Integrity Constraints for Probabilistic Spatio-Temporal Knowledgebases

Francesco Parisi[1] John Grant[2]

[1]Department of Informatics, Modeling, Electronics and System Engineering,
University of Calabria, Italy,
email: fparisi@dimes.unical.it
[2]Department of Computer Science and UMIACS,
University of Maryland, College Park, USA,
email: grant@cs.umd.edu

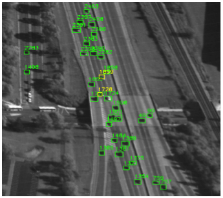
8^{th} International Conference on Scalable Uncertainty Management
St. Anne's College, Oxford, UK,
September 15-17, 2014

Tracking moving objects (1/2)

- Tracking moving objects is fundamental in several application contexts (e.g. environment protection, product traceability, traffic monitoring, mobile tourist guides, analysis of animal behavior, etc.)

http://www.merl.com/publications/TR2008-010

http://www.edimax.com/au/



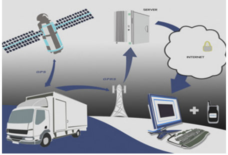
http://iris.usc.edu/people/medioni/current_research.html

http://www.i3b.org/content/wildlife-behavior

http://www.science20.com/news_articles/german_research_center_artificial_intelligence_smart_eye_tracking_glass

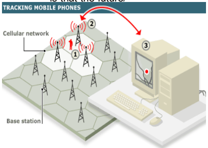
Tracking moving objects (2/2)

- Location estimation techniques have limited accuracy and precision
  - limitations of technologies used (e.g. GPS, Cellular networks, WiFi, Bluetooth, RFID, etc.)
  - limitations of the estimation methods (e.g., proximity to antennas, triangulation, signal strength sample map, dead reckoning, etc.)

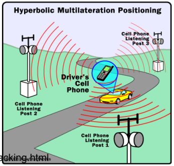


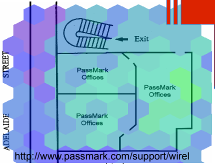
http://www.nitrobahn.com/conceptz/self-driving-cars-is-that-the-future/

http://www.ayantra.com/traffic-control-monitoring.html



http://www.gksoft.in/2014/07/mobile-phone-tracking.html





http://www.passmark.com/support/wireless_coverage_map.html

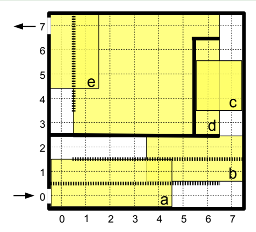
object inside a region at a time with (uncertain) probability

SPOT framework

- SPOT : a declarative framework for the representation and processing of probabilistic spatio-temporal data with uncertain probabilities [Parker, Subrahmanian, Grant. TKDE '07]
- A SPOT database is a set of atoms $loc(id, r, t)[\ell, u]$
- $loc(id, r, t)[\ell, u]$ means that "object id is/was/will be inside region r at time t with probability in the interval $[\ell, u]$".

Example

$loc(id_1, a, 1)[.4, .7]$
$loc(id_1, b, 1)[.4, .9]$
$loc(id_1, c, 9)[.9, 1]$
$loc(id_1, d, 15)[.6, 1]$
$loc(id_1, e, 18)[.7, 1]$
$loc(id_2, b, 2)[.5, .9]$
$loc(id_2, c, 12)[.9, 1]$
...

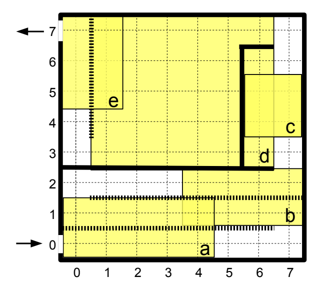


Limits of SPOT DBs

- Not general enough to represent additional knowledge concerning constraints on the movements of objects

Example

- *There cannot be two distinct objects in region c at any time point between* 1 *and* 20
- *No object can reach region e starting from region a in less than* 10 *time points*
- *Object id can go away from region c only if it stayed there for at least* 2 *time points*

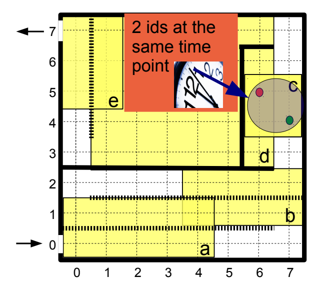


Limits of SPOT DBs

- Not general enough to represent additional knowledge concerning constraints on the movements of objects

Example

- *There cannot be two distinct objects in region c at any time point between* 1 *and* 20
- *No object can reach region e starting from region a in less than* 10 *time points*
- *Object id can go away from region c only if it stayed there for at least* 2 *time points*

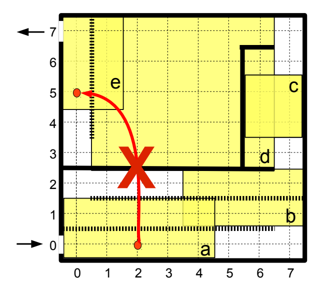


Limits of SPOT DBs

- Not general enough to represent additional knowledge concerning constraints on the movements of objects

Example

- *There cannot be two distinct objects in region c at any time point between* 1 *and* 20
- *No object can reach region e starting from region a in less than* 10 *time points*
- *Object id can go away from region c only if it stayed there for at least* 2 *time points*

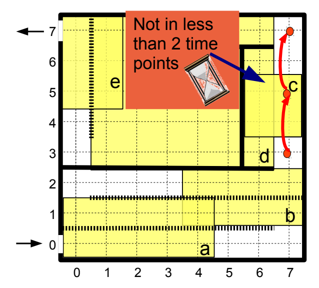


Limits of SPOT DBs

- Not general enough to represent additional knowledge concerning constraints on the movements of objects

Example

- *There cannot be two distinct objects in region c at any time point between 1 and 20*
- *No object can reach region e starting from region a in less than 10 time points*
- *Object id can go away from region c only if it stayed there for at least 2 time points*



Probabilistic spatio-temporal KBs

- A probabilistic spatio-temporal (PST) knowledgebase (KB) consists of

1) atomic statements, such as those representable in the SPOT framework
2) *spatio-temporal denial formulas*, a general class of formulas expressing constraints on moving objects

- Formal semantics, in terms of worlds, interpretations, and models
- Complexity of checking consistency of PST KBs
  - NP-complete in general
  - Mixed-binary linear programming algorithm providing sufficient conditions for checking consistency
  - A tractable case
- Using consistency checking for answering queries in PST KBs

Probabilistic spatio-temporal KBs

- A probabilistic spatio-temporal (PST) knowledgebase (KB) consists of

1) atomic statements, such as those representable in the SPOT framework
2) *spatio-temporal denial formulas*, a general class of formulas expressing constraints on moving objects

- Formal semantics, in terms of worlds, interpretations, and models
- Complexity of checking consistency of PST KBs
  - NP-complete in general
  - Mixed-binary linear programming algorithm providing sufficient conditions for checking consistency
  - A tractable case
- Using consistency checking for answering queries in PST KBs

Outline

PST atoms

- Notation: *ID* is the set of objects identifiers, *Space* is a grid of $N \times N$ points, T is a time interval

Definition (st-atom)

A *spatio-temporal atom* (*st-atom*) is of the form $loc(X, Y, Z)$, where:

- X is a variable ranging over *ID*, or a constant $id \in ID$;
- Y is a variable ranging over $\mathcal{P}(Space)$, or a constant $r \subseteq Space$
- Z is a variable ranging over T, or a constant $t \in T$.

Definition (PST atom – SPOT atom in the previous framework)

A PST *atom* is a ground st-atom $loc(id, r, t)$ annotated with a probability interval $[\ell, u] \subseteq [0, 1]$ – denoted as $loc(id, r, t)[\ell, u]$.

- $loc(id, r, t)[\ell, u]$ says that object *id* is/was/will be inside region r at time t with probability in the interval $[\ell, u]$

PST atoms

- Notation: ID is the set of objects identifiers, $Space$ is a grid of $N \times N$ points, T is a time interval

Definition (st-atom)

A *spatio-temporal atom* (*st-atom*) is of the form $loc(X, Y, Z)$, where:

- X is a variable ranging over ID, or a constant $id \in ID$;
- Y is a variable ranging over $\mathcal{P}(Space)$, or a constant $r \subseteq Space$
- Z is a variable ranging over T, or a constant $t \in T$.

Definition (PST atom – SPOT atom in the previous framework)

A PST *atom* is a ground st-atom $loc(id, r, t)$ annotated with a probability interval $[\ell, u] \subseteq [0, 1]$ – denoted as $loc(id, r, t)[\ell, u]$.

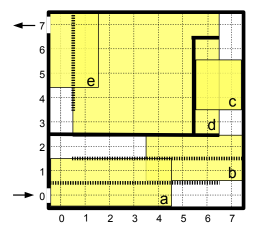
- $loc(id, r, t)[\ell, u]$ says that object id is/was/will be inside region r at time t with probability in the interval $[\ell, u]$

Example

- A set of PST atoms (i.e., a SPOT database)

Example

$loc(id_1, c, 9)[.9, 1]$
$loc(id_1, a, 1)[.4, .7]$
$loc(id_1, b, 1)[.4, .9]$
$loc(id_1, d, 15)[.6, 1]$
$loc(id_1, e, 18)[.7, 1]$
$loc(id_2, b, 2)[.5, .9]$
$loc(id_2, c, 12)[.9, 1]$
$loc(id_2, d, 18)[.6, .9]$
$loc(id_2, d, 20)[.2, .9]$



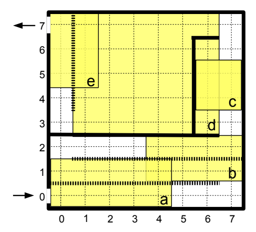
- To get PST KBs we add integrity constraints in the form of *spatio-temporal denial* formulas (*std* formulas for short)
- Such formulas are expressive enough to capture a large set of constraints

Example

- A set of PST atoms (i.e., a SPOT database)

Example

$loc(id_1, c, 9)[.9, 1]$
$loc(id_1, a, 1)[.4, .7]$
$loc(id_1, b, 1)[.4, .9]$
$loc(id_1, d, 15)[.6, 1]$
$loc(id_1, e, 18)[.7, 1]$
$loc(id_2, b, 2)[.5, .9]$
$loc(id_2, c, 12)[.9, 1]$
$loc(id_2, d, 18)[.6, .9]$
$loc(id_2, d, 20)[.2, .9]$



- To get PST KBs we add integrity constraints in the form of *spatio-temporal denial* formulas (*std* formulas for short)
- Such formulas are expressive enough to capture a large set of constraints

Spatio-temporal denial formula

Definition (Std- formula)

$$\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \neg[(\bigwedge_{i=1}^{k} loc(X_i, Y_i, Z_i)) \wedge \alpha(\mathbf{X}) \wedge \beta(\mathbf{Y}) \wedge \gamma(\mathbf{Z})]$$

- **X**, **Y**, and **Z** are sets whose variables range over *ID*, $\mathcal{P}(Space)$, and T
- $loc(X_i, Y_i, Z_i)$ are st-atoms such that X_i (resp., Y_i, Z_i) occurs in **X** (resp, **Y**, **Z**) — each variable in **X**, **Y**, and **Z** occurs in at least one st-atom
- $\alpha(\mathbf{X})$ is a conjunction of built-in predicates of the form $X_i \diamond X_j$, where X_i and X_j are variables in **X** or ids in *ID*, and $\diamond \in \{=, \neq\}$
- $\beta(\mathbf{Y})$ is a conjunction of built-in predicates $Y_i \diamond Y_j$, where Y_i and Y_j are variables in **Y** or regions, and $\diamond \in \{=, \neq, \subseteq, \supset, ov, nov\}$ (*ov* stands for "overlaps" and *nov* stands for "does not overlap")
- $\gamma(\mathbf{Z})$ is a conjunction of built-in predicates of the form $Z_i \diamond Z_j$, where each Z_i and Z_j is a time point in T or a variable in **Z** that may be followed by $+n$ where n is a positive integer, and $\diamond \in \{=, \neq, <, \geq\}$.

Spatio-temporal denial formula

Definition (Std- formula)

$$\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \neg[(\bigwedge_{i=1}^{k} loc(X_i, Y_i, Z_i)) \wedge \alpha(\mathbf{X}) \wedge \beta(\mathbf{Y}) \wedge \gamma(\mathbf{Z})]$$

- **X**, **Y**, and **Z** are sets whose variables range over *ID*, $\mathcal{P}(Space)$, and *T*
- $loc(X_i, Y_i, Z_i)$ are st-atoms such that X_i (resp., Y_i, Z_i) occurs in **X** (resp, **Y**, **Z**) — each variable in **X**, **Y**, and **Z** occurs in at least one st-atom
- $\alpha(\mathbf{X})$ is a conjunction of built-in predicates of the form $X_i \diamond X_j$, where X_i and X_j are variables in **X** or ids in *ID*, and $\diamond \in \{=, \neq\}$
- $\beta(\mathbf{Y})$ is a conjunction of built-in predicates $Y_i \diamond Y_j$, where Y_i and Y_j are variables in **Y** or regions, and $\diamond \in \{=, \neq, \subseteq, \supset, ov, nov\}$ (*ov* stands for "overlaps" and *nov* stands for "does not overlap")
- $\gamma(\mathbf{Z})$ is a conjunction of built-in predicates of the form $Z_i \diamond Z_j$, where each Z_i and Z_j is a time point in *T* or a variable in **Z** that may be followed by $+n$ where *n* is a positive integer, and $\diamond \in \{=, \neq, <, \geq\}$.

Spatio-temporal denial formula

Definition (Std- formula)

$$\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \neg[(\bigwedge_{i=1}^{k} loc(X_i, Y_i, Z_i)) \wedge \alpha(\mathbf{X}) \wedge \beta(\mathbf{Y}) \wedge \gamma(\mathbf{Z})]$$

- **X**, **Y**, and **Z** are sets whose variables range over *ID*, $\mathcal{P}(Space)$, and T
- $loc(X_i, Y_i, Z_i)$ are st-atoms such that X_i (resp., Y_i, Z_i) occurs in **X** (resp, **Y**, **Z**) — each variable in **X**, **Y**, and **Z** occurs in at least one st-atom
- $\alpha(\mathbf{X})$ is a conjunction of built-in predicates of the form $X_i \diamond X_j$, where X_i and X_j are variables in **X** or ids in *ID*, and $\diamond \in \{=, \neq\}$
- $\beta(\mathbf{Y})$ is a conjunction of built-in predicates $Y_i \diamond Y_j$, where Y_i and Y_j are variables in **Y** or regions, and $\diamond \in \{=, \neq, \subseteq, \supset, ov, nov\}$ (*ov* stands for "overlaps" and *nov* stands for "does not overlap")
- $\gamma(\mathbf{Z})$ is a conjunction of built-in predicates of the form $Z_i \diamond Z_j$, where each Z_i and Z_j is a time point in T or a variable in **Z** that may be followed by $+n$ where n is a positive integer, and $\diamond \in \{=, \neq, <, \geq\}$.

Spatio-temporal denial formula

Definition (Std- formula)

$$\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \neg [(\bigwedge_{i=1}^{k} loc(X_i, Y_i, Z_i)) \wedge \alpha(\mathbf{X}) \wedge \beta(\mathbf{Y}) \wedge \gamma(\mathbf{Z})]$$

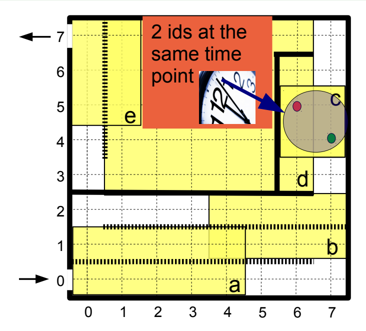
- **X**, **Y**, and **Z** are sets whose variables range over *ID*, $\mathcal{P}(Space)$, and T
- $loc(X_i, Y_i, Z_i)$ are st-atoms such that X_i (resp., Y_i, Z_i) occurs in **X** (resp, **Y**, **Z**) — each variable in **X**, **Y**, and **Z** occurs in at least one st-atom
- $\alpha(\mathbf{X})$ is a conjunction of built-in predicates of the form $X_i \diamond X_j$, where X_i and X_j are variables in **X** or ids in *ID*, and $\diamond \in \{=, \neq\}$
- $\beta(\mathbf{Y})$ is a conjunction of built-in predicates $Y_i \diamond Y_j$, where Y_i and Y_j are variables in **Y** or regions, and $\diamond \in \{=, \neq, \subseteq, \supset, ov, nov\}$ (*ov* stands for "overlaps" and *nov* stands for "does not overlap")
- $\gamma(\mathbf{Z})$ is a conjunction of built-in predicates of the form $Z_i \diamond Z_j$, where each Z_i and Z_j is a time point in T or a variable in **Z** that may be followed by $+n$ where n is a positive integer, and $\diamond \in \{=, \neq, <, \geq\}$.

Examples of spatio-temporal denial formulas

Example

1) *There cannot be two distinct objects in region c at any time point between* 1 *and* 20:

$$f_1 = \forall X_1, X_2, Z_1\ \neg[loc(X_1, c, Z_1) \wedge loc(X_2, c, Z_1) \wedge X_1 \neq X_2 \wedge Z_1 \geq 1 \wedge 20 \geq Z_1]$$

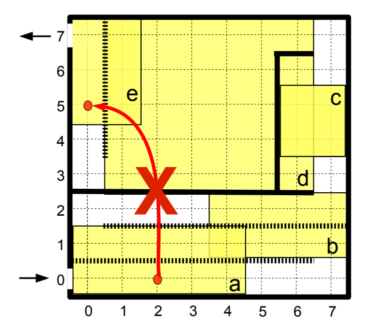


Examples of spatio-temporal denial formulas

Example

2) *No object can reach region e starting from region a in less than* 10 *time points*:

$$f_2 = \forall X_1, Z_1, Z_2\ \neg[loc(X_1, a, Z_1) \wedge loc(X_1, e, Z_2) \wedge Z_1 < Z_2 \wedge Z_2 < Z_1 + 10]$$

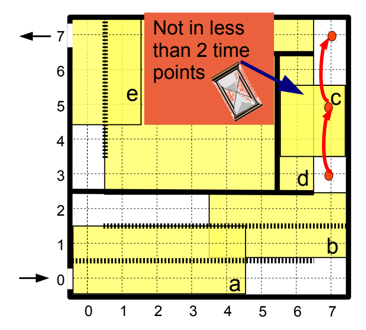


Examples of spatio-temporal denial formulas

Example

3) *Object id can go away from region c only if it stayed there for at least 2 time points*:

$f_3 = \forall Y_1, Y_2, Z_1, Z_2, Z_3 \; \neg[loc(id, Y_1, Z_1) \wedge loc(id, c, Z_2) \wedge loc(id, Y_2, Z_3) \wedge Y_1 \, nov \; c \wedge Y_2 \, nov \; c \wedge Z_2 = Z_1 + 1 \wedge Z_2 < Z_3 \wedge Z_2 + 2 \geq Z_3]$



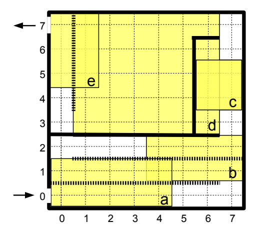
PST knowledgebases

Definition (PST KB)

A PST KB $\mathcal{K}$ is a pair $\langle \mathcal{A}, \mathcal{F} \rangle$, where $\mathcal{A}$ is a finite set of PST atoms and $\mathcal{F}$ is finite set of std-formulas.

Example

$$\begin{aligned}\mathcal{A} = \{ & loc(id_1, c, 9)[.9, 1] \\ & loc(id_1, a, 1)[.4, .7] \\ & loc(id_1, b, 1)[.4, .9] \\ & loc(id_1, d, 15)[.6, 1] \\ & loc(id_1, e, 18)[.7, 1] \\ & loc(id_2, b, 2)[.5, .9] \\ & loc(id_2, c, 12)[.9, 1] \\ & loc(id_2, d, 18)[.6, .9] \\ & loc(id_2, d, 20)[.2, .9]\}\end{aligned}$$



$$\begin{aligned}\mathcal{F} = \{ & f_1 = \forall X_1, X_2, Z_1\ \neg[loc(X_1, c, Z_1) \wedge loc(X_2, c, Z_1) \wedge X_1 \neq X_2 \wedge Z_1 \geq 1 \wedge 20 \geq Z_1] \\ & f_2 = \forall X_1, Z_1, Z_2\ \neg[loc(X_1, a, Z_1) \wedge loc(X_1, e, Z_2) \wedge Z_1 < Z_2 \wedge Z_2 < Z_1 + 10] \\ & f_3 = \forall Y_1, Y_2, Z_1, Z_2, Z_3\ \neg[loc(id, Y_1, Z_1) \wedge loc(id, c, Z_2) \wedge loc(id, Y_2, Z_3) \wedge \\ & \qquad Y_1\, nov\ c \wedge Y_2\, nov\ c \wedge Z_2 = Z_1 + 1 \wedge Z_2 < Z_3 \wedge Z_2 + 2 \geq Z_3]\}\end{aligned}$$

World

- A world specifies a possible trajectory for each object $id \in ID$ (i.e., says where in *Space* object *id* was/is/will be at each time $t \in T$)

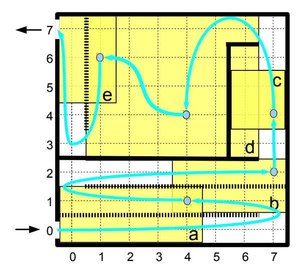
Definition (World)

A world w is a function, $w : ID \times T \rightarrow Space$

Example

World w_1 describes possible trajectories for id_1 and id_2 during the time interval $[0, 20]$:

$$w_1(id_1, t) = (4, 1) \text{ for } t \in [0, 5]$$
$$w_1(id_1, t) = (7, 2) \text{ for } t \in [6, 7]$$
$$w_1(id_1, t) = (7, 4) \text{ for } t \in [8, 10]$$
$$w_1(id_1, t) = (4, 4) \text{ for } t \in [11, 16]$$
$$w_1(id_1, t) = (1, 6) \text{ for } t \in [17, 20]$$



World

- A world specifies a possible trajectory for each object $id \in ID$ (i.e., says where in *Space* object *id* was/is/will be at each time $t \in T$)

Definition (World)

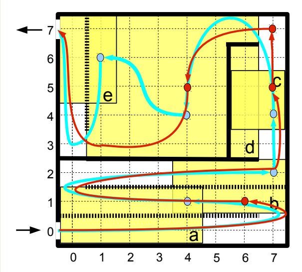
A world w is a function, $w : ID \times T \rightarrow Space$

Example

World w_1 describes possible trajectories for id_1 and id_2 during the time interval $[0, 20]$:

$w_1(id_1, t) = (4, 1)$ for $t \in [0, 5]$
$w_1(id_1, t) = (7, 2)$ for $t \in [6, 7]$
$w_1(id_1, t) = (7, 4)$ for $t \in [8, 10]$
$w_1(id_1, t) = (4, 4)$ for $t \in [11, 16]$
$w_1(id_1, t) = (1, 6)$ for $t \in [17, 20]$

$w_1(id_2, t) = (6, 1)$ for $t \in [0, 11]$
$w_1(id_2, t) = (7, 5)$ for $t \in [12, 15]$
$w_1(id_2, t) = (7, 7)$ for $t \in [16, 16]$
$w_1(id_2, t) = (4, 5)$ for $t \in [17, 20]$

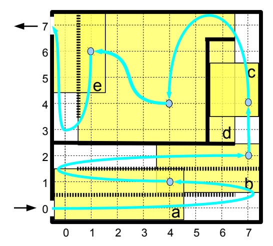


Satisfaction

- World w *satisfies* a ground st-atom $a = loc(id, r, t)$ (denoted as $w \models a$) iff $w(id, t) \in r$
- w satisfies a conjunction of ground st-atoms (i.e., a *ground* std-formula) iff w *satisfies* every st-atom in the conjunction

Example

$w_1(id_1, t) = (4, 1)$ for $t \in [0, 5]$
$w_1(id_1, t) = (7, 2)$ for $t \in [6, 7]$
$w_1(id_1, t) = (7, 4)$ for $t \in [8, 10]$
$w_1(id_1, t) = (4, 4)$ for $t \in [11, 16]$
$w_1(id_1, t) = (1, 6)$ for $t \in [17, 20]$



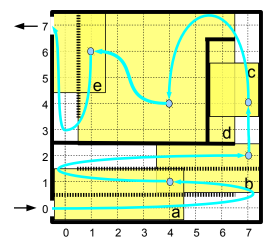
- $w_1 \models loc(id_1, b, 0)$, as $w_1(id_1, 0) = (4, 1)$ belongs to region b
- $w_1 \models \neg[loc(id_1, b, 0) \wedge loc(id_1, e, 15)]$ as $w_1 \not\models loc(id_1, e, 15)$, since $w_1(id_1, 15) = (4, 4) \notin e$

Satisfaction

- World w *satisfies* a ground st-atom $a = loc(id, r, t)$ (denoted as $w \models a$) iff $w(id, t) \in r$
- w satisfies a conjunction of ground st-atoms (i.e., a *ground* std-formula) iff w *satisfies* every st-atom in the conjunction

Example

$w_1(id_1, t) = (4, 1)$ for $t \in [0, 5]$
$w_1(id_1, t) = (7, 2)$ for $t \in [6, 7]$
$w_1(id_1, t) = (7, 4)$ for $t \in [8, 10]$
$w_1(id_1, t) = (4, 4)$ for $t \in [11, 16]$
$w_1(id_1, t) = (1, 6)$ for $t \in [17, 20]$



- $w_1 \models loc(id_1, b, 0)$, as $w_1(id_1, 0) = (4, 1)$ belongs to region b
- $w_1 \models \neg[loc(id_1, b, 0) \wedge loc(id_1, e, 15)]$ as $w_1 \not\models loc(id_1, e, 15)$, since $w_1(id_1, 15) = (4, 4) \notin e$

Ground std-formulas

- Given an std-formula f, Θ_f denotes the set of all substitutions of variables in $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ with constants in ID, $\mathcal{S}$, and T, respectively, where $\mathcal{S}$ is the set of all sets of *Space* that contain a single point
- The *ground* std-formula $\theta(f)$ resulting from applying $\theta \in \Theta_f$ to f is:

$$\theta(f) = \neg[(\bigwedge_{i=1}^{k} loc(\theta(X_i), \theta(Y_i), \theta(Z_i))) \wedge \alpha(\theta(\mathbf{X})) \wedge \beta(\theta(\mathbf{Y})) \wedge \gamma(\theta(\mathbf{Z}))]$$

Example

- $f_1 = \forall X_1, X_2, Z_1\ \neg[loc(X_1, c, Z_1) \wedge loc(X_2, c, Z_1) \wedge X_1 \neq X_2 \wedge Z_1 \geq 1 \wedge 20 \geq Z_1]$
- $\theta = \{X_1/id_1, X_2/id_2, Z_1/6\}$, where $id_1, id_2 \in ID$ and time point 6 is in T
- $\theta(f_1) = \neg[loc(id_1, c, 6) \wedge loc(id_2, c, 6)]$
  ($id_1 \neq id_2 \wedge 6 \geq 1 \wedge 6 \leq 20$, evaluating to *true*, is not reported in $\theta(f_1)$)

- World w *satisfies* an std-formula f (denoted as $w \models f$) iff for each substitution $\theta \in \Theta_f$, $w \models \theta(f)$

Ground std-formulas

- Given an std-formula f, Θ_f denotes the set of all substitutions of variables in $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ with constants in ID, $\mathcal{S}$, and T, respectively, where $\mathcal{S}$ is the set of all sets of *Space* that contain a single point
- The *ground* std-formula $\theta(f)$ resulting from applying $\theta \in \Theta_f$ to f is:

$$\theta(f) = \neg\left[\left(\bigwedge_{i=1}^{k} loc(\theta(X_i), \theta(Y_i), \theta(Z_i))\right) \wedge \alpha(\theta(\mathbf{X})) \wedge \beta(\theta(\mathbf{Y})) \wedge \gamma(\theta(\mathbf{Z}))\right]$$

Example

- $f_1 = \forall X_1, X_2, Z_1 \neg[loc(X_1, c, Z_1) \wedge loc(X_2, c, Z_1) \wedge X_1 \neq X_2 \wedge Z_1 \geq 1 \wedge 20 \geq Z_1]$
- $\theta = \{X_1/id_1, X_2/id_2, Z_1/6\}$, where $id_1, id_2 \in ID$ and time point 6 is in T
- $\theta(f_1) = \neg[loc(id_1, c, 6) \wedge loc(id_2, c, 6)]$
  ($id_1 \neq id_2 \wedge 6 \geq 1 \wedge 6 \leq 20$, evaluating to *true*, is not reported in $\theta(f_1)$)

- World w *satisfies* an std-formula f (denoted as $w \models f$) iff for each substitution $\theta \in \Theta_f$, $w \models \theta(f)$

Ground std-formulas

- Given an std-formula f, Θ_f denotes the set of all substitutions of variables in $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ with constants in ID, $\mathcal{S}$, and T, respectively, where $\mathcal{S}$ is the set of all sets of *Space* that contain a single point
- The *ground* std-formula $\theta(f)$ resulting from applying $\theta \in \Theta_f$ to f is:

$$\theta(f) = \neg\big[\big(\bigwedge_{i=1}^{k} loc(\theta(X_i), \theta(Y_i), \theta(Z_i))\big) \wedge \alpha(\theta(\mathbf{X})) \wedge \beta(\theta(\mathbf{Y})) \wedge \gamma(\theta(\mathbf{Z}))\big]$$

Example

- $f_1 = \forall X_1, X_2, Z_1 \; \neg[loc(X_1, c, Z_1) \wedge loc(X_2, c, Z_1) \wedge X_1 \neq X_2 \wedge Z_1 \geq 1 \wedge 20 \geq Z_1]$
- $\theta = \{X_1/id_1, X_2/id_2, Z_1/6\}$, where $id_1, id_2 \in ID$ and time point 6 is in T
- $\theta(f_1) = \neg[loc(id_1, c, 6) \wedge loc(id_2, c, 6)]$
  ($id_1 \neq id_2 \wedge 6 \geq 1 \wedge 6 \leq 20$, evaluating to *true*, is not reported in $\theta(f_1)$)

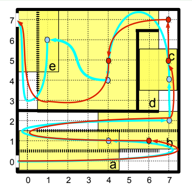
- World w *satisfies* an std-formula f (denoted as $w \models f$) iff for each substitution $\theta \in \Theta_f$, $w \models \theta(f)$

Interpretations

- An interpretation I for a PST KB $\mathcal{K}$ is a probability distribution function (PDF) over the set $\mathcal{W}(\mathcal{K})$ of all worlds of $\mathcal{K}$
- $I(w)$ is the probability that w describes the actual trajectories of all the objects

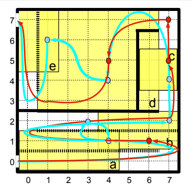
Example (Interpretation I)

w_1



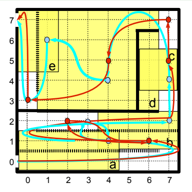
$I(w_1) = 0.7$

w_2



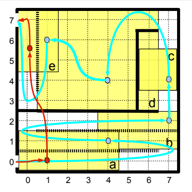
$I(w_2) = 0.2$

w_3



$I(w_3) = 0.1$

w_4



$I(w_4) = 0$

All other words are assigned probability equal to zero by interpretation I

- Only the interpretations that are compatible with the information in $\mathcal{K}$ (PST atoms + std-formulas) are models

Models

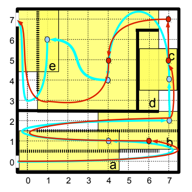
Definition (Model)

A model M for a PST KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ is an interpretation for $\mathcal{K}$ such that:

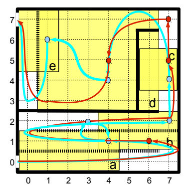
(i) $\forall\ loc(id, r, t)[\ell, u] \in \mathcal{A}, \quad \sum_{w \mid w \models loc(id,r,t)} M(w) \in [\ell, u];$

(ii) $\forall\ f \in \mathcal{F}, \quad \sum_{w \mid w \not\models f} M(w) = 0.$

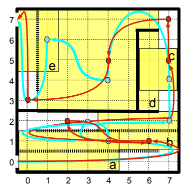
Example (Model M)



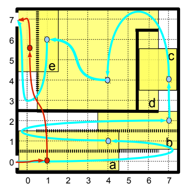
$M(w_1) = 0.7$



$M(w_2) = 0.2$



$M(w_3) = 0.1$



$M(w_4) = 0$

- For atom $loc(id_1, c, 9)[.9, 1]$,
  $\sum_{w \mid w \models loc(id_1,c,9)} M(w) = M(w_1) + M(w_2) + M(w_3) = 1 \in [.9, .1]$

Models

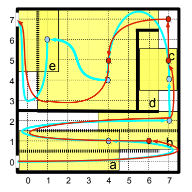
Definition (Model)

A model M for a PST KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ is an interpretation for $\mathcal{K}$ such that:

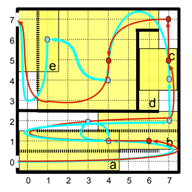
(i) $\forall\ loc(id, r, t)[\ell, u] \in \mathcal{A}, \quad \sum_{w \mid w \models loc(id,r,t)} M(w) \in [\ell, u];$

(ii) $\forall\ f \in \mathcal{F}, \quad \sum_{w \mid w \not\models f} M(w) = 0.$

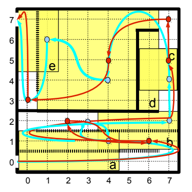
Example (Model M)



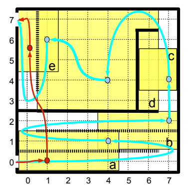
$M(w_1) = 0.7$



$M(w_2) = 0.2$



$M(w_3) = 0.1$



$M(w_4) = 0$

- $M(w_4) = 0$ since w_4 violates the constraint "no object can reach region e starting from region a in less than 10 time points"

Consistency

Definition (Model)

A model M for a PST KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ is an interpretation for $\mathcal{K}$ such that:

(i) $\forall\ loc(id, r, t)[\ell, u] \in \mathcal{A}, \quad \sum_{w \mid w \models loc(id,r,t)} M(w) \in [\ell, u];$

(ii) $\forall\ f \in \mathcal{F}, \quad \sum_{w \mid w \not\models f} M(w) = 0.$

Definition (Consistency)

PST KB $\mathcal{K}$ is consistent iff there is a model for it.

Outline

Complexity

Theorem

Deciding whether PST KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ is consistent is NP-complete.

- Membership: deciding whether $\mathcal{K}$ is consistent corresponds to checking the feasibility of

$$LP(\mathcal{K}) := \begin{cases} (1) & \forall\ loc(id,r,t)[\ell,u] \in \mathcal{A}, \\ & \quad (a)\ \ \ell \leq \sum_{w_i | w_i \models loc(id,r,t)} v_i \\ & \quad (b)\ \ \sum_{w_i | w_i \models loc(id,r,t)} v_i \leq u \\ (2) & \forall f \in \mathcal{F},\ \sum_{w_i \mid w_i \not\models f} v_i = 0 \\ (3) & \sum_{w_i \mid w_i \in \mathcal{W}(\mathcal{K})} v_i = 1 \\ (4) & \forall w_i \in \mathcal{W}(\mathcal{K}),\ v_i \geq 0 \end{cases}$$

- v_i represents probability $M(w_i)$ assigned to $w_i \in \mathcal{W}(\mathcal{K})$ by $M \in \mathbf{M}(\mathcal{K})$
- Exponential number of variables v_i ($|\mathcal{W}(\mathcal{K})| = |Space|^{|ID|\cdot|T|}$)

Complexity

Theorem

Deciding whether PST *KB* $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ *is consistent is NP-complete.*

- Membership: deciding whether $\mathcal{K}$ is consistent corresponds to checking the feasibility of

$$LP(\mathcal{K}) := \begin{cases} (1) & \forall\, loc(id, r, t)[\ell, u] \in \mathcal{A}, \\ & \quad (a)\;\; \ell \leq \sum_{w_i | w_i \models loc(id,r,t)} v_i \\ & \quad (b)\;\; \sum_{w_i | w_i \models loc(id,r,t)} v_i \leq u \\ (2) & \forall f \in \mathcal{F}, \sum_{w_i \,|\, w_i \not\models f} v_i = 0 \\ (3) & \sum_{w_i \,|\, w_i \in \mathcal{W}(\mathcal{K})} v_i = 1 \\ (4) & \forall w_i \in \mathcal{W}(\mathcal{K}),\;\; v_i \geq 0 \end{cases}$$

- v_i represents probability $M(w_i)$ assigned to $w_i \in \mathcal{W}(\mathcal{K})$ by $M \in \mathbf{M}(\mathcal{K})$
- Exponential number of variables v_i ($|\mathcal{W}(\mathcal{K})| = |Space|^{|ID| \cdot |T|}$)

Complexity

Theorem

Deciding whether PST KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ is consistent is NP-complete.

- Membership: deciding whether $\mathcal{K}$ is consistent corresponds to checking the feasibility of

$$LP(\mathcal{K}) := \begin{cases} (1) & \forall\, loc(id, r, t)[\ell, u] \in \mathcal{A}, \\ & \quad (a)\ \ \ell \leq \sum_{w_i \mid w_i \models loc(id,r,t)} v_i \\ & \quad (b)\ \ \sum_{w_i \mid w_i \models loc(id,r,t)} v_i \leq u \\ (2) & \forall f \in \mathcal{F},\ \sum_{w_i \mid w_i \not\models f} v_i = 0 \\ (3) & \sum_{w_i \mid w_i \in \mathcal{W}(\mathcal{K})} v_i = 1 \\ (4) & \forall w_i \in \mathcal{W}(\mathcal{K}),\ v_i \geq 0 \end{cases}$$

- v_i represents probability $M(w_i)$ assigned to $w_i \in \mathcal{W}(\mathcal{K})$ by $M \in \mathbf{M}(\mathcal{K})$
- Exponential number of variables v_i ($|\mathcal{W}(\mathcal{K})| = |Space|^{|ID| \cdot |T|}$)

Membership in NP

- It can be shown that $LP(\mathcal{K})$ is feasible iff there is a solution for $LP(\mathcal{K})$ consisting of at most $2 \cdot |\mathcal{A}| + |\mathcal{F}| + 1$ non-zero variables (it follows from a well-known result on the size of solutions of linear programming problems [Papadimitriou, Steiglitz '82])
- Guess an assignment s' consisting of $2 \cdot |\mathcal{A}| + |\mathcal{F}| + 1$ pairs $\langle v_i, \text{value of } v_i\rangle$,
- Check in polynomial time whether s' is a solution of $LP^*(\mathcal{K})$, obtained from $LP(\mathcal{K})$ by keeping in it only the variables in s'
- If s' is a solution of $LP^*(\mathcal{K})$, then $LP(\mathcal{K})$ is feasible

Membership in NP

- It can be shown that $LP(\mathcal{K})$ is feasible iff there is a solution for $LP(\mathcal{K})$ consisting of at most $2 \cdot |\mathcal{A}| + |\mathcal{F}| + 1$ non-zero variables (it follows from a well-known result on the size of solutions of linear programming problems [Papadimitriou, Steiglitz '82])
- Guess an assignment s' consisting of $2 \cdot |\mathcal{A}| + |\mathcal{F}| + 1$ pairs $\langle v_i, \text{value of } v_i \rangle$,
- Check in polynomial time whether s' is a solution of $LP^*(\mathcal{K})$, obtained from $LP(\mathcal{K})$ by keeping in it only the variables in s'
- If s' is a solution of $LP^*(\mathcal{K})$, then $LP(\mathcal{K})$ is feasible

NP-hardness

- Reduction from Hamiltonian path problem
- Consider a graph $G = \langle V, E \rangle$ with vertexes $V = \{v_0, \ldots, v_k\}$ and edges E
- Assume $ID = \{id\}$, $Space = V$, and $T = [0, \ldots, k]$
- Define $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ such that
  - $\mathcal{A}$ consists of the PST atom $loc(id, v_0, 0)[1, 1]$, and
  - $\mathcal{F}$ consists of std-formulas f_1^i (with $i \in [0..k]$) and f_2 such that:

    i) $f_1^i = \forall Z_1, Z_2 \neg[loc(id, \{v_i\}, Z_1) \wedge loc(id, Space \setminus V', Z_2) \wedge Z_2 = Z_1 + 1]$ where V' is the set of vertexes v_j s.t. $(v_i, v_j) \in E$

    ii) $f_2 = \forall Y_1, Z_1, Z_2 \neg[loc(id, Y_1, Z_1) \wedge loc(id, Y_1, Z_2) \wedge Z_1 \neq Z_2]$
- There is a model for $\mathcal{K}$ iff there is a Hamiltonian path in G

$\Rightarrow$ Every world which is assigned a probability greater than zero by a model encodes a Hamiltonian path

$\Leftarrow$ Given a Hamiltonian path, define a model as a PDF over worlds

NP-hardness

- Reduction from Hamiltonian path problem
- Consider a graph $G = \langle V, E \rangle$ with vertexes $V = \{v_0, \ldots, v_k\}$ and edges E
- Assume $ID = \{id\}$, $Space = V$, and $T = [0, \ldots, k]$
- Define $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ such that

- $\mathcal{A}$ consists of the PST atom $loc(id, v_0, 0)[1, 1]$, and
- $\mathcal{F}$ consists of std-formulas f_1^i (with $i \in [0..k]$) and f_2 such that:

  i) $f_1^i = \forall Z_1, Z_2\ \neg[loc(id, \{v_i\}, Z_1) \wedge loc(id, Space \setminus V', Z_2) \wedge Z_2 = Z_1 + 1]$ where V' is the set of vertexes v_j s.t. $(v_i, v_j) \in E$

  ii) $f_2 = \forall Y_1, Z_1, Z_2\ \neg[loc(id, Y_1, Z_1) \wedge loc(id, Y_1, Z_2) \wedge Z_1 \neq Z_2]$

- There is a model for $\mathcal{K}$ iff there is a Hamiltonian path in G

$\Rightarrow$ Every world which is assigned a probability greater than zero by a model encodes a Hamiltonian path

$\Leftarrow$ Given a Hamiltonian path, define a model as a PDF over worlds

NP-hardness

- Reduction from Hamiltonian path problem
- Consider a graph $G = \langle V, E \rangle$ with vertexes $V = \{v_0, \ldots, v_k\}$ and edges E
- Assume $ID = \{id\}$, $Space = V$, and $T = [0, \ldots, k]$
- Define $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ such that
  - $\mathcal{A}$ consists of the PST atom $loc(id, v_0, 0)[1, 1]$, and
  - $\mathcal{F}$ consists of std-formulas f_1^i (with $i \in [0..k]$) and f_2 such that:

    i) $f_1^i = \forall Z_1, Z_2\ \neg[loc(id, \{v_i\}, Z_1) \wedge loc(id, Space \setminus V', Z_2) \wedge Z_2 = Z_1 + 1]$ where V' is the set of vertexes v_j s.t. $(v_i, v_j) \in E$

    ii) $f_2 = \forall Y_1, Z_1, Z_2\ \neg[loc(id, Y_1, Z_1) \wedge loc(id, Y_1, Z_2) \wedge Z_1 \neq Z_2]$
- There is a model for $\mathcal{K}$ iff there is a Hamiltonian path in G

$\Rightarrow$ Every world which is assigned a probability greater than zero by a model encodes a Hamiltonian path

$\Leftarrow$ Given a Hamiltonian path, define a model as a PDF over worlds

MBLP($\mathcal{K}$)

- A mixed-binary linear programming problem whose feasibility entails consistency
- Variable $v_{id,t,p}$ represents the probability that *id* is at point *p* at time *t*
- Binary variables δ

Definition (MBLP($\mathcal{K}$))

MBLP($\mathcal{K}$) consists of the following (in)equalities:

(1) $\forall loc(id, r, t)[\ell, u] \in \mathcal{A}$: $\ell \leq \sum_{p \in r} v_{id,t,p} \leq u$;

(2) $\forall id \in ID, t \in T$: $\sum_{p \in Space} v_{id,t,p} = 1$;

(3) $\forall p \in Space, id \in ID, t \in T$: $v_{id,t,p} \geq 0$;

(4) $\forall f \in \mathcal{F}, \forall \theta \in \Theta_f$ s.t. $\theta(f)$ is logically equivalent to the negation of the conjunction of st-atoms $\bigwedge_{i=1}^{k} loc(\theta(X_i), \theta(Y_i), \theta(Z_i))$, the inequalities:

 (a) $\forall i \in [1..k]$: $\sum_{p \in \theta(Y_i)} v_{\theta(X_i),\theta(Z_i),p} \leq \delta_i$;

 (b) $\sum_{i=1}^{k} \delta_i = k - 1$; // at least one st-atom is false

 (c) $\forall i \in [1..k]$: $\delta_i \in \{0, 1\}$.

MBLP($\mathcal{K}$)

- A mixed-binary linear programming problem whose feasibility entails consistency
- Variable $v_{id,t,p}$ represents the probability that *id* is at point *p* at time *t*
- Binary variables δ

Definition (MBLP($\mathcal{K}$))

$MBLP(\mathcal{K})$ consists of the following (in)equalities:

(1) $\forall loc(id, r, t)[\ell, u] \in \mathcal{A}$: $\ell \leq \sum_{p \in r} v_{id,t,p} \leq u$;

(2) $\forall id \in ID, t \in T$: $\sum_{p \in Space} v_{id,t,p} = 1$;

(3) $\forall p \in Space, id \in ID, t \in T$: $v_{id,t,p} \geq 0$;

(4) $\forall f \in \mathcal{F}, \forall \theta \in \Theta_f$ s.t. $\theta(f)$ is logically equivalent to the negation of the conjunction of st-atoms $\bigwedge_{i=1}^{k} loc(\theta(X_i), \theta(Y_i), \theta(Z_i))$, the inequalities:

(a) $\forall i \in [1..k]$: $\sum_{p \in \theta(Y_i)} v_{\theta(X_i),\theta(Z_i),p} \leq \delta_i$;

(b) $\sum_{i=1}^{k} \delta_i = k - 1$; // at least one st-atom is false

(c) $\forall i \in [1..k]$: $\delta_i \in \{0, 1\}$.

Using MBLP($\mathcal{K}$)

Theorem

If MBLP($\mathcal{K}$) is feasible then $\mathcal{K}$ is consistent

- Techniques for solving linear optimization problems can be adopted to address the consistency checking problem
- The converse of the theorem above does not hold

Example

Let $ID = \{id\}$, $T = [0, 1]$, $Space = \{p_0, p_1\}$, $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ where
$\mathcal{A} = \{loc(id, p_0, 0)[0.5, 0.5], loc(id, p_1, 1)[0.5, 0.5]\}$ and
$\mathcal{F} = \{\neg[loc(id, \{p_0\}, 0) \wedge loc(id, \{p_1\}, 1)]\}$

w_i	w_1	w_2	w_3	w_4
$w_i(id, 0)$	p_0	p_0	p_1	p_1
$w_i(id, 1)$	p_0	p_1	p_0	p_1
$M(w_i)$	0.5	0	0	0.5

- M is a model for $\mathcal{K}$

Using MBLP($\mathcal{K}$)

Theorem

If MBLP($\mathcal{K}$) is feasible then $\mathcal{K}$ is consistent

- Techniques for solving linear optimization problems can be adopted to address the consistency checking problem
- The converse of the theorem above does not hold

Example

Let $ID = \{id\}$, $T = [0, 1]$, $Space = \{p_0, p_1\}$, $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ where
$\mathcal{A} = \{loc(id, p_0, 0)[0.5, 0.5], loc(id, p_1, 1)[0.5, 0.5]\}$ and
$\mathcal{F} = \{\neg[loc(id, \{p_0\}, 0) \wedge loc(id, \{p_1\}, 1)]\}$

w_i	w_1	w_2	w_3	w_4
$w_i(id, 0)$	p_0	p_0	p_1	p_1
$w_i(id, 1)$	p_0	p_1	p_0	p_1
$M(w_i)$	0.5	0	0	0.5

$0.5 \leq v_{id,0,p_0} \leq 0.5$; $\quad 0.5 \leq v_{id,1,p_1} \leq 0.5$;
$v_{id,0,p_0} + v_{id,0,p_1} = 1$; $\quad v_{id,1,p_0} + v_{id,1,p_1} = 1$;
$v_{id,0,p_0} \leq \delta_1$; $\quad v_{id,1,p_1} \leq \delta_2$;
$\delta_1 + \delta_2 = 1$; $\quad \delta_1, \delta_2 \in \{0, 1\}$; $v_{id,i,p_j} \geq 0$

- *M* is a model for $\mathcal{K}$ but *MBLP*($\mathcal{K}$) is not feasible, as it includes the inequalities above

Unary std-formulas are tractable

- *Unary* std-formulas consist of only one st-atom and a conjunction of built-in predicates

Example

- "*There is no object in region r at any time between* 5 *and* 10" :
$$\forall X_1, Z_1 \ \neg[loc(X_1, r, Z_1) \wedge Z_1 \geq 5 \wedge 10 \geq Z_1]$$
- "*Object id is always in region r*":
$$\forall Y_1, Z_1 \ \neg[loc(id, Y_1, Z_1) \wedge Y_1 \, nov \ r].$$

Theorem

Let $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ *be a* PST *KB such that* $\mathcal{F}$ *consists of unary std-formulas only. Then, deciding whether* $\mathcal{K}$ *is consistent is in PTIME.*

Unary std-formulas are tractable

- *Unary* std-formulas consist of only one st-atom and a conjunction of built-in predicates

Example

- "*There is no object in region r at any time between* 5 *and* 10" :
$$\forall X_1, Z_1 \ \neg[loc(X_1, r, Z_1) \wedge Z_1 \geq 5 \wedge 10 \geq Z_1]$$
- "*Object id is always in region r*":
$$\forall Y_1, Z_1 \ \neg[loc(id, Y_1, Z_1) \wedge Y_1\, nov\ r].$$

Theorem

Let $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ *be a* PST *KB such that* $\mathcal{F}$ *consists of unary std-formulas only. Then, deciding whether* $\mathcal{K}$ *is consistent is in PTIME.*

Outline

Queries

- Query $(?id, q, ?t, [\ell, u])$ says: "Given a region q and a probability interval $[\ell, u]$, find all objects id and times t such that id is inside region q at time t with a probability in the interval $[\ell, u]$."
- Two semantics for interpreting this statement

Definition (Optimistic/Cautious Query Answers)

Let $\mathcal{K}$ be a consistent PST KB, and $Q = (?id, q, ?t, [\ell, u])$ a query. Then, $\langle id, t \rangle$ is

- an optimistic answer to Q w.r.t. $\mathcal{K}$ iff there is a model M for $\mathcal{K}$ s.t.

$$\sum_{w \mid w \models loc(id,q,t)} M(w) \in [\ell, u]$$

- a cautious answer to Q w.r.t. $\mathcal{K}$ iff for each a model M for $\mathcal{K}$ it holds that

$$\sum_{w \mid w \models loc(id,q,t)} M(w) \in [\ell, u]$$

Queries

- Query $(?id, q, ?t, [\ell, u])$ says: "Given a region q and a probability interval $[\ell, u]$, find all objects id and times t such that id is inside region q at time t with a probability in the interval $[\ell, u]$."
- Two semantics for interpreting this statement

Definition (Optimistic/Cautious Query Answers)

Let $\mathcal{K}$ be a consistent PST KB, and $Q = (?id, q, ?t, [\ell, u])$ a query.
Then, $\langle id, t\rangle$ is

- an optimistic answer to Q w.r.t. $\mathcal{K}$ iff there is a model M for $\mathcal{K}$ s.t.

$$\sum_{w \mid w \models loc(id,q,t)} M(w) \in [\ell, u]$$

- a cautious answer to Q w.r.t. $\mathcal{K}$ iff for each a model M for $\mathcal{K}$ it holds that

$$\sum_{w \mid w \models loc(id,q,t)} M(w) \in [\ell, u]$$

Answering Queries

- Consistency checking can be used to answer queries under both optimistic and cautious semantics

Proposition

Let $\mathcal{K} = \langle \mathcal{A}, \mathcal{F} \rangle$ *be a consistent* PST *KB, and* $Q = (?id, q, ?t, [\ell, u])$*. Then,*

- $\langle id, t \rangle$ *is an optimistic answer to* Q *w.r.t.* $\mathcal{K}$ *iff* $\langle \mathcal{A} \cup \{loc(id, q, t)[\ell, u]\}, \mathcal{F} \rangle$ *is consistent.*
- $\langle id, t \rangle$ *is a cautious answer to* Q *w.r.t.* $\mathcal{K}$ *iff* $\langle \mathcal{A} \cup \{loc(id, q, t)[0, \ell - \epsilon]\}, \mathcal{F} \rangle$ *and* $\langle \mathcal{A} \cup \{loc(id, q, t)[u + \epsilon, 1]\}, \mathcal{F} \rangle$ *are not consistent.*

- $\epsilon = 1/(ma)^m$ where $m = 2 \cdot |\mathcal{A}| + |F| + 1$ and a is the maximum among the numerators and denominators of the probabilities in $\mathcal{K}$
- The size of ϵ is polynomial w.r.t. the size of $\mathcal{K}$
- The value of ϵ can be determined by applying a well-known result on boundedness of solutions of linear programming problems [Papadimitriou, Steiglitz '82].

Outline

Conclusions and future work

- Knowledge representation for probabilistic spatio-temporal data
- The knowledge is represented as
  - spatio-temporal atoms describing the location of objects in time with a probability interval
  - spatio-temporal denial formulas describing the integrity constraints the system must satisfy
- We showed that
  - consistency checking is NP-complete
  - sufficient conditions for checking consistency via linear programming
  - a class of formulas for which consistency checking is PTIME
  - using consistency checking for answering queries under both optimistic and cautious semantics
- Further issues that we plan to investigate:
  - other tractable cases
  - complexity of query answering for consistent PST KBs
  - repairing inconsistent PST KBs and answering queries
  - process queries after updates

Conclusions and future work

- Knowledge representation for probabilistic spatio-temporal data
- The knowledge is represented as
  - spatio-temporal atoms describing the location of objects in time with a probability interval
  - spatio-temporal denial formulas describing the integrity constraints the system must satisfy
- We showed that
  - consistency checking is NP-complete
  - sufficient conditions for checking consistency via linear programming
  - a class of formulas for which consistency checking is PTIME
  - using consistency checking for answering queries under both optimistic and cautious semantics
- Further issues that we plan to investigate:
  - other tractable cases
  - complexity of query answering for consistent PST KBs
  - repairing inconsistent PST KBs and answering queries
  - process queries after updates

Conclusions and future work

- Knowledge representation for probabilistic spatio-temporal data
- The knowledge is represented as
  - spatio-temporal atoms describing the location of objects in time with a probability interval
  - spatio-temporal denial formulas describing the integrity constraints the system must satisfy
- We showed that
  - consistency checking is NP-complete
  - sufficient conditions for checking consistency via linear programming
  - a class of formulas for which consistency checking is PTIME
  - using consistency checking for answering queries under both optimistic and cautious semantics
- Further issues that we plan to investigate:
  - other tractable cases
  - complexity of query answering for consistent PST KBs
  - repairing inconsistent PST KBs and answering queries
  - process queries after updates

Thank you!

... any question?

Selected References

Parker, A., Subrahmanian, V.S., Grant, J.
A logical formulation of probabilistic spatial databases.
IEEE TKDE, pp. 1541–1556, 2007.

Papadimitriou, C.H., Steiglitz, K.,
Combinatorial optimization: algorithms and complexity.
Prentice-Hall, Inc., 1982.