Handling time in RDF

Claudio Gutierrez
(Joint work with C. Hurtado and A. Vaisman)

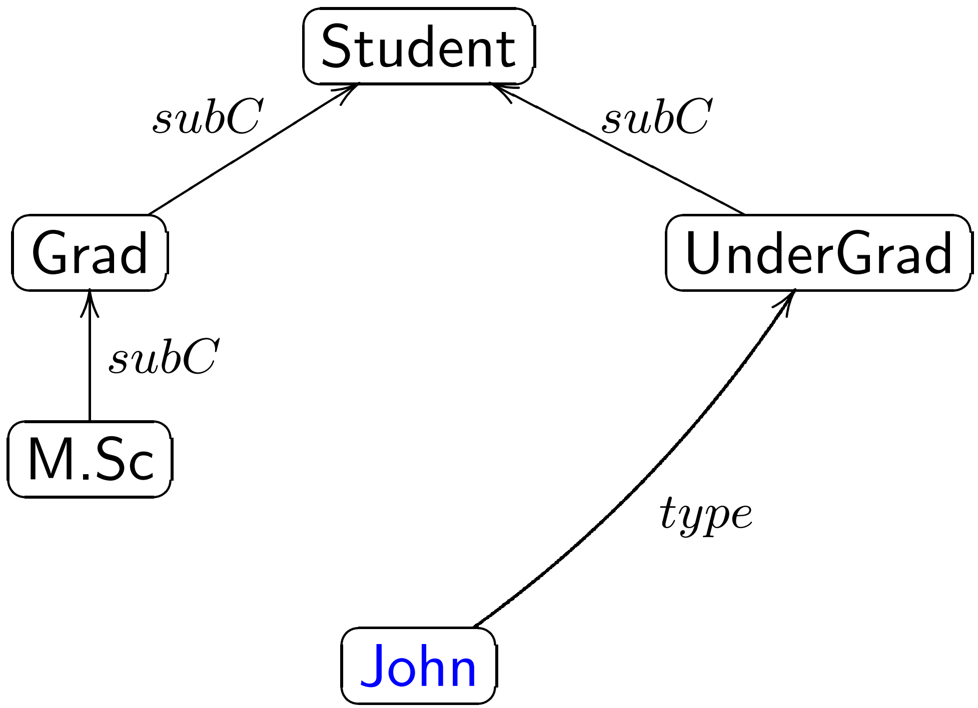
Department of Computer Science
Universidad de Chile

UPM, Madrid, January 2009

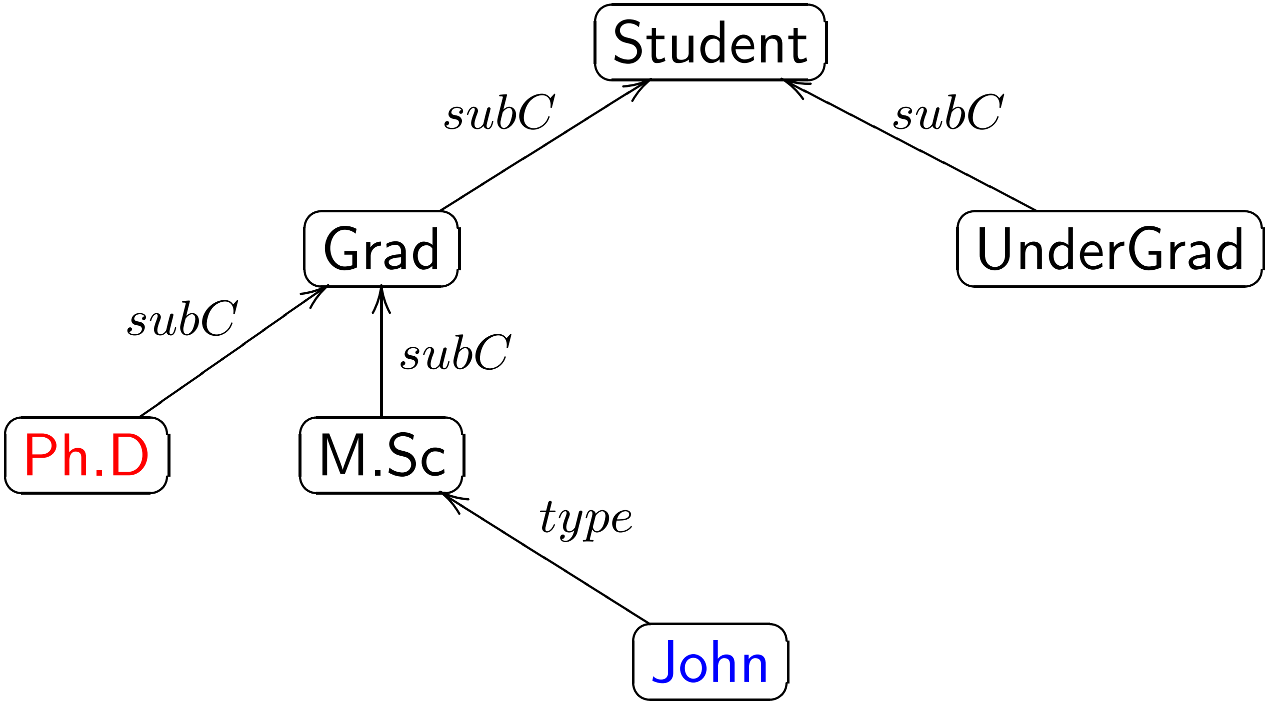
Outline

- Introducing time into RDF
- Temporal RDF Graphs
- Semantics of Temporal RDF Graphs
- Syntax for Temporal Graphs
- Querying Time in RDF

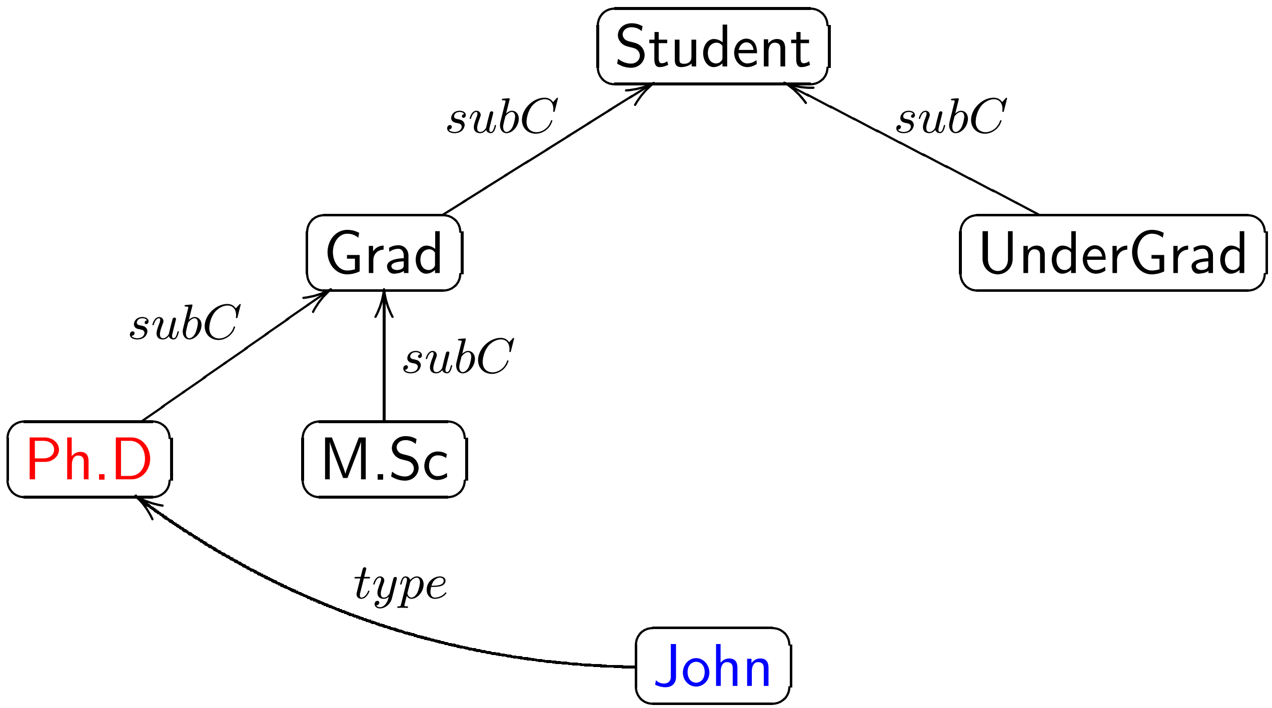
Introducing time into RDF



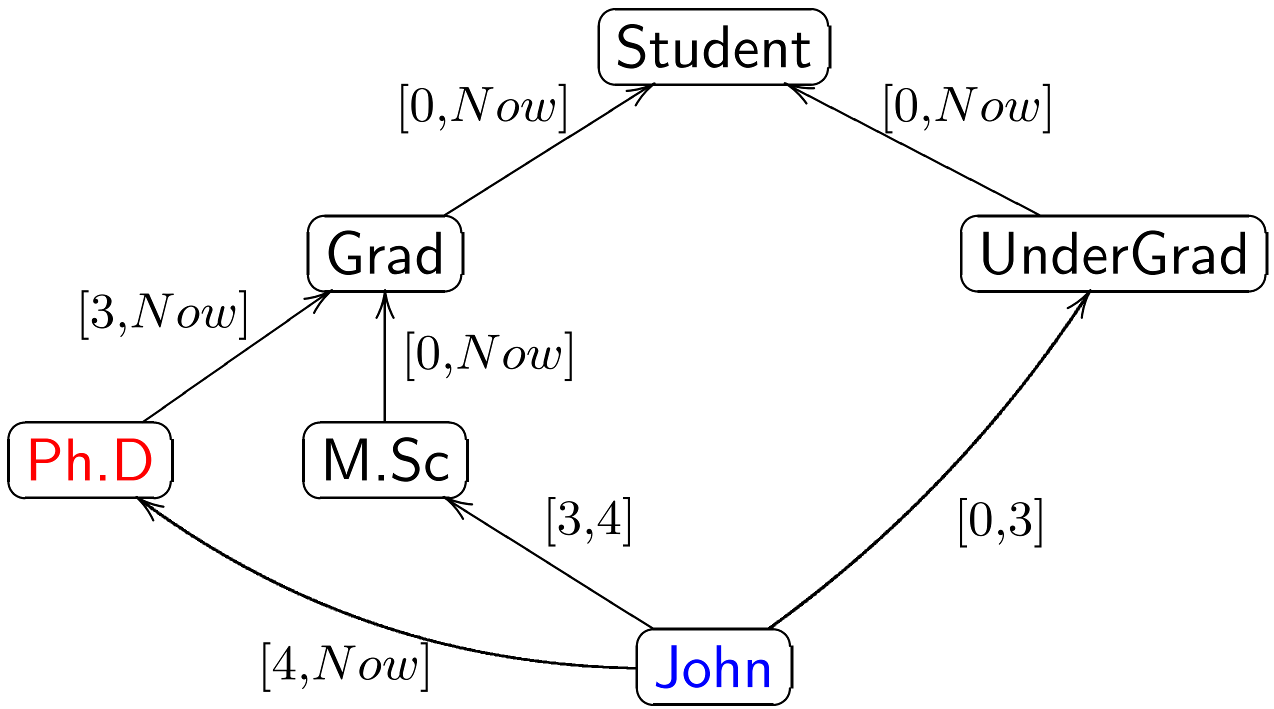
Introducing time into RDF



Introducing time into RDF



Temporal Graph



General Issues

- Versioning versus Labeling
  - Label elements subject to change
  - Maintain a snapshot of each state of the graph

General Issues

- Versioning versus Labeling
  - Label elements subject to change
  - Maintain a snapshot of each state of the graph
- Time Points versus Time Intervals.

$$[4, 31] = [4] \cup [5] \cup \cdots \cup [30] \cup [31]$$

General Issues

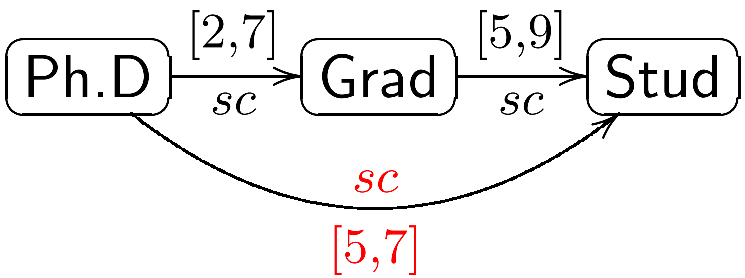
- Versioning versus Labeling
  - Label elements subject to change
  - Maintain a snapshot of each state of the graph
- Time Points versus Time Intervals.

$$[4, 31] = [4] \cup [5] \cup \cdots \cup [30] \cup [31]$$

- Temporal Query Language
  - Point based (variables refer to point times)
  - Interval based (variables refer to intervals)

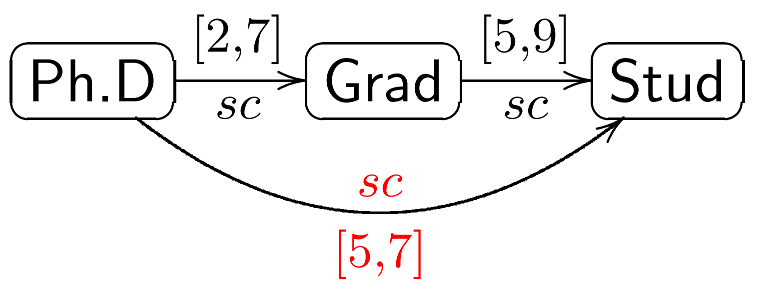
RDF Intrinsic Issues

- Notion of temporal Entailment $\models_\tau$

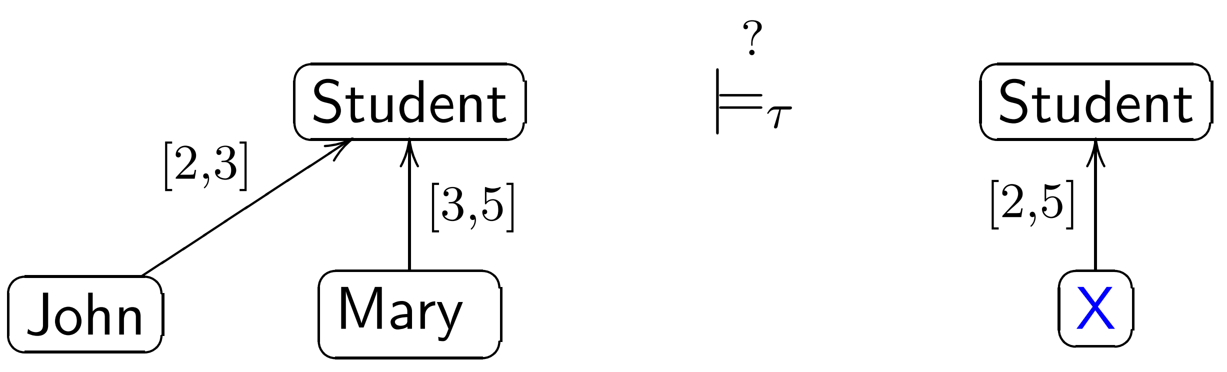


RDF Intrinsic Issues

- Notion of temporal Entailment $\models_\tau$

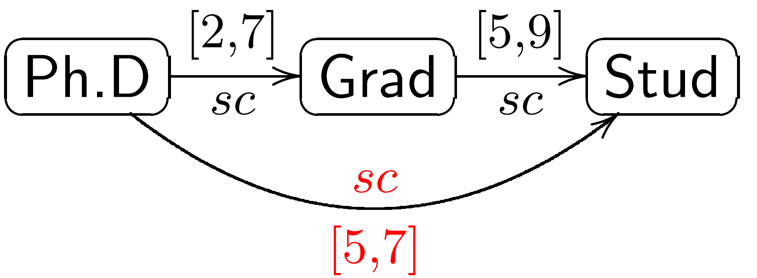


- Treatment of temporal Blank Nodes:

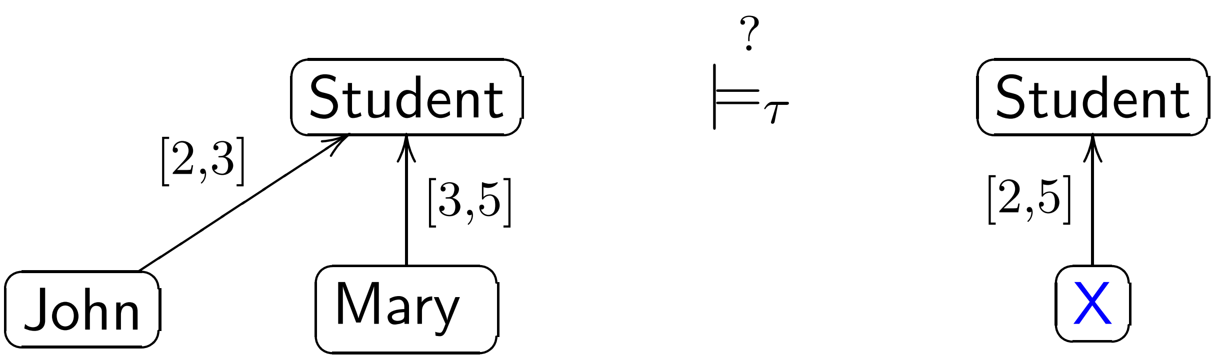


RDF Intrinsic Issues

- Notion of temporal Entailment $\models_\tau$



- Treatment of temporal Blank Nodes:



- Vocabulary for temporal labeling

Definitions

Temporal Triple: an RDF triple with a temporal label, e.g. $(a, b, c)[t]$

Temporal Graph: set of temporal triples

Snapshot of graph G at time t:

$$G(t) = \{(a, b, c) \ : \ (a, b, c)[t] \in G\}$$

Notion of temporal entailment $G_1 \models_\tau G_2$

Semantics

Ground Case:

$$G_1 \models_\tau G_2 \text{ if for each } t,\ G_1(t) \models G_2(t)$$

Semantics

Ground Case:

$$G_1 \models_\tau G_2 \text{ if for each } t,\ G_1(t) \models G_2(t)$$

Non Ground Case:
$G_1 \models_\tau G_2$ if there are ground instances $\mu_1(G_1)$ and $\mu_2(G_2)$ such that for each t:

$$\mu_1(G_1)(t) \models_\tau \mu_2(G_2)(t)$$

Semantics

Ground Case:

$$G_1 \models_\tau G_2 \ \text{ if for each } t,\ G_1(t) \models G_2(t)$$

Non Ground Case:
$G_1 \models_\tau G_2$ if there are ground instances $\mu_1(G_1)$ and $\mu_2(G_2)$ such that for each t:

$$\mu_1(G_1)(t) \ \models_\tau \ \mu_2(G_2)(t)$$

Proposition. For ground graphs, $G_1 \models_\tau G_2$ implies $G_1(t) \models G_2(t)$ for all times t.

Semantics (cont.)

The temporal closure $tcl(G)$ is a maximal set of temporal triples G' such that:

– G' contains G

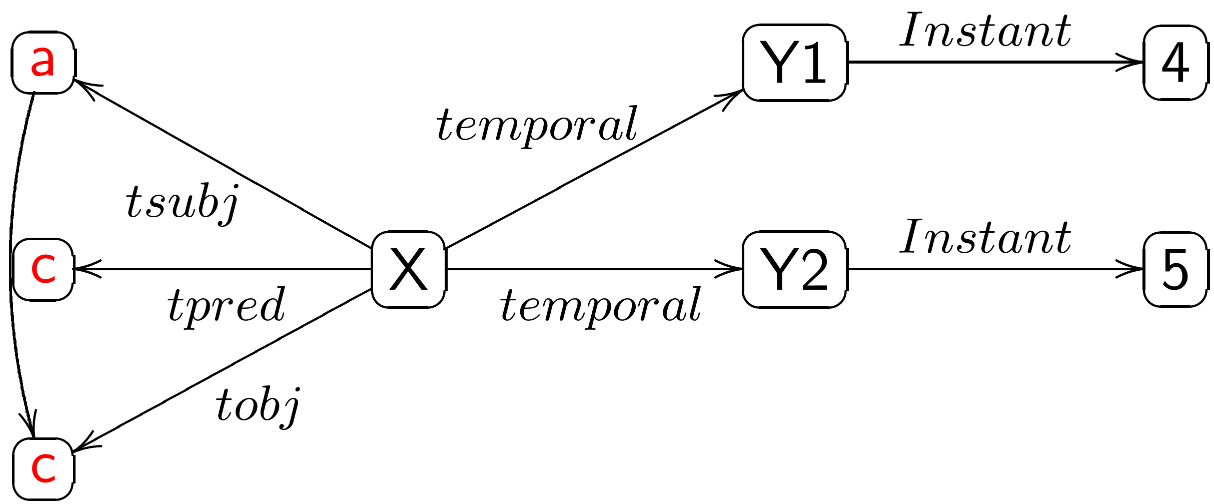
– G is equivalent to G'

Proposition.
$G_1 \models_\tau G_2$ iff $tcl(G_1) \models_\tau G_2$

Proposition. Deciding if G' is the closure of G is DP-complete.

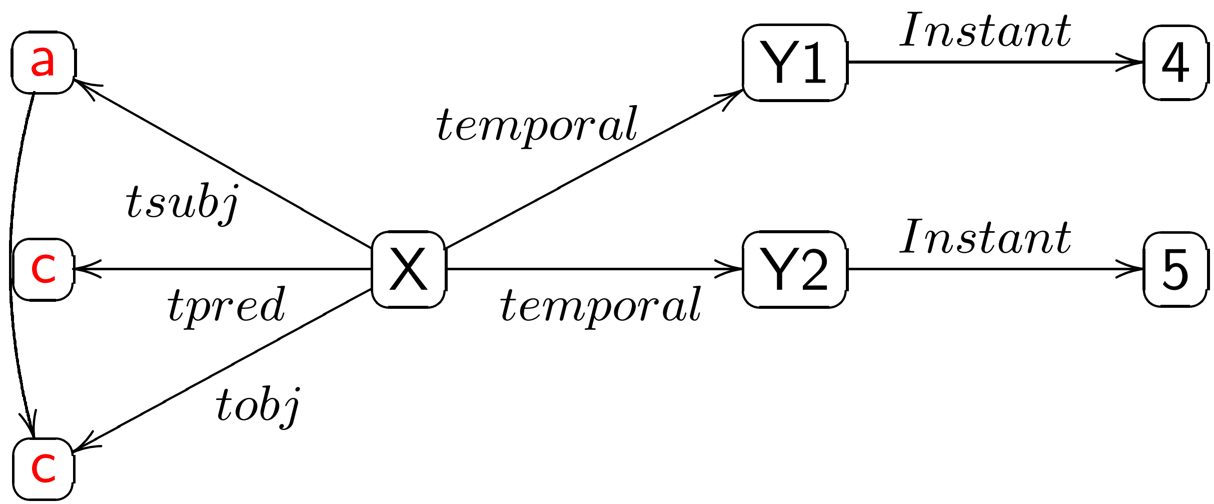
Syntax for $(a, b, c)[4, 5]$

- Point version

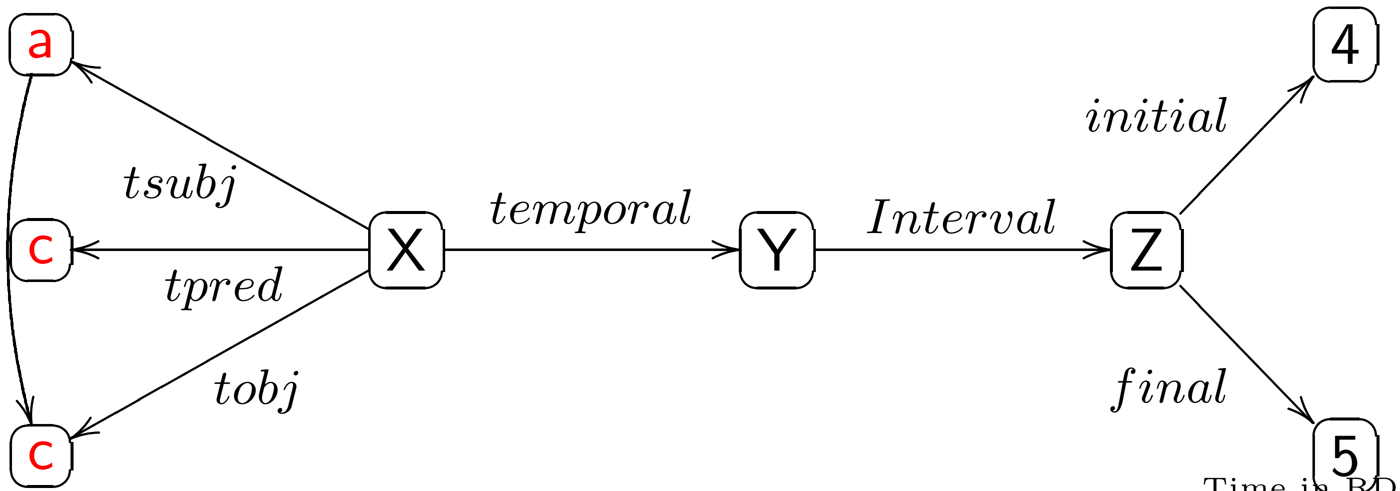


Syntax for $(a, b, c)[4, 5]$

- Point version



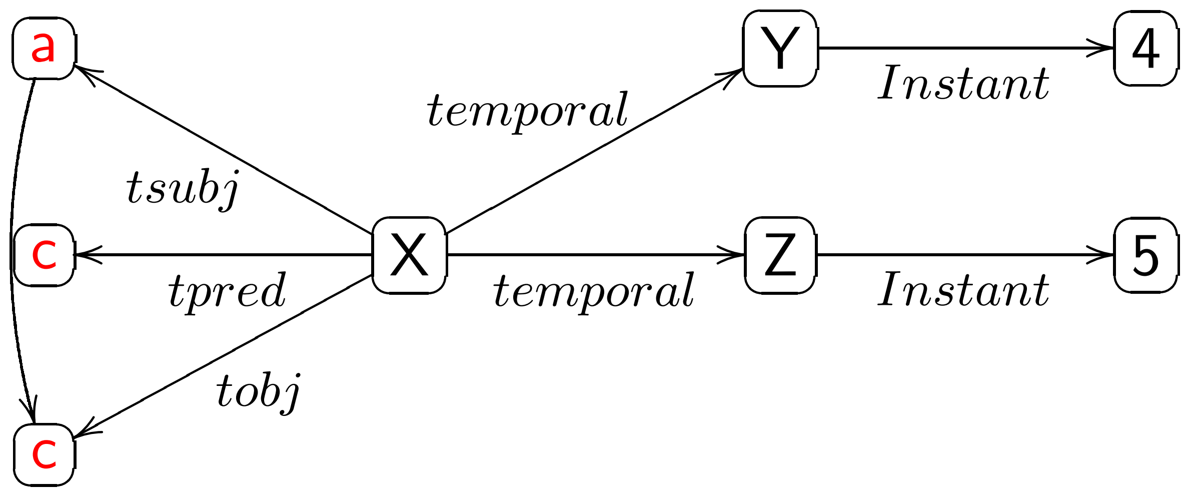
- Interval version



Syntax (cont.): rules

Rule 1-2: Equivalence betwen point and interval versions

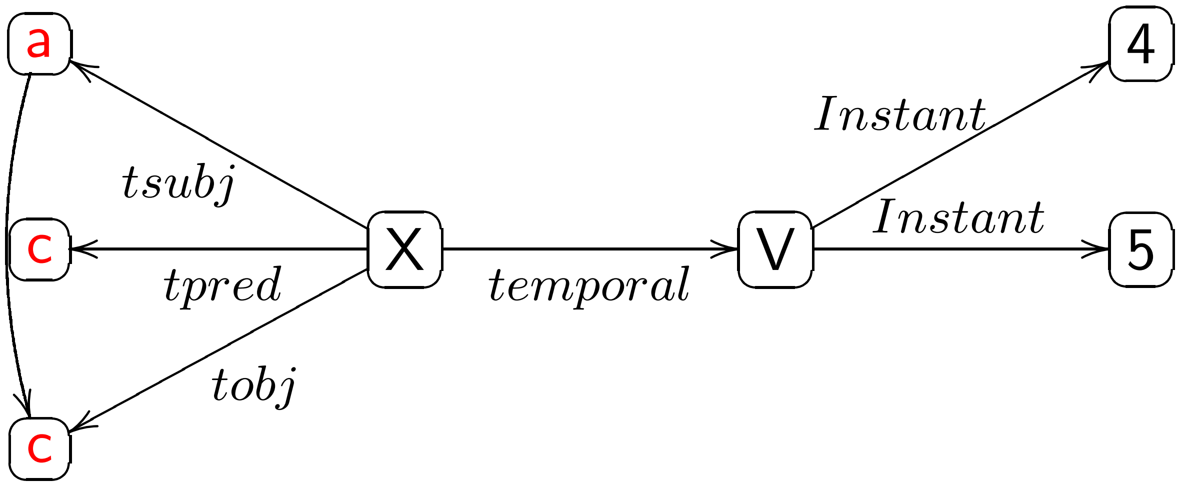
Rule 3: Normalization of point-version:



Syntax (cont.): rules

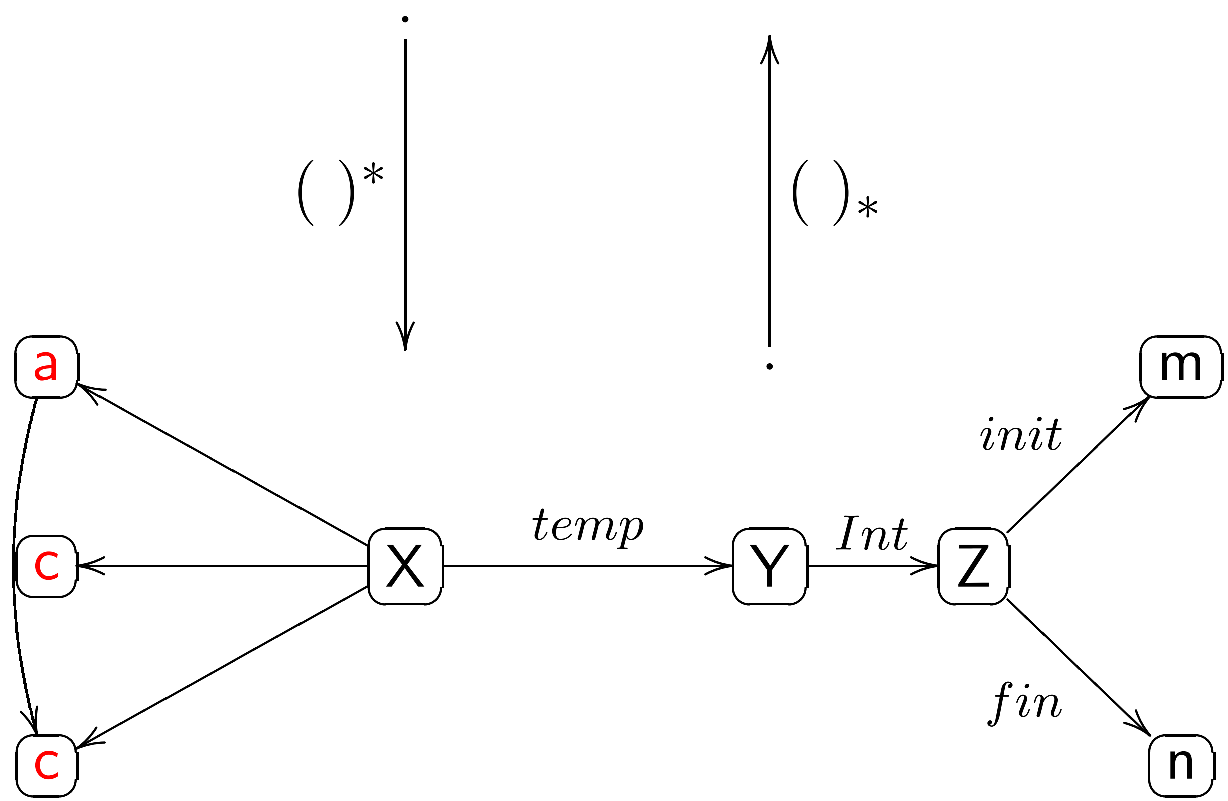
Rule 1-2: Equivalence betwen point and interval versions

Rule 3: Normalization of point-version:



Syntax works well

$$(a, b, c)[m, n]$$



Syntax works well (cont.)

Theorem.

1. $G_1 \models_\tau G_2$ implies $(G_1)^* \models (G_2)^*$
2. $G_2 \models G_2$ implies $(G_1)_* \models_\tau (G_2)_*$
3. $(G^*)_* = G$ and $G \models (G_*)^*$

Theorem. Let $\vdash$ be the deductive system formed by RDFS rules plus Temporal rules. Then:
$G_1 \models_\tau G_2$ iff $(G_1)^* \vdash (G_2)^*$

Querying Temporal RDF

Proposal: Conjunctive fragment with
– interval and point variables
– aggregate functions
– constructor of graphs for answers

Querying Temporal RDF

Proposal: Conjunctive fragment with
– interval and point variables
– aggregate functions
– constructor of graphs for answers

- Students who have taken a Master course between year 2000
- Students taking Ph.D courses together and the time when this occurred
- Time intervals when the IT Master program was offered
- Students applying for jobs at time t after finishing their Ph.D program in no more than 4 years

What we have:

1. Semantics for Temporal RDF graphs
2. Syntax to incorporate the framework into standard RDF
3. Sound and complete inference rules for temporal graphs
4. Complexity bounds showing temporal RDF preserves complexity of RDF
5. Sketch of Temporal RDF query language