq \Delta(p_1(t'), p_2(t'))$
away	$\text{reverse}(\text{towards}(s_1, s_2))$
follows	$\forall t' \exists t \text{ duration}(t, t') \leq \alpha$ $\wedge \Delta(p_1(t), p_2(t)) > \Delta(p_1(t'), p_2(t))$ $\wedge \Delta(p_1(t), p_2(t)) < \Delta(p_1(t), p_2(t'))$

**ST Histories.** If we treat time as an additional dimension, then we can represent a moving two-dimensional spatial object  $s$  as a three-dimensional object in space-time. At each time point, the corresponding space-time region of  $s$  has a 2D spatial representation (a spatial *slice*). The space-time object is formed by taking all such slices over time. An *ST object*  $o \in O$  is a variable associated with an ST domain  $D$  (e.g., the domain of 2D polygons over time). An *instance* of an object  $i \in D$  is an element from the domain. Given  $O = \{o_1, \dots, o_n\}$ , and domains  $D_1, \dots, D_n$  such that  $o_i$  is associated with domain  $D_i$ , then a *configuration* of objects  $\psi$  is a one-to-one mapping between object variables and instances from the domain,  $\psi(o_i) \in D_i$ . For example, a variable  $o_1$  is associated with the domain  $D_1$  of moving 2D points over time. An *ST point* moving in a straight line starting at spatial coordinates  $(0, 0)$  at time 0 and arriving at 2D spatial coordinates  $(10, 0)$  at time 1 is an instance of  $D_1$ . A configuration is defined that maps  $o_1$  to a 3D line with end points  $(0, 0, 0), (10, 0, 1)$ , i.e.,  $\psi(o_1) = [(0, 0, 0), (10, 0, 1)]$ .

**ST Relations.** Let  $D_1, \dots, D_n$  be spatio-temporal domains. A spatio-temporal relation  $r$  of arity  $n$  ( $n > 0$ ) is defined as  $r \subseteq D_1 \times \dots \times D_n$ . That is, each spatio-temporal relation is an equivalence class of instances of *ST* objects. Given a set of objects  $O$ , a relation  $r$  of arity  $n$  can be asserted as a constraint that must hold between objects  $o_1, \dots, o_n \in O$ , denoted  $r(o_1, \dots, o_n)$ . The constraint  $r(o_1, \dots, o_n)$  is satisfied by configuration  $\psi$  if  $(\psi(o_1), \dots, \psi(o_n)) \in r$ . For example, if *pp* is a topological relation *proper part*, and  $O = \{o_1, o_2\}$  is a set of moving polygon objects, then  $\text{pp}(o_1, o_2)$  is the constraint that moving polygon  $o_1$  is a proper part of  $o_2$ .

Table 1 presents definitions for  $ST$  relations that hold between  $s_1$  and  $s_2$ , where  $t, t'$  range over a (dense) time interval with start and end time points  $t_0$  and  $t_N$  in which  $s_1$  and  $s_2$  occur and  $t \leq t'$ . We define mereotopological relations using the Region Connection Calculus (RCC) [16]: all spatio-temporal RCC relations between  $ST$  regions are defined based on the RCC relations of their slices. An  $ST$  region  $s_1$  *follows*  $ST$  region  $s_2$  if, at each time step,  $s_1$  moves towards a previous location of  $s_2$ , and  $s_2$  moves away from a previous location of  $s_1$ <sup>1</sup>. We formalise the semantics of spatial reasoning by encoding qualitative spatial relations as systems of polynomial equations and inequalities. The task of determining whether a set of spatial relations is consistent is then equivalent to determining whether the set of polynomial constraints are satisfiable [3, 19]. Integrating logic-based constraint solving with arithmetic constraints has a long and rich history – see, e.g., [12].

## 2.1 Spatio-Temporal Consistency

Consider the topological *disconnected* relation. There is no polygon that is *disconnected* from itself, i.e., the relation is *irreflexive*. The following algebraic properties of  $ST$  relations are expressed as ASP rules and constraints: reflexivity, irreflexivity, symmetry, asymmetry, converse, implication, mutual (pair-wise) inconsistency, and transitive inconsistency. We have automatically derived these properties using our polynomial constraint solver *a priori* and generated the corresponding ASP rules. A violation of these properties corresponds to *3-path inconsistency* [14], i.e., there does not exist any combination of polygons that can violate these properties. In particular, a total of 1586 space-time constraints result.

**Ground Polygons.** We can determine whether a given  $ST$  relation  $r$  holds between two ground polygons  $P, Q$  by directly checking whether the corresponding polynomial constraints are satisfied, i.e. polynomial constraint variables are replaced by the real values assigned to the ground polygon vertices. This is accomplished during the *grounding* phase of ASP. For instance, two ground polygons are *disconnected* if the distance between them is greater than zero.

**Unground Translation.** Given ground polygons  $P_0, P_1$ , *unground* polygon  $P'_0$ , and unground translation  $t = (t_x, t_y)$ , let  $P'_0$  be a  $t$  translation of  $P_0$  such that  $r$  holds between  $P'_0, P_1$ . The (exact) set of real value pairs that can be assigned to  $(t_x, t_y)$  such that  $P'_0, P_1$  satisfy  $r$  is precisely determined using the Minkowski sum method [20]; we refer to this set as the *solution set* of  $t$  for  $r$ . Given  $n$  ground polygons  $P_1, \dots, P_n$ , and  $n$  relations  $r_1, \dots, r_n$  such that relation  $r_i$  is asserted to hold between polygon  $P_0, P_i$ , for  $1 \leq i \leq n$ , let  $M_i$  be the solution set of  $t$  for  $r_i$ . The conjunction of relations  $r_1, \dots, r_n$  is consistent if the *intersection* of solution sets  $M_1, \dots, M_n$  is non-empty. Computing and intersecting solution sets is accomplished during the *grounding* phase of ASP.

**System implementation.** We have implemented our  $ST$  reasoning module in Clingo (v5.1.0) [7] based on an integration with specialised polynomial constraint solvers via numerical optimisation and Satisfiability Modulo Theories supporting real arithmetic.

<sup>1</sup> We introduce a user-specified maximum duration threshold  $\alpha$  between these two time points to prevent unwanted scenarios being defined as *follows* events such as  $s_1$  taking one step towards  $s_2$  and then stopping while  $s_2$  continues to move away from  $s_1$ .

Table 2:  $ST$  entities and relation predicates.

Predicate	Description
<b><math>ST</math> Entities</b>	
<code>polygon(Pg, (X1, Y2, ..., Xn, Yn))</code>	Polygon Pg has $n$ ground vertices $(x_1, y_1), \dots, (x_n, y_n)$ .
<code>translation(Pg1, Pg2)</code>	Polygon Pg2 is an unground translation of Pg1.
<code>st_object(E)</code>	E is a spatio-temporal entity.
<code>st_object(E, at(Time), id(Pg))</code>	2D polygon Pg is a spatial <i>slice</i> of spatio-temporal entity E at time point Time.
<b><math>ST</math> Relations</b>	
<code>spacetime(STAspect, E, time(T1, T2))</code>	Derive unary ST relations for STAspect (topology, size, or movement) for entity E from time T1 to T2.
<code>spacetime(STAspect, E1, E2, time(T1, T2))</code>	Derive binary ST relations for STAspect (topology, size, or movement) between entities E1, E2 from time T1 to T2.
<code>topology(Rel, E1, E2, time(T1, T2))</code>	Topological relation Rel is asserted to hold between ST entities E1, E2 from time T1 to T2.
<code>size(Rel, E1, E2, time(T1, T2))</code>	Size relation Rel is asserted to hold between ST entities E1, E2 from time T1 to T2.
<code>movement(Rel, E, time(T1, T2))</code>	Unary movement relation Rel is asserted to hold for ST entity E from time T1 to T2.
<code>movement(Rel, E1, E2, time(T1, T2))</code>	Binary movement relation Rel is asserted to hold between ST entities E1, E2 from time T1 to T2.
<code>spatial(witness, E, EWitness)</code>	Ground entity EWitness is a consistent witness for unground entity E.

All models produced by our system are models in the usual ASP sense that are also *spatially consistent*. We install a *hook* in Clingo's answer set search for checking whether complete and partial assignments are spatially consistent using our spatial reasoning module  $ST$ . If a spatial inconsistency is detected, the system *backtracks* the search (using Clingo's native backtracking mechanism) and creates a spatial conflict clause (no-good) using Clingo's learnt constraint mechanism. Thus, we leverage from all features provided by the standard Clingo solver with an extension to support spatio-temporal reasoning.

### 3 REASONING WITH ASP MODULO SPACE-TIME

Table 2 presents our system's predicate interface. Our system provides special predicates for (1) declaring spatial objects, and (2) relating objects spatio-temporally. Each  $ST$  object is represented with `st_object/3` relating the identifier of the  $ST$  entity, time point of this slice, and identifier of the associated geometric representation.

```
st_object(EntityId, at(Time), id(PolygonId)).
```

Polygons are represented using the `polygon/2` predicate that relates an identifier of the geometric representation with a list of  $x, y$  vertex coordinate pairs, e.g.:

```
polygon(id(pgBx2_0), (268, 0, 303, 0, 303, 5)).
```

**Deriving  $ST$  relations.** The predicate `spacetime/3` is used to specify the entities between which  $ST$  relations should be derived:

```
% derive properties of entity e1 during time 25 to 75
spacetime(movement, e1, time(25, 75)).
% derive relations between entities e5, e6 time 1 to 10
spacetime(movement, e5, e6, time(1, 10)).
% derive relations between all entities for time 10 to 20
spacetime(movement, Entity1, Entity2, time(1, 10)):-
    st_object(Entity1, _, _), st_object(Entity2, _, _).
```

**Purely qualitative reasoning.** If no geometric information for slices is given then our system satisfies 3-consistency, e.g., the following program includes transitively inconsistent spatio-temporal relations:

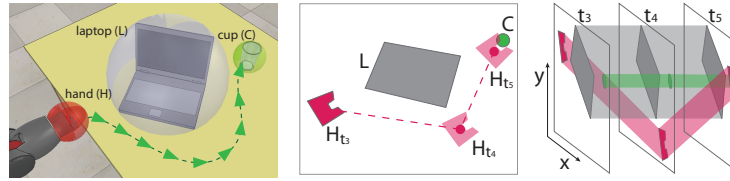


Fig. 2: Application in cognitive robotics – computing a motion trajectory.

```
st_object(s1). st_object(s2). st_object(s3).
topology(ntpp,s1,s2). topology(pp,s2,s3). topology(dc,s1,s3).
UNSATISFABLE
```

**Mixed qualitative-numerical reasoning.** A new  $ST$  object can be specified that consists of *translated* slices of a given  $ST$  object. Our system determines whether translations exist that satisfy all given spatio-temporal constraints. Our system produces the solution set and a spatial witness that minimises the translation distance.

```
translation(st1, translated_st1).
topology(pp, translated_st1, st2).
spatial(witness, translated_st1, witness_st1).
```

► **Application Example: Motion planning.** We show how  $ST$  regions can be used for motion planning, e.g., in robotic manipulation tasks using abduction.

**Example.** *An agent (a robot with a manipulator) is at a desk in front of a laptop. A cup of coffee is positioned behind the laptop and the agent wants to get the cup of coffee without the risk of spilling the coffee on the laptop. The agent should not hit the computer while performing the task.*

This task requires abducting intermediate states that are consistent with the domain constraints. We model the laptop, hand, and cup from a top-down perspective as  $ST$  regions with polygonal slices, and give the initial shapes.

```
%% domain objects
desk_object(laptop). desk_object(hand). desk_object(cup).
```

The initial configuration is given for time 0:

```
%% polygonal shapes of objects
polygon(shape(laptop), (0,0, ...)).
polygon(shape(hand), (-105, 3, ...)).
polygon(shape(cup), (205, 54, ...)).
%% initial position of objects at time 0
st_object(Object, at(0), id(shape(Object))) :- desk_object(Object).
```

We model the scenario from time 0 to 2.

```
%% modelling two time steps
time(1..2).
%% it is possible that objects move at each time step
{moves(Object, at(T))} :- desk_object(Object), time(T).
%% objects that move are represented by an (unground) translation of their polygon
translation(shape(Object), translated_shape(Object, at(T))) :- moves(Object, at(T)).
%% at the end of the time step we need a witness of moving objects
spatial(witness, translated_shape(Object, at(T)),
        shape(Object, at(T))) :- moves(Object, at(T)).
%% slice of moving object at time T is a translated polygon
st_object(Object, at(T), id(translated_shape(Object, at(T)))) :-
    moves(Object, at(T)).
%% slice of stationary object at time T is polygon from last time step
st_object(Object, at(T), id(shape(Object, at(LastT)))) :-
    desk_object(Object), time(T), LastT = T - 1, not moves(Object, at(T)).
```

The goal is for the hand to make contact with the cup:

```
topology(c, hand, cup, time(1,2)).
```

We model default domain assumptions, e.g., the cup does not move by default. We express this by assigning costs to interpretations where objects move.

```
cost(0,Object) :- Object = (cup; laptop), time(T), ~moves(Object, at(T)).
cost(1,Object) :- Object = (cup; laptop), time(T), moves(Object, at(T)).
#minimize{ C, X : cost(C,X)}.
```

The spatio-temporal constraints for planning the motion trajectory are that the hand and cup must remain disconnected from the laptop.

```
topology(dc, laptop, hand, time(0,2)). topology(dc, laptop, cup, time(0,2)).
```

Our system finds a consistent and optimal answer set where neither the laptop nor cup move in the period before the robot hand has made contact with the cup. Given the spatio-temporal constraints in this optimal answer set, our system then produces a consistent motion trajectory witness of the solution set (Fig. 2).

## 4 SUMMARY AND RELATED WORK

ASP Modulo extensions for handling specialised domains and abstraction mechanisms provides a powerful means for the utilising ASP as a foundational knowledge representation and reasoning (KR) method for a wide-range of application contexts. This approach is clearly demonstrated in work such as ASPMT [2, 9, 13], CLINGCON [8], and ASPMT(QS) [19]. Most closely related to our research is the ASPMT founded *non-monotonic spatial reasoning* system ASPMT(QS) [19]. Whereas ASPMT(QS) provides a valuable blueprint for the integration and formulation of geometric and spatial reasoning within answer set programming modulo theories, the developed system is a first-step and lacks support for a rich spatio-temporal ontology or an elaborate characterisation of complex ‘space-time’ objects as native (the focus there has been on enabling non-monotonicity with a basic spatial and temporal ontology). Furthermore (1) we generate *all* spatially consistent models compared to only one model in the standard ASPMT pipeline; (2) we compute optimal answer sets, e.g., add support preferences, which allows us to rank models, specify weak constraints; (3) we support quantification of space-time regions. The outlook of this work is geared towards enhancing the application of the developed specialised ASP Modulo Space-Time component specifically for non-monotonic spatio-temporal reasoning about large datasets in the domain of visual stimulus interpretation [18], as well as constraint-based motion control in the domain of home-based and industrial robotics [17].

**Acknowledgements.** This work was partially supported by the Germany Research Foundation (DFG) as part of the CRC EASE; and partially by the NCN grant 2016/23/N/HS1/02168.

## Bibliography

- [1] Aiello, M., Pratt-Hartmann, I., van Benthem, J.: Handbook of spatial logics. Springer (2007) [2](#)
- [2] Bartholomew, M., Lee, J.: System aspmt2smt: Computing ASPMT Theories by SMT Solvers. In: Logics in Artificial Intelligence, pp. 529–542. Springer (2014) [1](#), [7](#)
- [3] Bhatt, M., Lee, J.H., Schultz, C.: CLP(QS): A Declarative Spatial Reasoning Framework. In: COSIT 2011 - Spatial Information Theory, pp. 210–230. Springer-Verlag (2011) [4](#)
- [4] Bhatt, M., Loke, S.: Modelling dynamic spatial systems in the situation calculus. Spatial Cognition and Computation **8**(1), 86–130 (2008) [1](#)
- [5] Bomanson, J., Gebser, M., Janhunen, T., Kaufmann, B., Schaub, T.: Answer set programming modulo acyclicity\*. Fundam. Inform. **147**(1), 63–91 (2016) [1](#)
- [6] Erdem, E., Gelfond, M., Leone, N.: Applications of answer set programming. AI Magazine **37**(3), 53–68 (2016) [1](#)
- [7] Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: *Clingo* = ASP + control: Preliminary report. In: Leuschel, M., Schrijvers, T. (eds.) Technical Communications of ICLP. vol. 14(4-5) (2014), theory and Practice of Logic Programming, Online Supplement [4](#)
- [8] Gebser, M., Ostrowski, M., Schaub, T.: Constraint answer set solving. In: Hill, P., Warren, D. (eds.) In Proc. 12th Int. Conf on Logic Programming (ICLP’09). LNCS, vol. 5649, pp. 235–249. Springer-Verlag (2009) [1](#), [7](#)
- [9] Gelfond, M.: Answer sets. Handbook of knowledge representation **1**, 285 (2008) [1](#), [7](#)
- [10] Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: ICLP/SLP. vol. 88, pp. 1070–1080 (1988) [1](#)
- [11] Hazarika, S.M.: Qualitative spatial change: space-time histories and continuity. Ph.D. thesis, The University of Leeds (2005) [2](#)
- [12] Kanellakis, P.C., Kuper, G.M., Revesz, P.Z.: Constraint query languages (preliminary report). In: In Proc. 9th ACM Symposium on Principles of Database Systems. pp. 299–313. ACM (1990) [4](#)
- [13] Lee, J., Meng, Y.: Answer set programming modulo theories and reasoning about continuous changes. In: IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013 (2013) [1](#), [7](#)
- [14] Ligozat, G.: Qualitative Spatial and Temporal Reasoning. Wiley-ISTE (2011) [2](#), [4](#)
- [15] Muller, P.: A qualitative theory of motion based on spatio-temporal primitives. KR **98**, 131–141 (1998) [2](#)
- [16] Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. KR **92**, 165–176 (1992) [2](#), [4](#)
- [17] Suchan, J., Bhatt, M.: Deep semantic abstractions of everyday human activities - on commonsense representations of human interactions. In: ROBOT 2017: Third Iberian Robotics Conference - Volume 1, Seville, Spain, November 22-24, 2017. Advances in Intelligent Systems and Computing, vol. 693, pp. 477–488. Springer (2017) [7](#)
- [18] Suchan, J., Bhatt, M., Wałęga, P.A., Schultz, C.: Visual Explanation by High-Level Abduction: On Answer-Set Programming Driven Reasoning About Moving Objects. In: McIlraith, S.A., Weinberger, K.Q. (eds.) In Proc. 32nd AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018. AAAI Press (2018) [1](#), [7](#)
- [19] Wałęga, P.A., Bhatt, M., Schultz, C.: ASPMT(QS): Non-Monotonic Spatial Reasoning with Answer Set Programming Modulo Theories. In: LPNMR: Logic Programming and Non-monotonic Reasoning - 13th Int. Conf. Lexington, KY, USA (2015) [1](#), [4](#), [7](#)
- [20] Wallgrün, J.O.: Topological adjustment of polygonal data. In: Advances in Spatial Data Handling, pp. 193–208. Springer (2013) [4](#)