Digital Sandbox

Exploiting inference to improve temporal RDF annotations and queries for machine reading

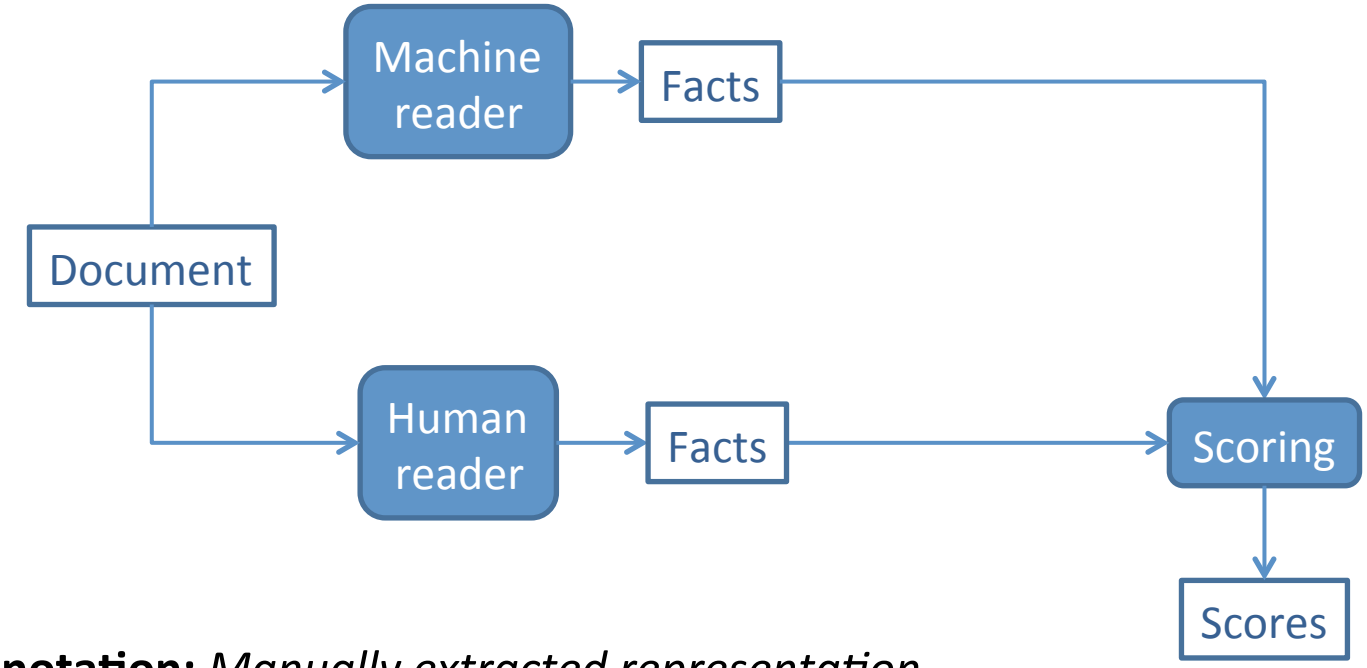
Robert C. Schrag, Ph.D.

STIDS 2012

"Machine reading" context

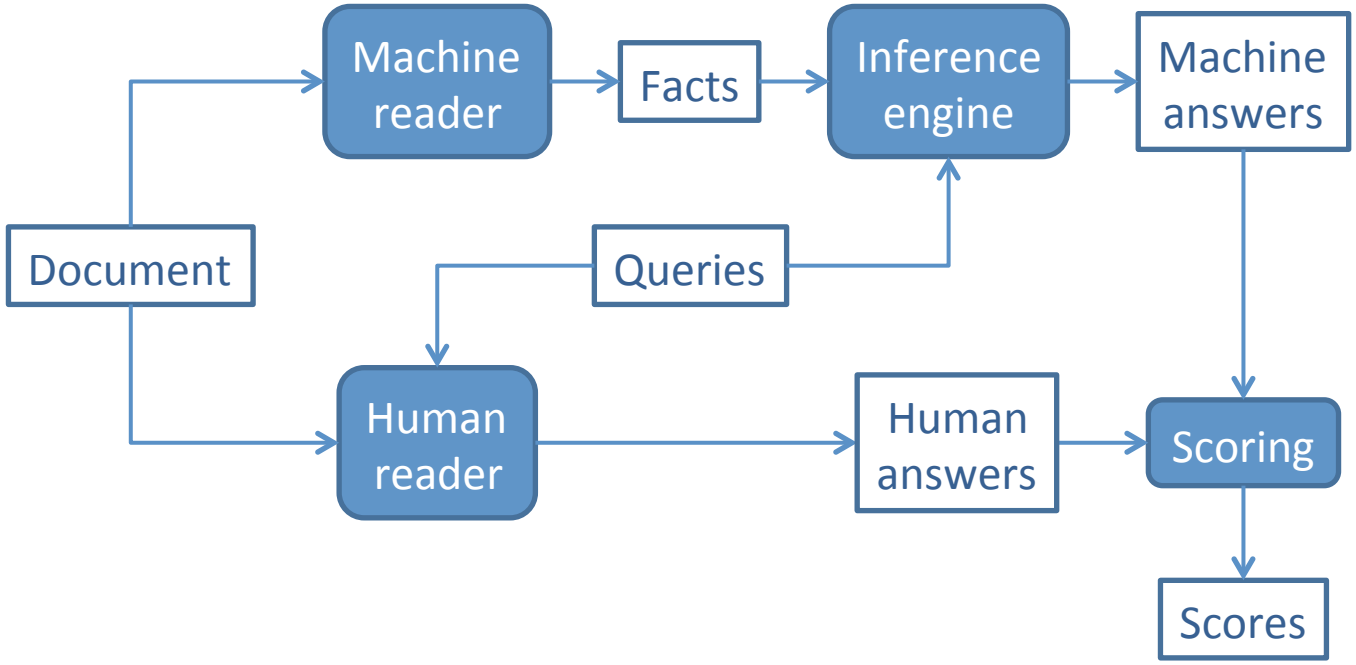
- **Working definition:** *Automatic extraction of formal knowledge from natural language text*
- **DARPA program**, 2009–2012:
  - Machine Reading > SAIC evaluation team > Global InfoTek KR&R team > Schrag
  - 2011 focus on **extracting temporal knowledge**
    - Extraction target formalism = variant of **event calculus**
    - **Query-based evaluation…**
    - Inference capabilities specified via **formal logic axioms**
    - We developed a **temporal reasoning engine** implementing the axioms.
- ***This* paper:** Using the engine to improve manually authored text annotations and queries
- ***Next* paper:** Revising axioms to correct anomalies associated with "naïve" time point representation

Annotation-based evaluation (traditional, "intrinsic")



Annotation: *Manually extracted representation used as a reference in evaluation (or training)*

Query-based evaluation (DARPA Machine Reading, "extrinsic")



Temporal reasoning engine

Temporal logic:

- Based (loosely) on event calculus (Kowalski, Shanahan)
- Fluent observation: **fluent *Alive* holds throughout time interval *Lifetime***
- Events, fluent initiation and termination: **event *Birth* initiates fluent *Alive***
- Limited temporal persistence and clipping: **fluent *Alive* persists until event *Death***

Reference implementation:

- Answers any conjunctive query
- Not yet heavily optimized
- Fast enough for evaluation purposes

How the engine supports language R&D

Done:

- Vet an evaluation process end to end, validating / refining:
  - Axioms (inference)
  - Ontologies (reading targets)
  - Formats (input, output)
  - Test query formalizations

Proposed:

- Provide on-line feedback during temporal text annotation.
- Reuse techniques from above in query authoring.

How the engine supports language R&D

Done:

- Vet an evaluation process end to end, validating / refining:
  - Axioms (inference)
  - Ontologies (reading targets)
  - Formats (input, output)
  - Test query formalizations *(1st)*

Proposed:

- Provide on-line feedback during temporal text annotation *(2nd)*.
- Reuse techniques from above in query authoring *(3rd)*.

Validating test query formalizations *(1st)*

Diagnose issues.

Formalize.

Given NL…	Document	Query	Answer(s)
Produce RDF…	Manually author selected statements to support expected inference.	Formalize query.	Execute query to get results.

Compare answers.

Apply temporal reasoning engine.

Methodology used to debug subtle errors in (manual) query formulations

Query English

Find all persons who were born in Ljubljana in the 1950s **and attended Ljubljana University in the 1980s,** *the titles that they held, the organizations in which they held these titles, and the maximal known time periods over which they* **attended** *and held these titles.*

Intended Source Document Excerpts

Lame duck Slovenian PM to take over EU presidency
Ljubljana, Dec 28, 2007 (AFP)

- *Slovenia's Janez Jansa will take over the EU presidency on January 1 as essentially a lame duck prime minister...*
- *Born on September 17, 1958 in Ljubljana,* **Jansa graduated from Ljubljana University in 1984...**
- *Defence minister from 1990 to 1994 and again in 2000...*
- *...he was elected prime minister on November 9, 2004...*

Expected formal knowledge

F_school: attendsSchool (Janez_Jansa Ljubljana_University)

holdsThroughout(F_school I_school)
clippedForward(F_school I_school)
hasTimeIntervalSpecString(I_school [,1984])
hasPersonBorn(birth Janez_Jansa)
occursAt(birth P_birth)
hasTimePointSpecString(P_birth 1958-09-17)
hasPersonBorn(birth Janez_Jansa)
hasBirthEventGPE-spec(birth GPEspec)
hasCityTownOrVillage(GPEspec ljubljana_Ljubljana_Slovenia)
hasNationState(GPEspec Slovenia)
type(Defence_Minister MinisterTitle)

BirthEvent(Janez_Jansa, Ljubljana)

F_PTIO_DM_1: personHasTitleInOrganization(Janez_Jansa Defence_Minister Slovenia)

holdsThroughout(F_PTIO_DM_1 I_PTIO_DM_1)
clippedBackward(F_PTIO_DM_1 I_PTIO_DM_1)
clippedForward(F_PTIO_DM_1 I_PTIO_DM_1)
hasTimeIntervalSpecString(I_PTIO_DM_1 [1990,1994])

F_PTIO_DM_2: personHasTitleInOrganization(Janez_Jansa Defence_Minister Slovenia)

holdsThroughout(F_PTIO_DM_2 I_PTIO_DM_2)
clippedBackward(F_PTIO_DM_2 I_PTIO_DM_2)
clippedforward(F_PTIO_DM_2 I_PTIO_DM_2)
hasTimeIntervalSpecString(I_PTIO_DM_2 [2000,2000])

F_PTIO_PM: personHasTitleInOrganization(Janez_Jansa Prime_Minister Slovenia)

holdsThroughout(F_PTIO_PM I_PTIO_PM)
clippedBackward(F_PTIO_PM I_PTIO_PM)
hasBeginningTimePoint(I_PTIO_PM I_PTIO_PM_beginning)
hasTimePointSpecString(Slovenia_2004_Election_Day 2004-11-09)
timePointGreaterThanOrEqualTo(I_PTIO_PM_beginning Slovenia_2004_Election_Day)
hasReportingAspect(I_PTIO_PM Ongoing)
ref(annotation I_PTIO_PM)
annotation(document annotation)
hasReportingChronusSpecString(document 2007-12-28)

Query Formalization 1

```
?F_school attendsSchool(?head Ljubljana_University)
          hasTimeIntervalSpecString(?I_range "[1980,1989]")
          holdsWithin(?F_school ?I_range)
          maximallyHoldsThroughout(?F_school ?I_school)
          (hasTimeIntervalSpecString ?I_school ?aiss)
  ?F_title personHasTitleInOrganization(?person ?title ?org)
          maximallyHoldsThroughout(?F_title ?I_title)
          hasTimeIntervalSpecString(?I_title ?tiss)
          hasPersonBorn(?birth ?person)
          hasBirthEventGPE-spec(?birth ?GPEspec)
          hasNationState(?GPEspec Slovenia)
          hasCityTownOrVillage(?GPEspec ljubljana_Ljubljana_Slovenia)
          hasTimeIntervalSpecString(?I_range "[1950,1959]")
          occursWithin(?birth ?I_range)
```

No answers!

Query Formalization 4

?F_school attendsSchool(?person Ljubljana_University)
hasTimeIntervalSpecString(?I_range_school "[1980-01-01,1989-12-31]")
holdsWithin(?F_school ?I_range_school)
maximallyHoldsThroughout(?F_school ?I_school)
hasTimeIntervalSpecString(?I_school ?aiss)
?F_title personHasTitleInOrganization(?person ?title ?org)
maximallyHoldsThroughout(?F_title ?I_title)
hasTimeIntervalSpecString(?I_title ?tiss)
hasPersonBorn(?birth ?person)
hasBirthEventGPE-spec(?birth ?GPEspec)
hasNationState(?GPEspec Slovenia)
hasCityTownOrVillage(?GPEspec ljubljana_Ljubljana_Slovenia)
hasTimeIntervalSpecString(?I_range_birth "[1950-01-01,1959-12-31]")
occursWithin(?birth ?I_range_birth)

Query Formalization 4

```
?F_school attendsSchool(?person Ljubljana_University)
          hasTimeIntervalSpecString(?I_range_school "[1980-01-01,1989-12-31]")
          holdsWithin(?F_school ?I_range_school)
          maximallyHoldsThroughout(?F_school ?I_school)
          hasTimeIntervalSpecString(?I_school ?aiss)
?F_title personHasTitleInOrganization(?person ?title ?org)
          maximallyHoldsThroughout(?F_title ?I_title)
          hasTimeIntervalSpecString(?I_title ?tiss)
          hasPersonBorn(?birth ?person)
          hasBirthEventGPE-spec(?birth ?GPEspec)
          hasNationState(?GPEspec Slovenia)
          hasCityTownOrVillage(?GPEspec ljubljana_Ljubljana_Slovenia)
          hasTimeIntervalSpecString(?I_range_birth "[1950-01-01,1959-12-31]")
          occursWithin(?birth ?I_range_birth)
```

Query Expected Answer (1 of 3)

F_school attendsSchool (Janez_Jansa Ljubljana_University)
hasTimeIntervalSpecString(TimeInterval-32 "[1980-01-01,1989-12-31]")
holdsWithin(F_school TimeInterval-32)
maximallyHoldsThroughout(F_school I_school)
hasTimeIntervalSpecString (I_school "[[,1984-12-31],[1984-01-01,1984-12-31]]:[0D,]")

F_PTIO_PM personHasTitleInOrganization(Janez_Jansa Prime_Minister Slovenia)
maximallyHoldsThroughout (F_PTIO_PM I_PTIO_PM)
hasTimeIntervalSpecString (I_PTIO_PM "[[2004-11-09,2007-12-28],[2007-12-28,]]:[0D,]")
hasPersonBorn(birth Janez_Jansa)
hasBirthEventGPE-spec(birth GPEspec)
hasNationState(GPEspec Slovenia)
hasCityTownOrVillage(GPEspec ljubljana_Ljubljana_Slovenia)
hasTimeIntervalSpecString TimeInterval-39 "[1950-01-01,1959-12-31]")
occursWithin(birth TimeInterval-39)

Clarifications *(1st part)*?

Improving annotations *(2nd)*

Proposal: Graphical time map **display** including fluent observations and events

- On-line **inference** to elucidate inter-relationships and potential contradictions
- Visual **feedback** to let users help assure quality themselves
- Time map-based **widgets** supporting user knowledge entry

Temporal inference scope in language research support

Typical:

- Strictly qualitative reasoning neglecting metric information from dates and durations.
- Quantitative information evaluated only locally—in temporal expressions (AKA "TIMEXs").

Our point-based engine...

- Accommodates lower and upper bounds on dates / times and durations widely available in text.
- Uses global constraint propagation to calculate points' earliest, latest possible dates / times that can inform annotators.

Fluent annotation workflow: Atemporal content

...Jansa graduated from Ljubljana University...

Source text

Formalization

attendsSchool(Janez_Jansa Ljubljana_University)

1. Select relation.
2. Specify argument identifiers, respecting co-reference.
3. Select / highlight / designate corresponding text.
4. Capture any counter-factual modality info.

Fluent annotation workflow: Temporal content

Reporting date

Dec 28, 2007…

*…Jansa **graduated from** Ljubljana University **in 1984**…*

1. Select one of time interval or point.
2. Capture any beginning date and backward clipping info.
3. Capture any **ending date** and **forward clipping info**.
4. Capture any duration info.
5. If ending point is unconstrained w.r.t. reporting date:
   a. Capture reporting aspect.
   b. Capture any reporting lag info.
6. Capture any other relative temporal info available.

Fluent clipped forward at ending point

attendsSchool(Janez_Jansa Ljubljana_University)

1984 — **Entered info (user writes)**

[,1984-12-31]

[1984-01-01,1984-12-31] — **Inferred info (user reads)**

Transition event annotation workflow

...Born on September 17, 1958 in Ljubljana, Jansa...

BirthEvent(Janez_Jansa, Ljubljana)
○
1958-09-17

1. Select event type.
2. Specify argument identifiers, respecting co-reference.
3. Select / highlight / designate corresponding text.
4. Capture any hypothetical modality info.
5. Capture any date info.
6. If an event's date is otherwise unconstrained w.r.t. reporting date:
   a. Capture reporting aspect.
   b. Capture any reporting lag info.
7. Capture any other relative temporal info available.

Displaying integrated time maps

BirthEvent(Janez_Jansa, Ljubljana)

○

1958-09-17

attendsSchool(Janez_Jansa Ljubljana_University)

○———————● [0D,15Y3M12D]

[1958-09-18,1984-12-31] 1984

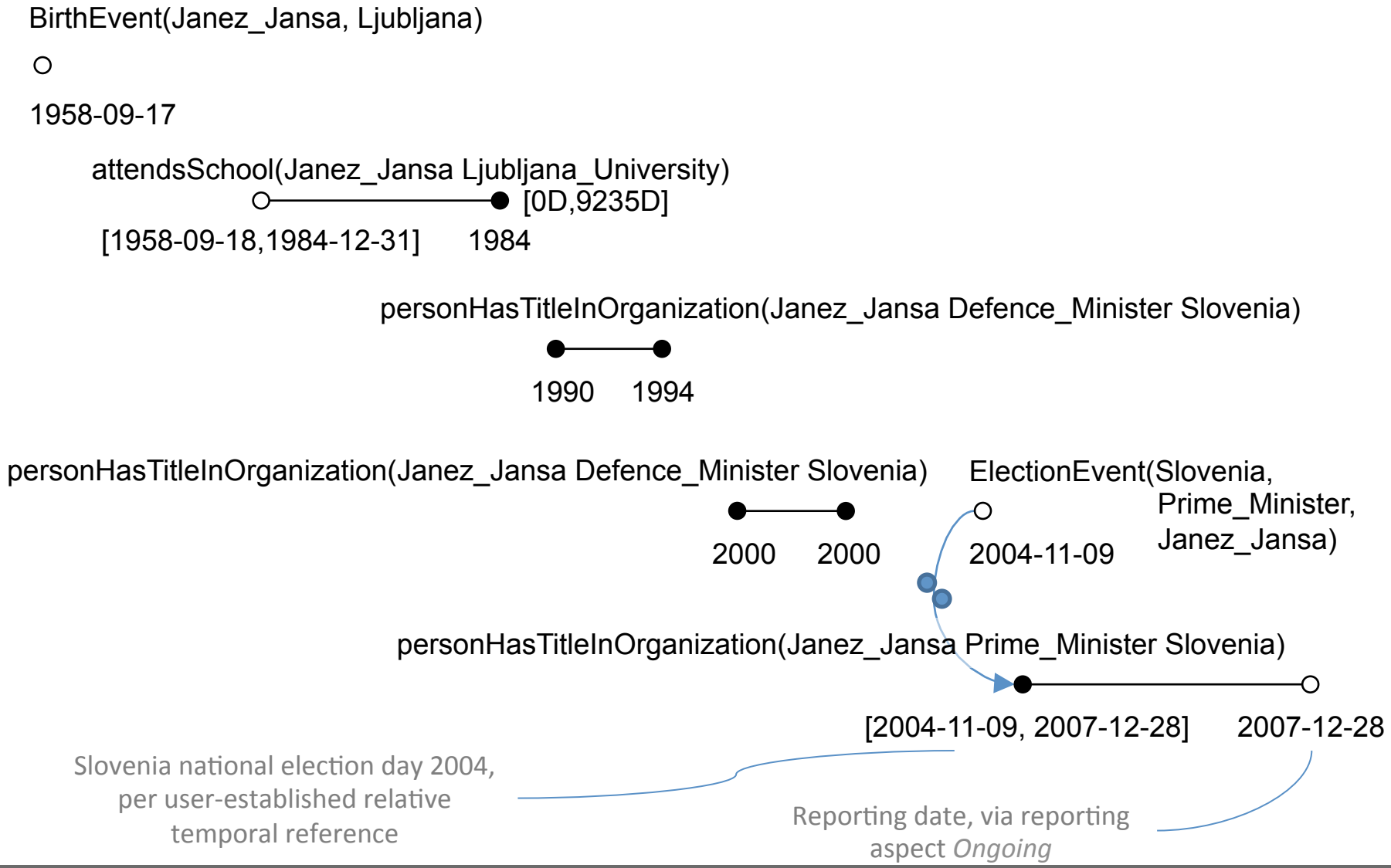
Automatically:

- Display in order any time points that are ordered unambiguously.
- Display inferred bounds.
  - *Rules:* Can't attend school before being alive; being born makes one alive.
- Highlight bounds contradictions.

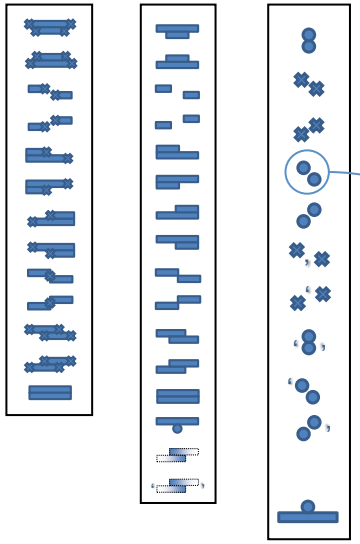
On demand:

- Trace back from bounds to user-entered information.
  - Date of the birth event
- Display / hide entered or inferred bounds on…
  - Beginning points, ending points
  - Durations
- Focus on a particular time window, location, person, …
- Highlight time points that are ordered / unordered w.r.t. to a selected, reference time point.

Displaying integrated time maps



Establishing relative temporal reference



Widget: User selects…

- *Relation (from menu)*
- *Subject , object (via mouse)*

ElectionEvent(Slovenia, Prime_Minister, Janez_Jansa)

2004-11-09

personHasTitleInOrganization(Janez_Jansa Prime_Minister Slovenia)

Expected formal knowledge

F_school: attendsSchool (Janez_Jansa Ljubljana_University)

holdsThroughout(F_school I_school)
clippedForward(F_school I_school)
hasTimeIntervalSpecString(I_school [,1984])
hasPersonBorn(birth Janez_Jansa)
occursAt(birth P_birth)
hasTimePointSpecString(P_birth 1958-09-17)
hasPersonBorn(birth Janez_Jansa)
hasBirthEventGPE-spec(birth GPEspec)
hasCityTownOrVillage(GPEspec ljubljana_Ljubljana_Slovenia)
hasNationState(GPEspec Slovenia)
type(Defence_Minister MinisterTitle)

BirthEvent(Janez_Jansa, Ljubljana)

F_PTIO_DM_1: personHasTitleInOrganization(Janez_Jansa Defence_Minister Slovenia)

holdsThroughout(F_PTIO_DM_1 I_PTIO_DM_1)
clippedBackward(F_PTIO_DM_1 I_PTIO_DM_1)
clippedForward(F_PTIO_DM_1 I_PTIO_DM_1)
hasTimeIntervalSpecString(I_PTIO_DM_1 [1990,1994])

F_PTIO_DM_2: personHasTitleInOrganization(Janez_Jansa Defence_Minister Slovenia)

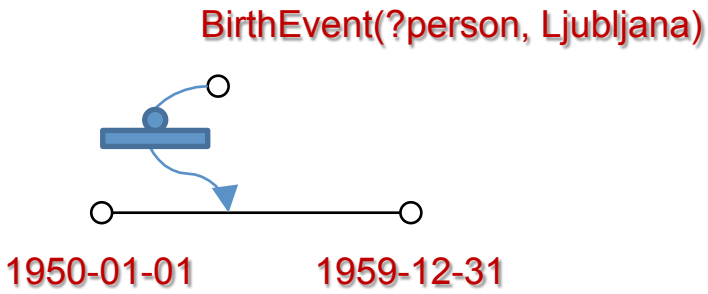
holdsThroughout(F_PTIO_DM_2 I_PTIO_DM_2)
clippedBackward(F_PTIO_DM_2 I_PTIO_DM_2)
clippedforward(F_PTIO_DM_2 I_PTIO_DM_2)
hasTimeIntervalSpecString(I_PTIO_DM_2 [2000,2000])

F_PTIO_PM: personHasTitleInOrganization(Janez_Jansa Prime_Minister Slovenia)

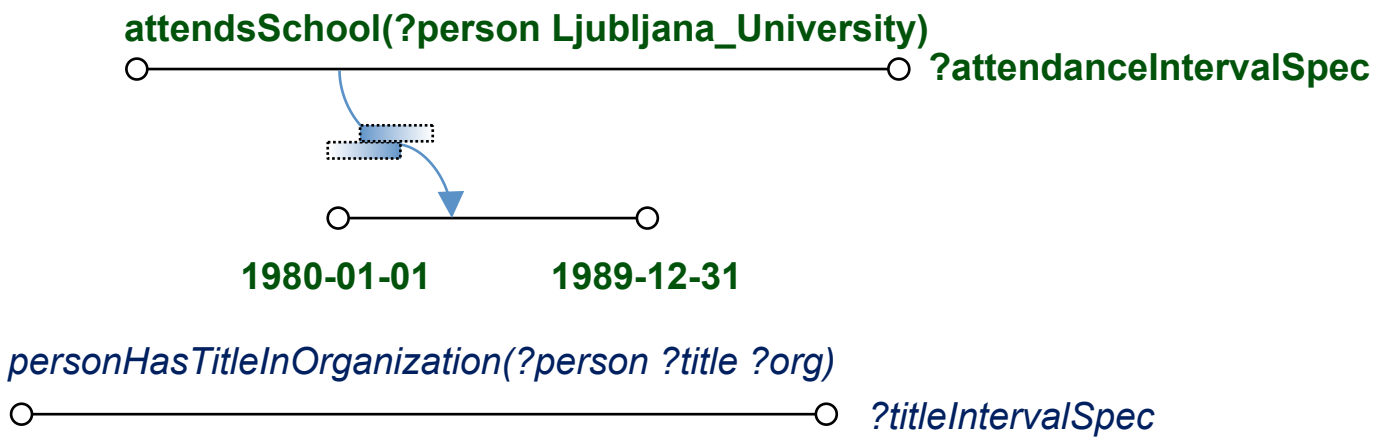
holdsThroughout(F_PTIO_PM I_PTIO_PM)
clippedBackward(F_PTIO_PM I_PTIO_PM)
hasBeginningTimePoint(I_PTIO_PM I_PTIO_PM_beginning)
hasTimePointSpecString(Slovenia_2004_Election_Day 2004-11-09)
timePointGreaterThanOrEqualTo(I_PTIO_PM_beginning Slovenia_2004_Election_Day)
hasReportingAspect(I_PTIO_PM Ongoing)
ref(annotation I_PTIO_PM)
annotation(document annotation)
hasReportingChronusSpecString(document 2007-12-28)

Clarifications *(2nd part)*?

Adaptation to query authoring *(3rd)*



Query 1: *Find all* ***persons who were born in Ljubljana in the 1950s*** *and* ***attended Ljubljana University in the 1980s****, the titles that they held, the organizations in which they held these titles, and the maximal known time periods over which* ***they attended*** *and held these titles.*



Query Formalization 4

?F_school attendsSchool(?person Ljubljana_University)
hasTimeIntervalSpecString(?I_range_school "[1980-01-01,1989-12-31]")
holdsWithin(?F_school ?I_range_school)
maximallyHoldsThroughout(?F_school ?I_school)
hasTimeIntervalSpecString(?I_school ?aiss)

?F_title personHasTitleInOrganization(?person ?title ?org)
maximallyHoldsThroughout(?F_title ?I_title)
hasTimeIntervalSpecString(?I_title ?tiss)

hasPersonBorn(?birth ?person)
hasBirthEventGPE-spec(?birth ?GPEspec)
hasNationState(?GPEspec Slovenia)
hasCityTownOrVillage(?GPEspec ljubljana_Ljubljana_Slovenia)
hasTimeIntervalSpecString(?I_range_birth "[1950-01-01,1959-12-31]")
occursWithin(?birth ?I_range_birth)

Questions *(1st paper)*?

Digital Sandbox

Best-practice time point ontology for event calculus-based temporal reasoning

Robert C. Schrag, Ph.D.

STIDS 2012

Machine reading context

- Our temporal reasoning engine implementation was based on an axiom set that…
  - (Naively) treats least time units (viz. days) as time "points."
  - Was intended to simplify the axiom set specified to reading teams.
  - Resulted in anomalous event calculus representation and reasoning.

Naïve time point anomalies

Within a finest represented time unit (e.g., day)

…

A. Can't order time points
B. Can't order contradictory fluent observations (so, can't avoid contradictions)
C. Can't order events
D. Can't avoid contradictions caused by events that should be ordered

[lower, upper] bounds on the calendar or clock distance (in the naïve approach) from time point *S* to time point *O*

Subject on top
Object on bottom

timePointEqualTo(*S*,*O*)		[0, 0]
timePointLessThan(*S*,*O*)		[1, ∞]
timePointGreaterThan(*S*,*O*)		[−∞, −1]
timePointLessThanOrEqualTo(*S*,*O*)		[0, ∞]
timePointGreaterThanOrEqualTo(*S*,*O*)		[−∞, 0]
hasNextTimePoint(*S*,*O*)		[1, 1]
hasPreviousTimePoint(*S*,*O*)		[−1, −1]
timePointTouches(*S*,*O*)		[−1, 1]
timePointLessThanOrTouching(*S*,*O*)		[−1, ∞]
timePointGreaterThanOrTouching(*S*,*O*)		[−∞, 1]
pointInInterval(*S*,*O*)		
pointIsInterval(*S*,*O*)		

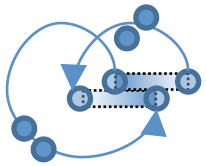
✖ *marked time points may not coincide.*

, *marked time points are consecutive.*

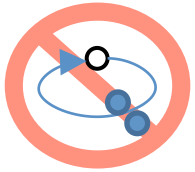
∞ = *Infinite duration*

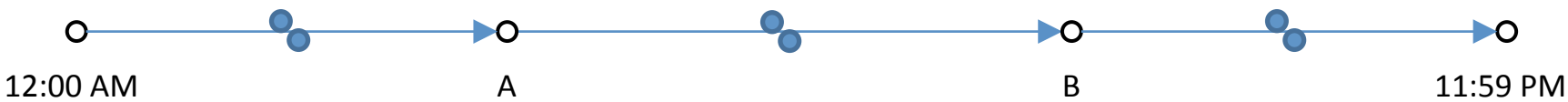
Subject on top
Object on bottom

timeIntervalIntersects(*S*,*O*)

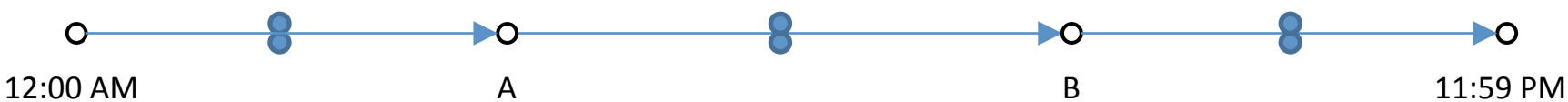


Anomaly A: **Can’t order time points**

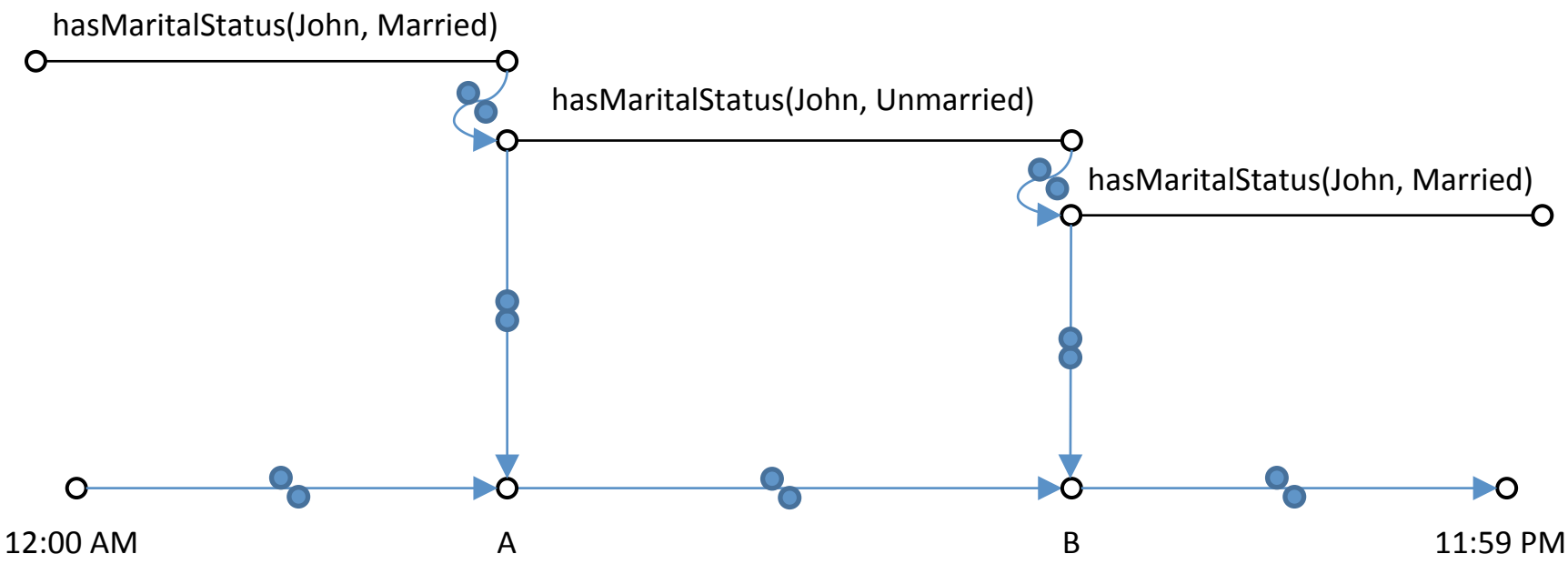




We can’t have both the above and the below—*all points equal to the full day.*

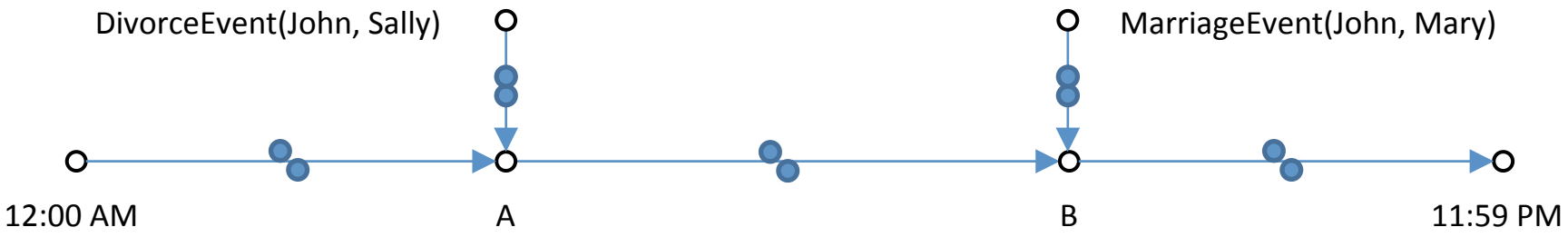


Anomaly B: **Can't order contradictory fluents**



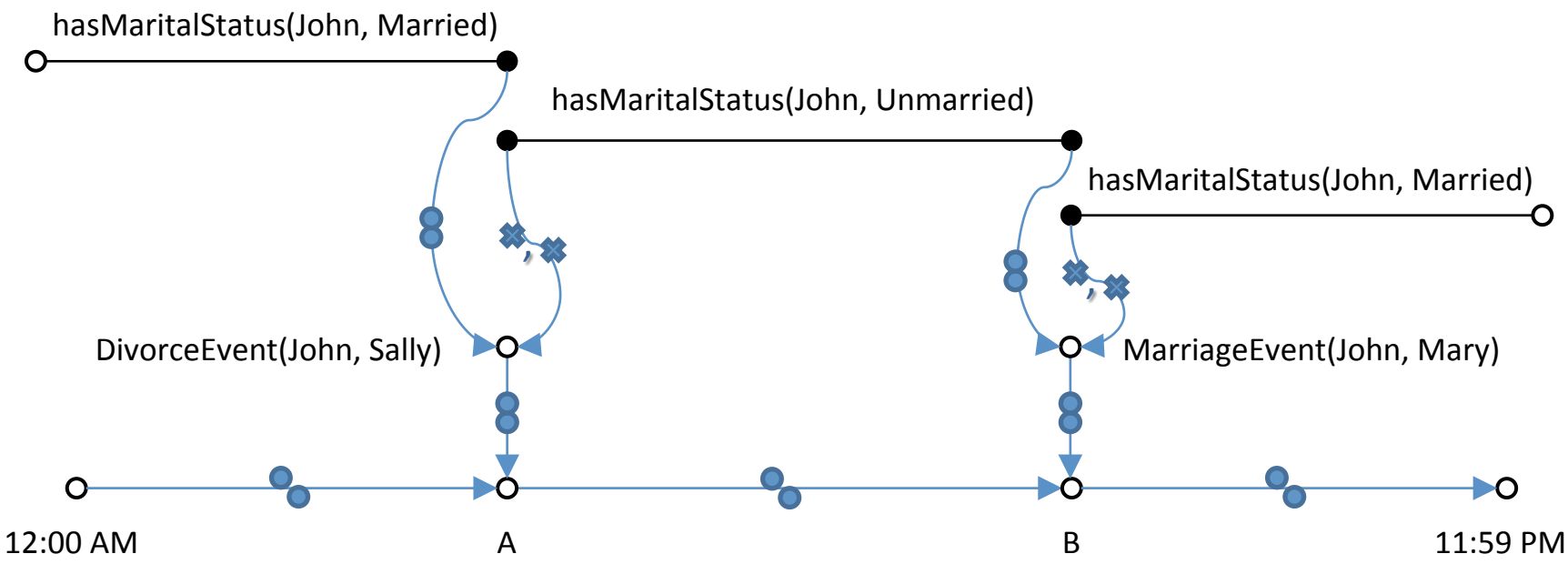
When we can't order points, we can't separate fluents holding within the same day, so can't avoid spurious contradiction.

Anomaly C: **Can’t order events**



When we can’t order points, we can’t order events.

Anomaly D: **Can't avoid event-initiated contradictions**



When we can't order events, we can't avoid spurious contradiction regarding the fluents they terminate and initiate.

Solution *(as in Tom Dean's TMM system)*

- Introduce infinitesimal temporal distance, ϵ.
- Represent bounds on point-to-point distance using two dimensions:
  - Integer / infinite part ("1" = finest time unit)
  - Infinitesimal part ("1" = "adjacency" distance)
- Track these dimensions separately during constraint propagation.
  - Implementation: Exploit complex number arithmetic.

[lower, upper] bounds on the calendar or clock distance (in the naïve approach) from time point *S* to time point *O*

Subject on top
Object on bottom

timePointEqualTo(*S*,*O*)		[0, 0]
timePointLessThan(*S*,*O*)		[1, ∞]
timePointGreaterThan(*S*,*O*)		[−∞, −1]
timePointLessThanOrEqualTo(*S*,*O*)		[0, ∞]
timePointGreaterThanOrEqualTo(*S*,*O*)		[−∞, 0]
hasNextTimePoint(*S*,*O*)		[1, 1]
hasPreviousTimePoint(*S*,*O*)		[−1, −1]
timePointTouches(*S*,*O*)		[−1, 1]
timePointLessThanOrTouching(*S*,*O*)		[−1, ∞]
timePointGreaterThanOrTouching(*S*,*O*)		[−∞, 1]
pointInInterval(*S*,*O*)		
pointIsInterval(*S*,*O*)		

✖ *marked time points may not coincide.*

, *marked time points are consecutive.*

∞ = *Infinite duration*

[lower, upper] bounds on the calendar or clock distance (in the **preferred** approach) from time point *S* to time point *O*

Subject on top
Object on bottom

timePointEqualTo(*S*,*O*)		[0, 0]
timePointLessThan(*S*,*O*)		[ϵ, ∞]
timePointGreaterThan(*S*,*O*)		[−∞, −ϵ]
timePointLessThanOrEqualTo(*S*,*O*)		[0, ∞]
timePointGreaterThanOrEqualTo(*S*,*O*)		[−∞, 0]
hasNextTimePoint(*S*,*O*)		[ϵ, ϵ]
hasPreviousTimePoint(*S*,*O*)		[−ϵ, −ϵ]
timePointTouches(*S*,*O*)		[−ϵ, ϵ]
timePointLessThanOrTouching(*S*,*O*)		[−ϵ, ∞]
timePointGreaterThanOrTouching(*S*,*O*)		[−∞, ϵ]

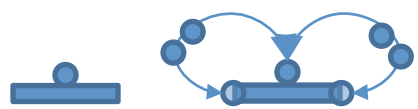
marked time points may not coincide.

marked time points are consecutive.

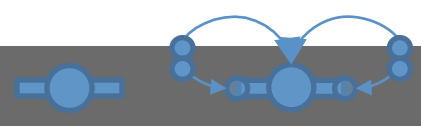
∞ = Infinite duration

ϵ = Infinitesimal duration

pointInInterval(*S*,*O*)



pointIsInterval(*S*,*O*)



Constraint propagation axioms

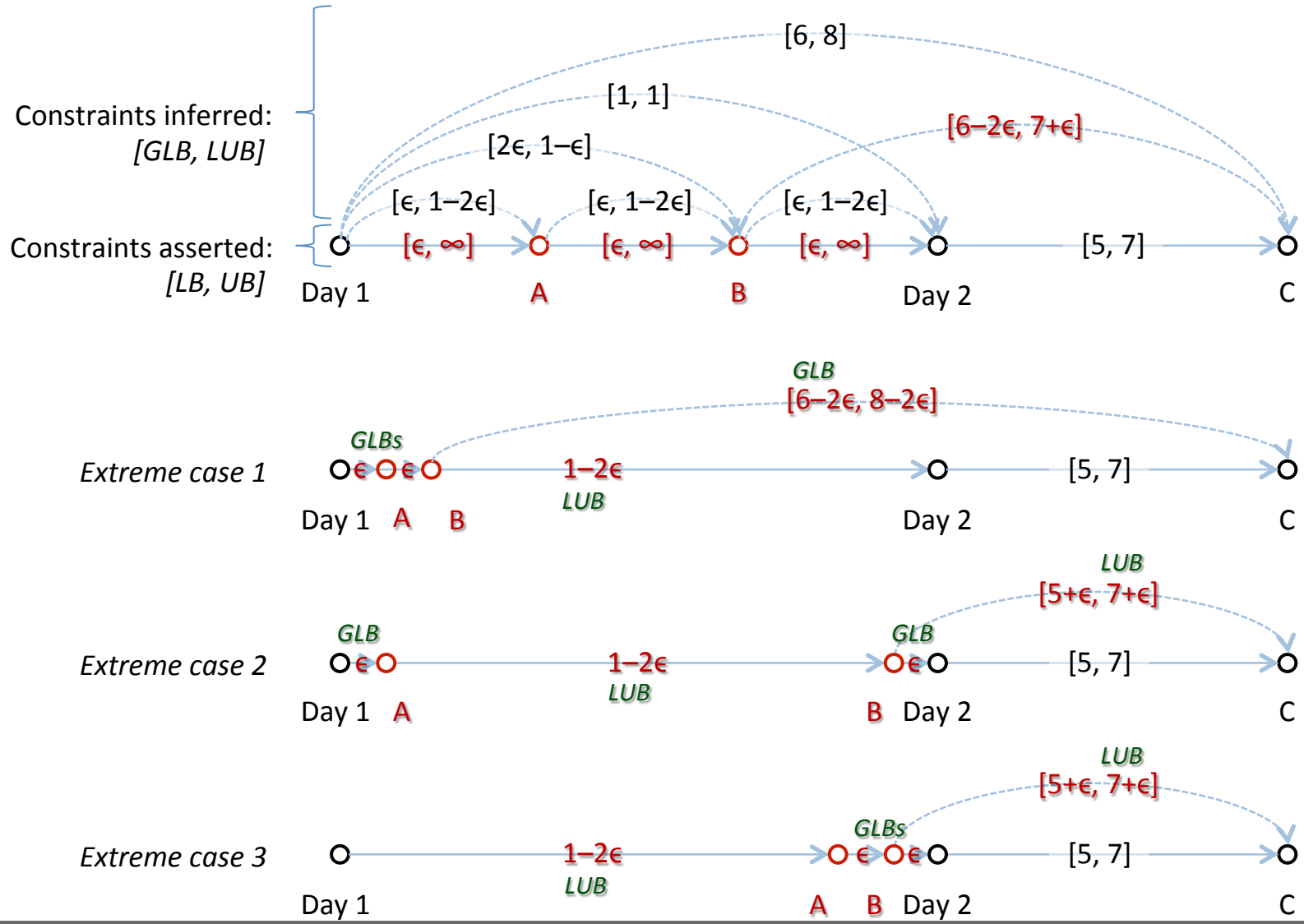
Tighten lower bound:
distance(a, b, [w, y]) ← distance(a, b, [x, y]) ∧ distance(a, b, [w, z]) ∧ infinite>(w, x)

Tighten upper bound:
distance(a, b, [x, z]) ← distance(a, b, [x, y]) ∧ distance(a, b, [w, z]) ∧ infinite>(y, z)

Use sum to compose constraints sharing a point at opposite ends:
distance(a, c, [mo, np]) ← distance(a, b, [m, n]) ∧ distance(b, c, [o, p])
∧ infinite+(m, o, mo)
∧ infinite+(n, p, np)

Reverse constraint direction:
distance(b, a, [–y, –x]) ↔ distance(a, b, [x, y]) ∧ infinite–(x, –x) ∧ infinite–(y, –y)

Constraint propagation example

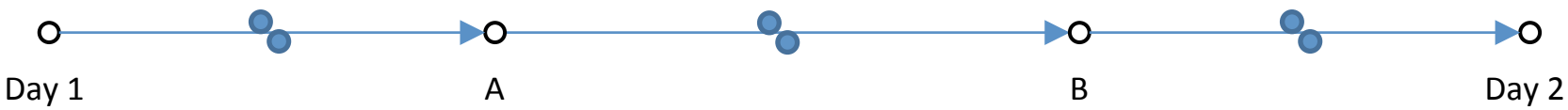


Arithmetic axioms handling infinities *and infinitesimals*

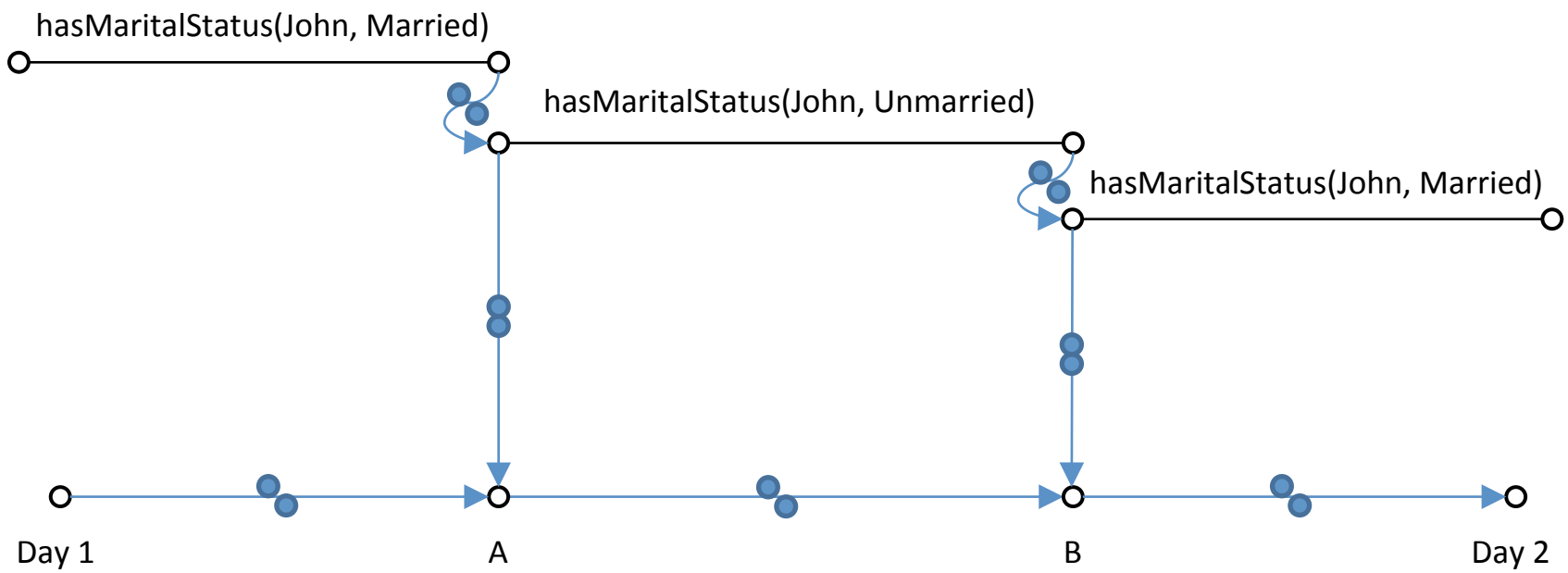
infinite(–∞)
infinite(∞)
infinite+(–∞, –∞, –∞)
infinite+(∞, ∞, ∞)
infinite+(a, –∞, –∞) ← ¬infinite(a)
infinite+(–∞, b, –∞) ←¬infinite(b)
infinite+(a, ∞, ∞)← ¬infinite(a)
infinite+(∞, b, ∞) ← ¬infinite(b)
infinite+(a, b, a + b) ← ¬infinite(a) ^ ¬infinite(b)
infinite–(–∞, ∞)
infinite–(∞, –∞)
infinite–(a, –a) ← ¬infinite(a)
infinite>(∞, –∞)
infinite>(a, –∞) ← ¬infinite(a)
infinite>(∞, b) ← ¬infinite(b)
infinite>(a, b) ← ¬infinite(a) ^ ¬infinite(b) ^ finite>(a, b)
finite>(a, b) ← real>(a, b) ˇ *(real=(a, b) ^ imaginary>(a, b))*

Straightforward when the underlying program language supports *complex arithmetic.*

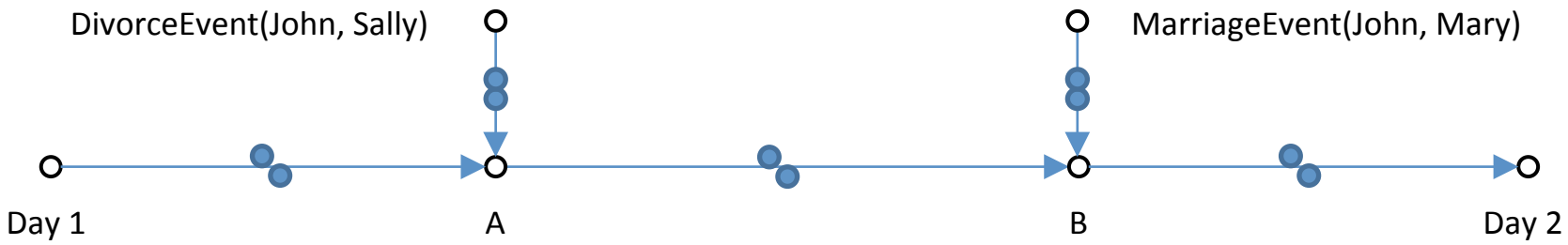
Anomaly A: **Can't order time points**



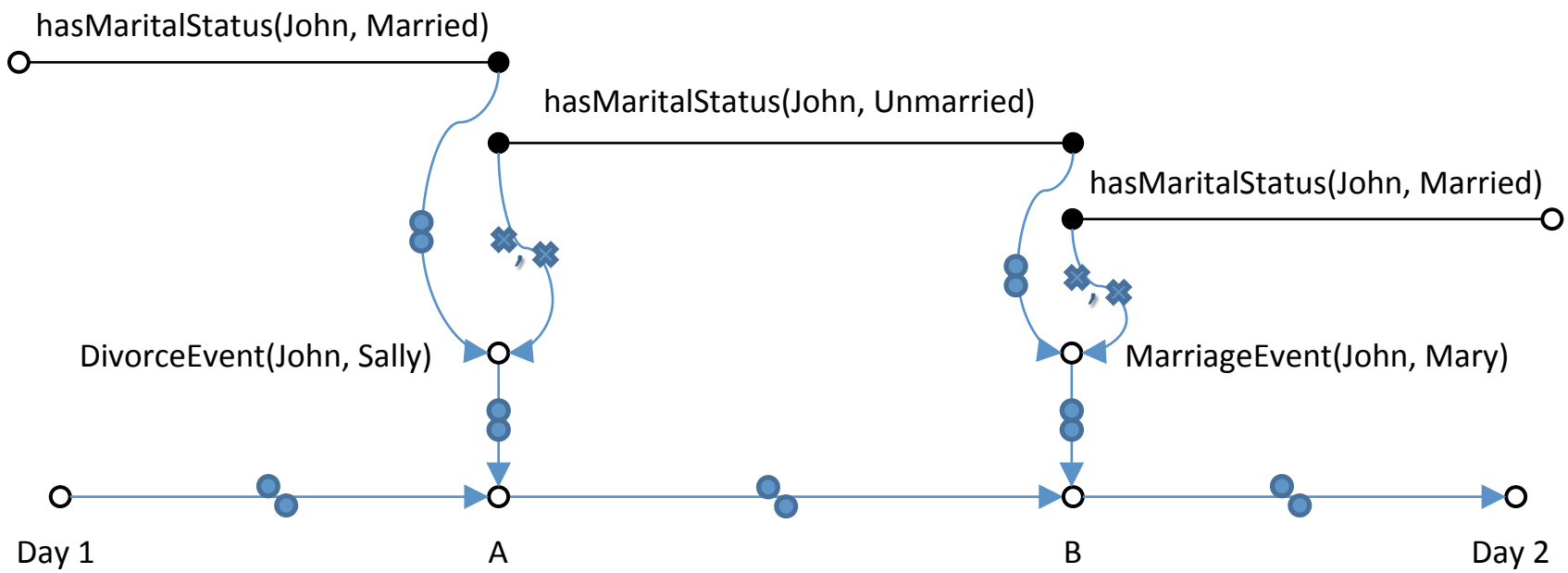
Anomaly B: **Can't order contradictory fluents**



Anomaly C: **Can't order events**



Anomaly D: **Can't avoid event-initiated contradictions**



Questions ?