

Institut für Informatik
LFE für Programmier- und Modellierungssprachen

Instantiating Hierarchical Timed Automata for Semantic Query Optimization of Complex Event Processing

Sandro Gießl

Projektarbeit im Rahmen des
Fortgeschrittenenpraktikums

Betreuung: Prof. Dr. François Bry,
Olga Poppe

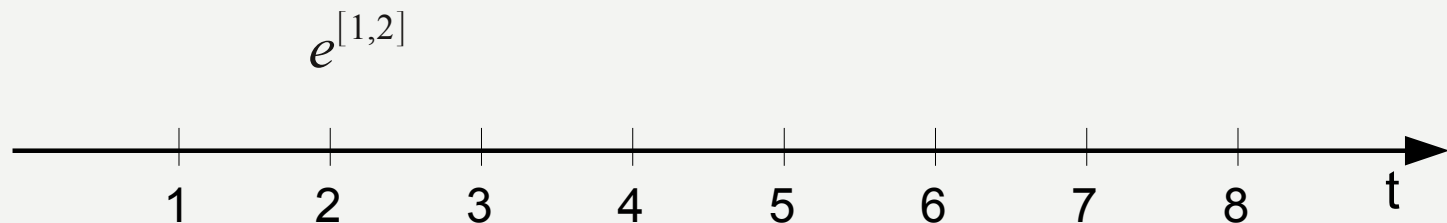


Outline of the Talk

1. Introduction
2. Related Work
3. Syntax and Semantics of Instantiating Hierarchical Timed Automata (IHTA)
4. Conclusion
5. Future Work

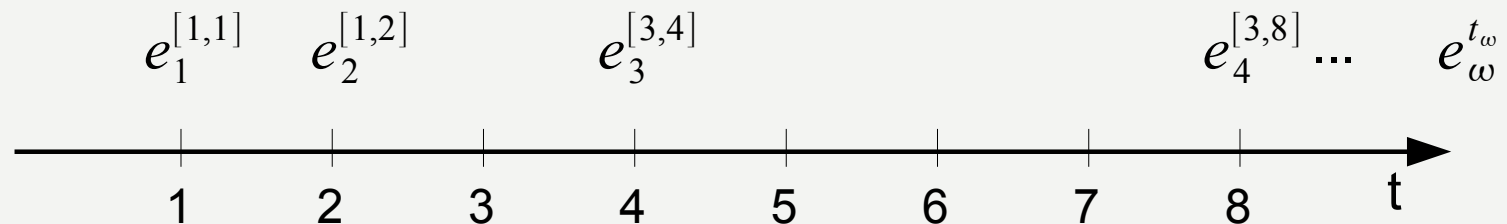
Complex Event Processing (CEP)

- Event
 - Message of something happening
 - at a specific time (the *occurrence time*)
- Event Stream
- Event Query



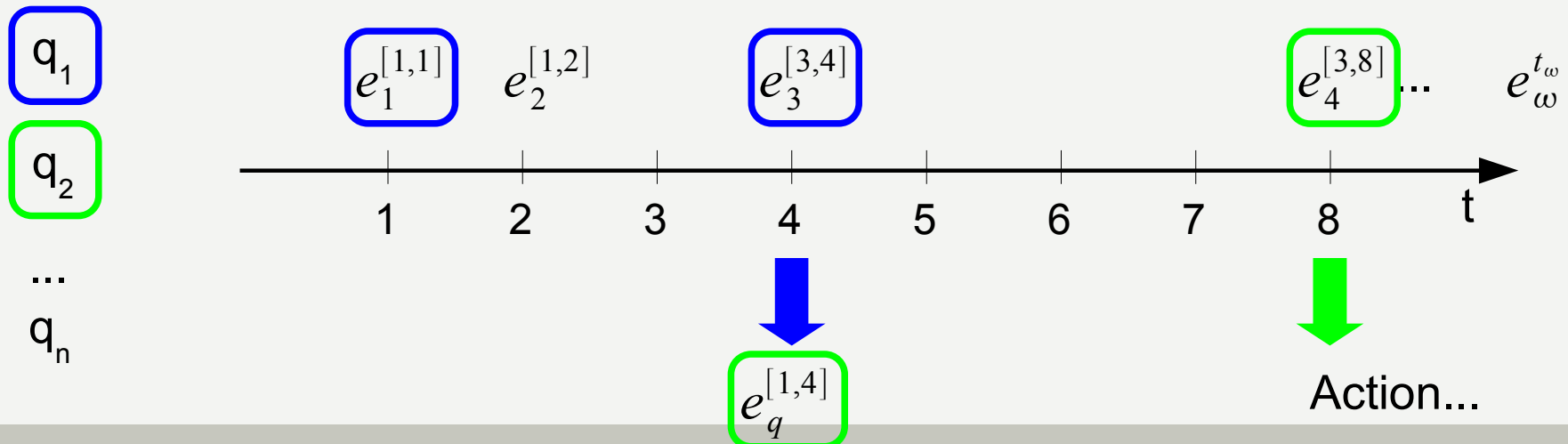
Complex Event Processing (CEP)

- Event
 - Message of something happening
 - at a specific time (the *occurrence time*)
- Event Stream
- Event Query



Complex Event Processing (CEP)

- Event
 - Message of something happening
 - at a specific time (the *occurrence time*)
- Event Stream
- Event Query



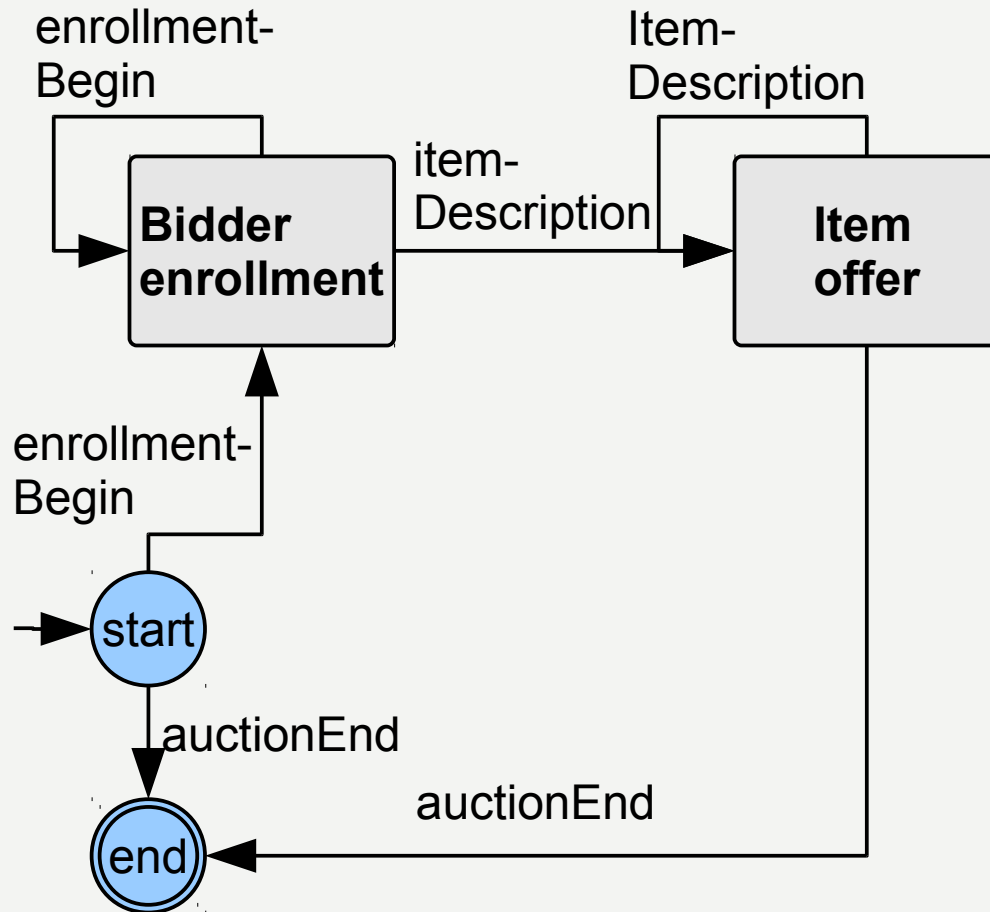
Semantic Query Optimization (SQO)

- Future contents of event streams often predetermined by **application logic** and **physical laws**
- Example: An Auction
 - Bidder Registration: Within the first 20 minutes
 - Item Offer:
 - Bids with an increasing price
 - Hammer beats after 30 seconds without any bid
 - Item sell after three consecutive hammer beats
- Use semantics for a more efficient event query evaluation:
 - bid event queries return no answer during the first 20 minutes of an auction
 - A query for the number of item sells can return '1' when there was at least one Bid

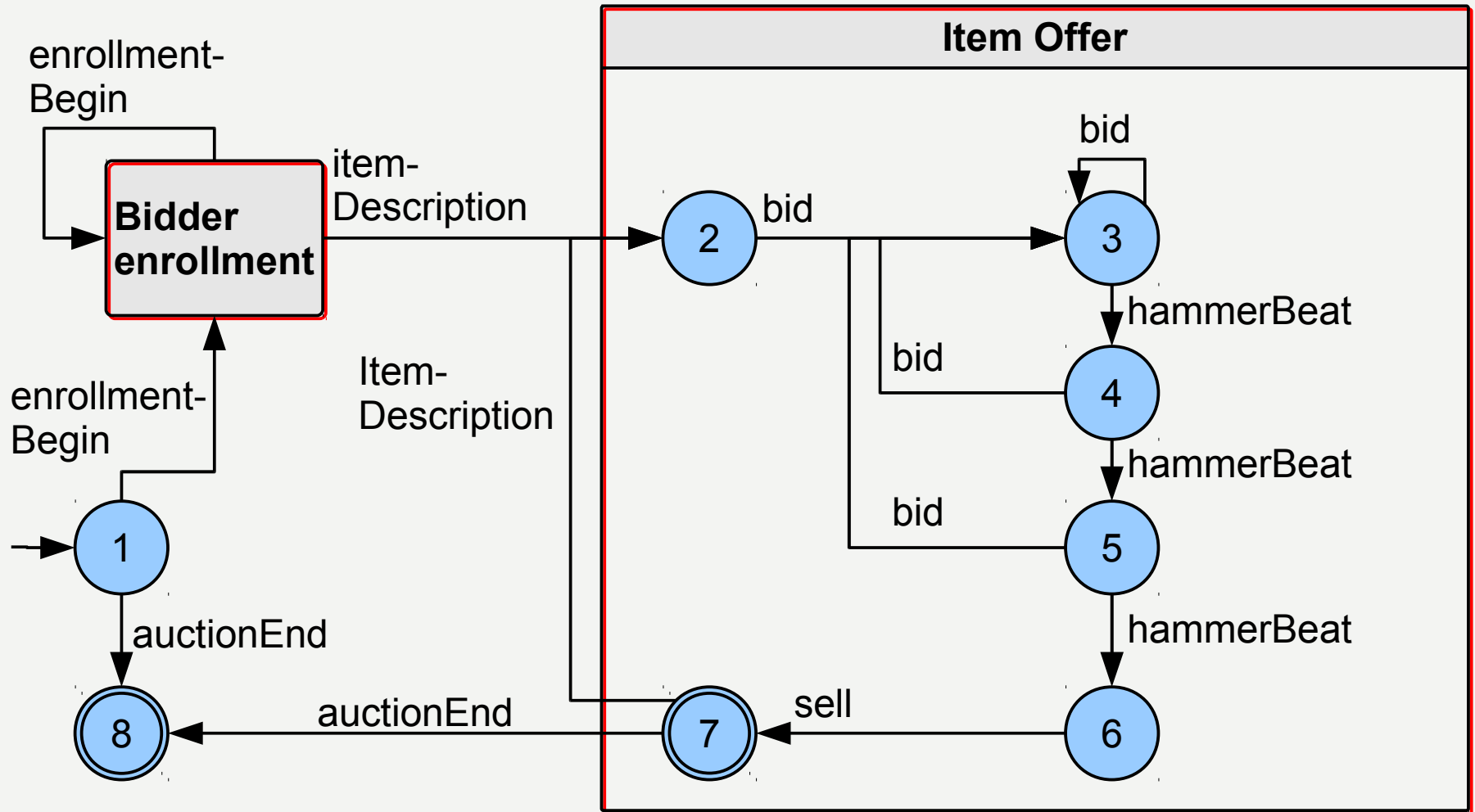
Subject of the Work

- SQO requires a formal representation of application semantics
- **Instantiating Hierarchical Timed Automata (IHTA)**

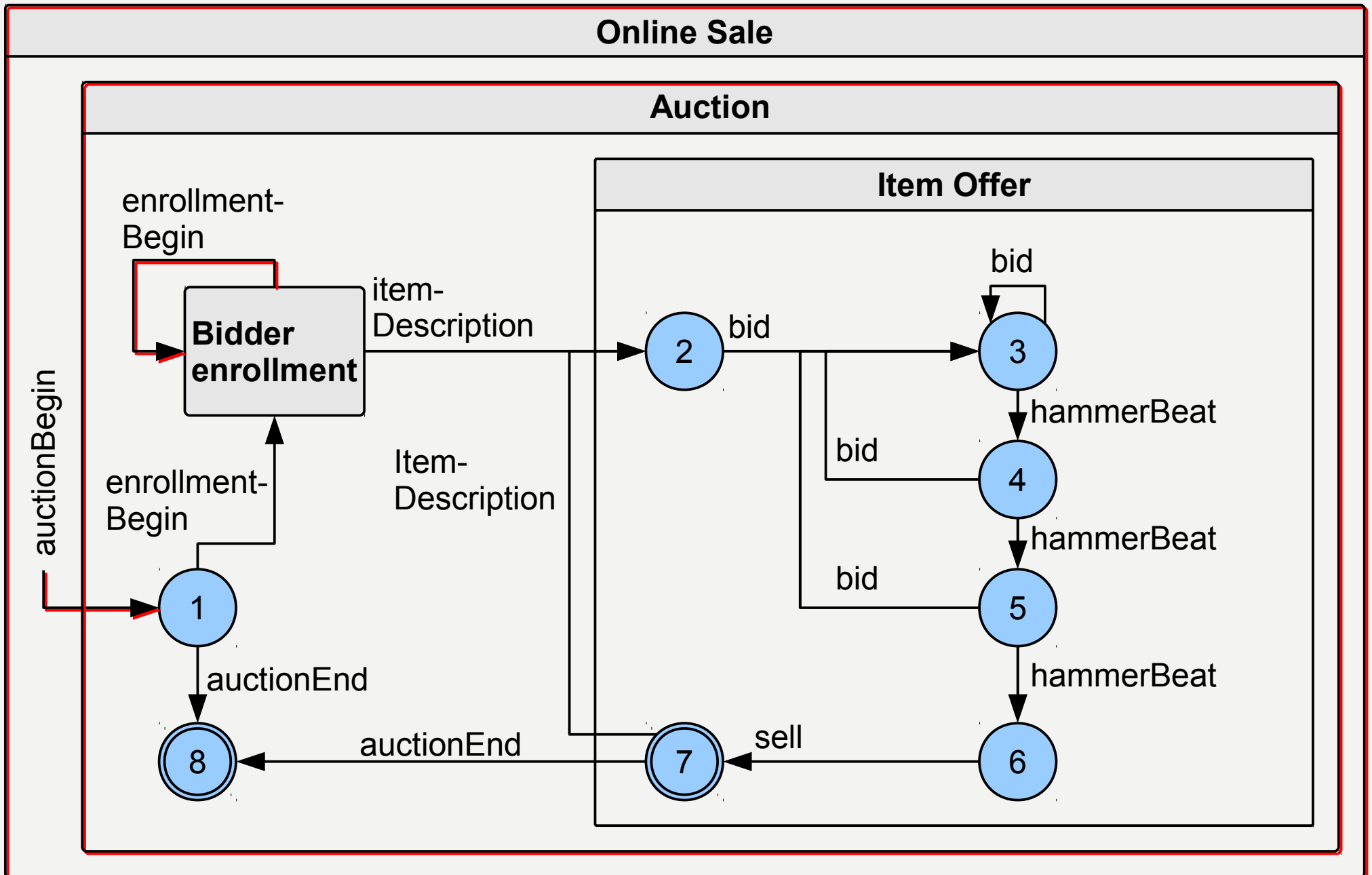
Requirements: State



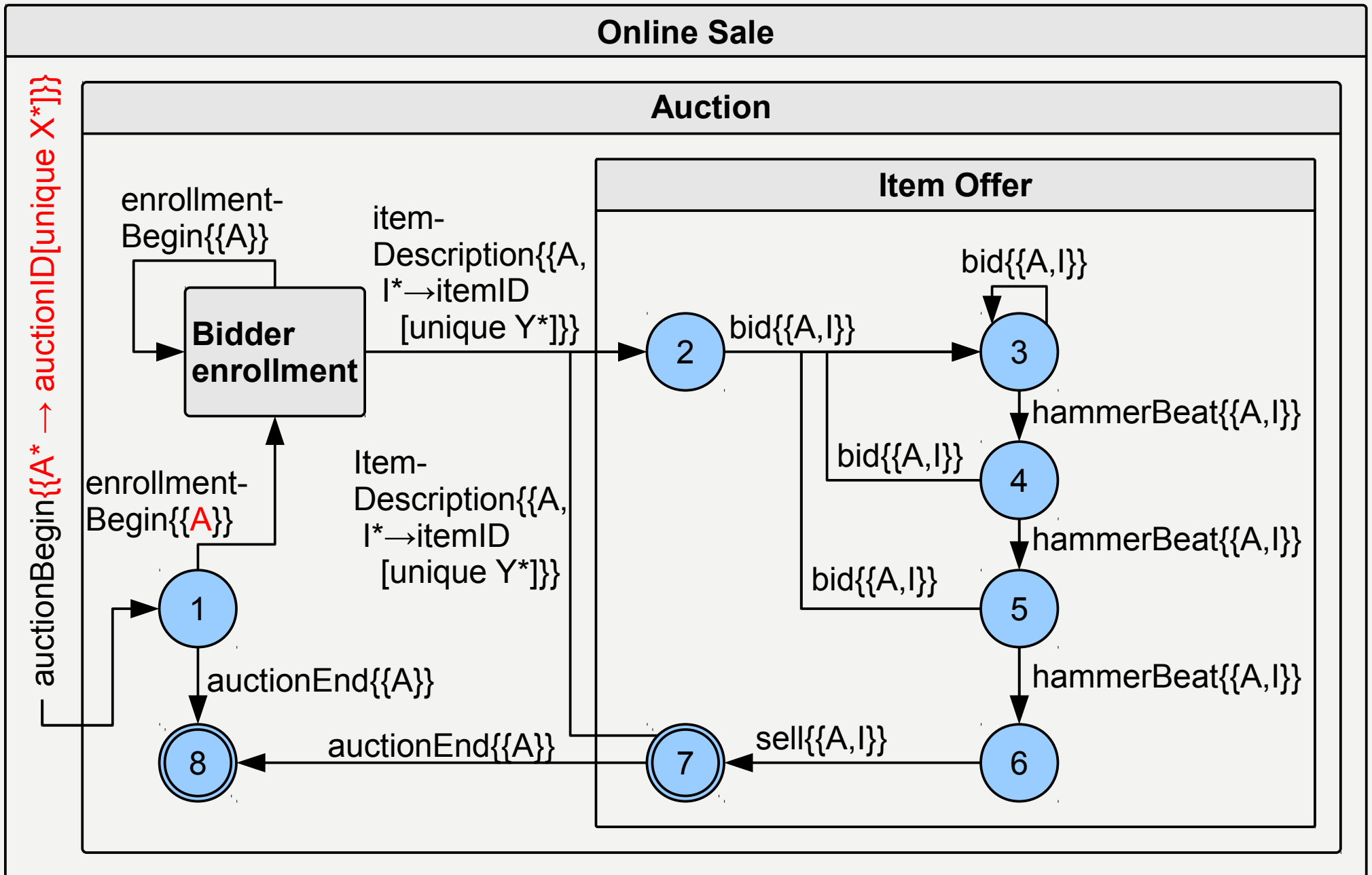
Requirements: Hierarchy



Requirements: Instantiation



Requirements: Event Data



Related Work

- Timed Finite Automata (TFA)

Model-checking and solving of scheduling problems

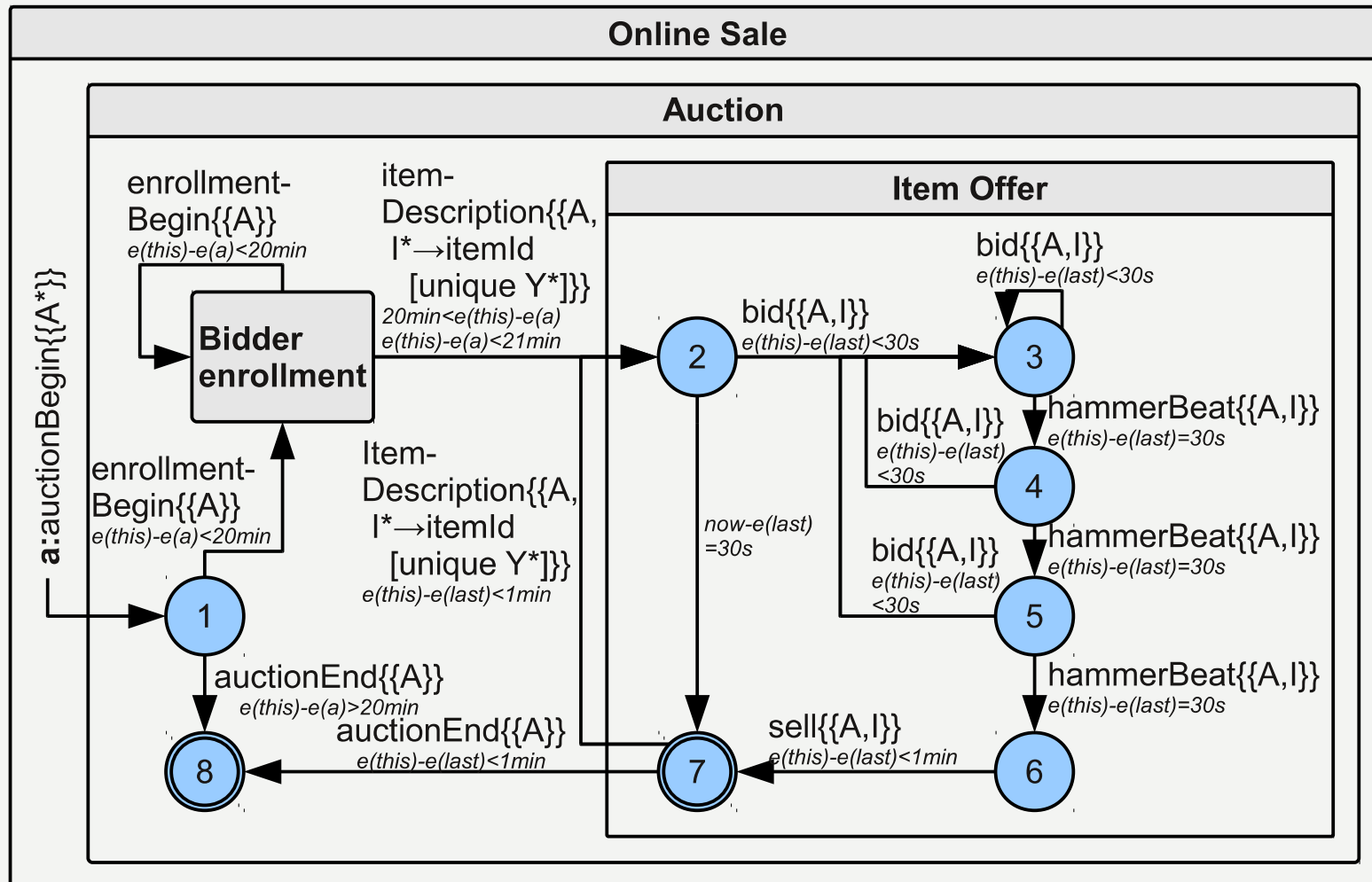
- Hierarchical Timed Automata (HTA)

Model-checking of complex real-time systems

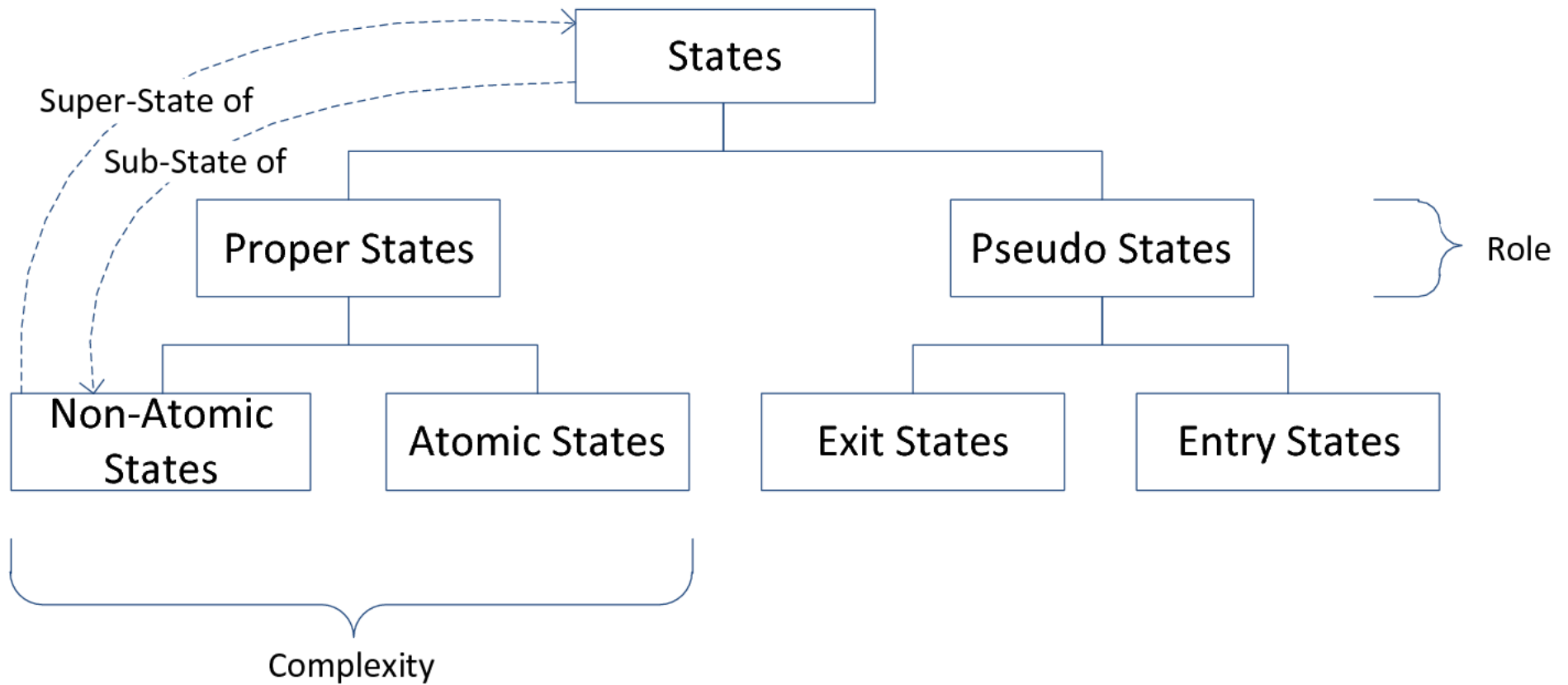
	TFA	HTA	IHTA
Event Data	No	No	Yes
Event Occurrence Time	Time Point	Time Point	Time Interval
Temporal Constraints on ...	Clocks	Clocks	Event Occurrence Times
Order of Events in a Stream	Strict ($<$)	Strict ($<$)	Partial (\leq)
Non-Determinism Semantics	Yes	Yes	Yes, Operational
Hierarchy of states	No	Yes	Yes
Number of concurrent processes	Bounded	Bounded	Unbounded

IHTA Syntax

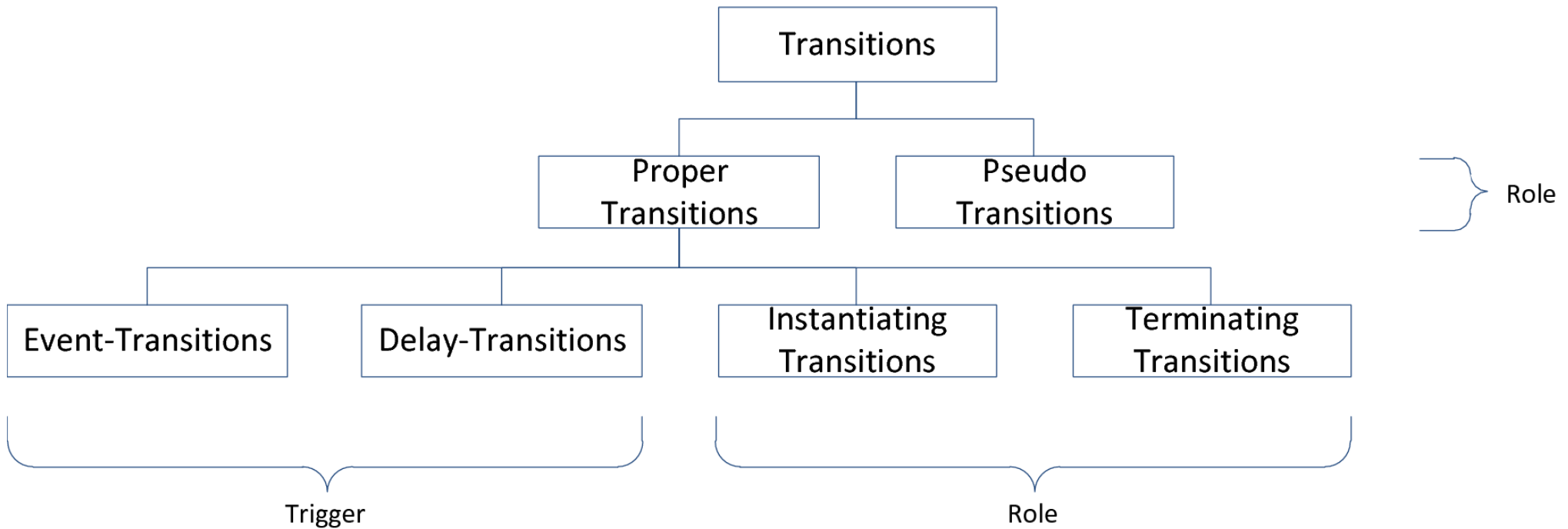
- States, Transitions
- Event Query Terms, Temporal Constraints



Classification of States of IHTA



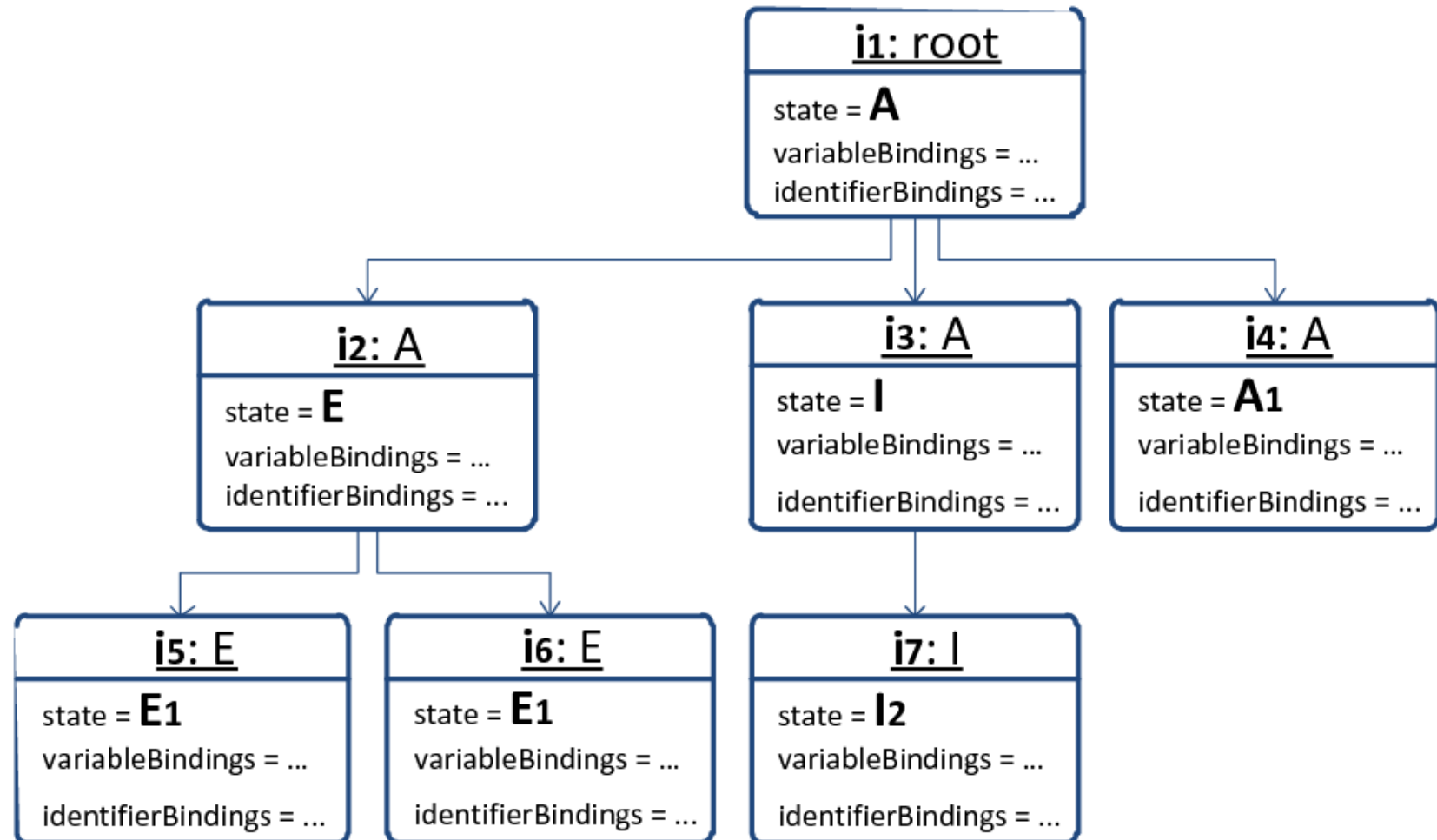
Classification of Transitions of IHTA



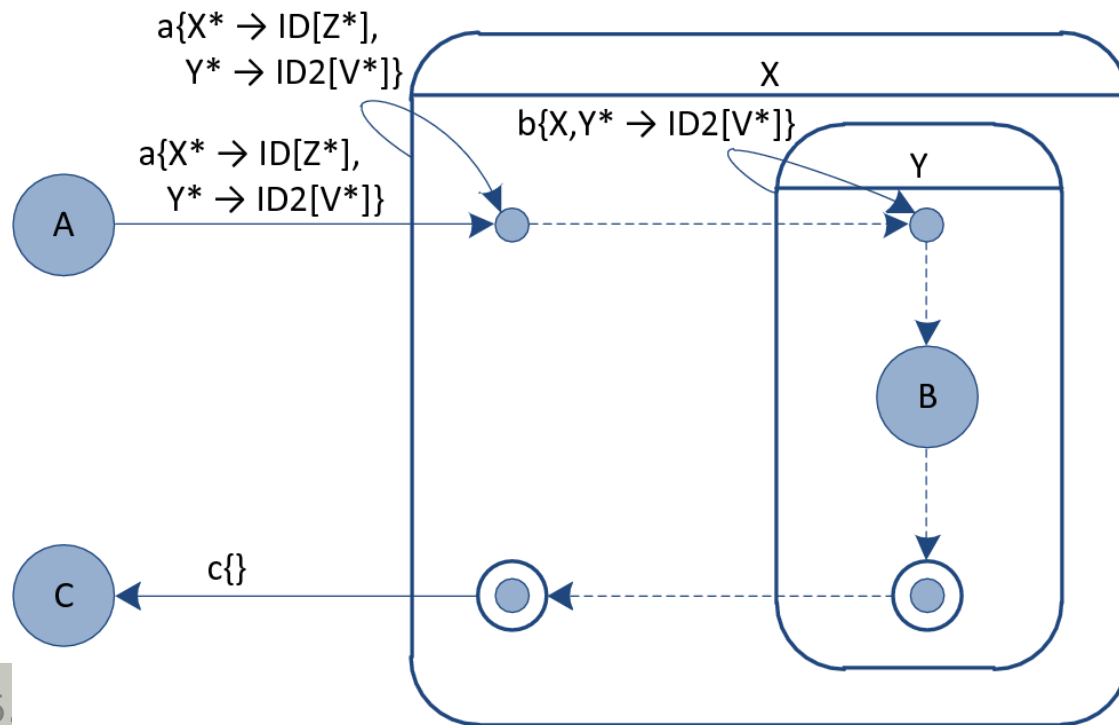
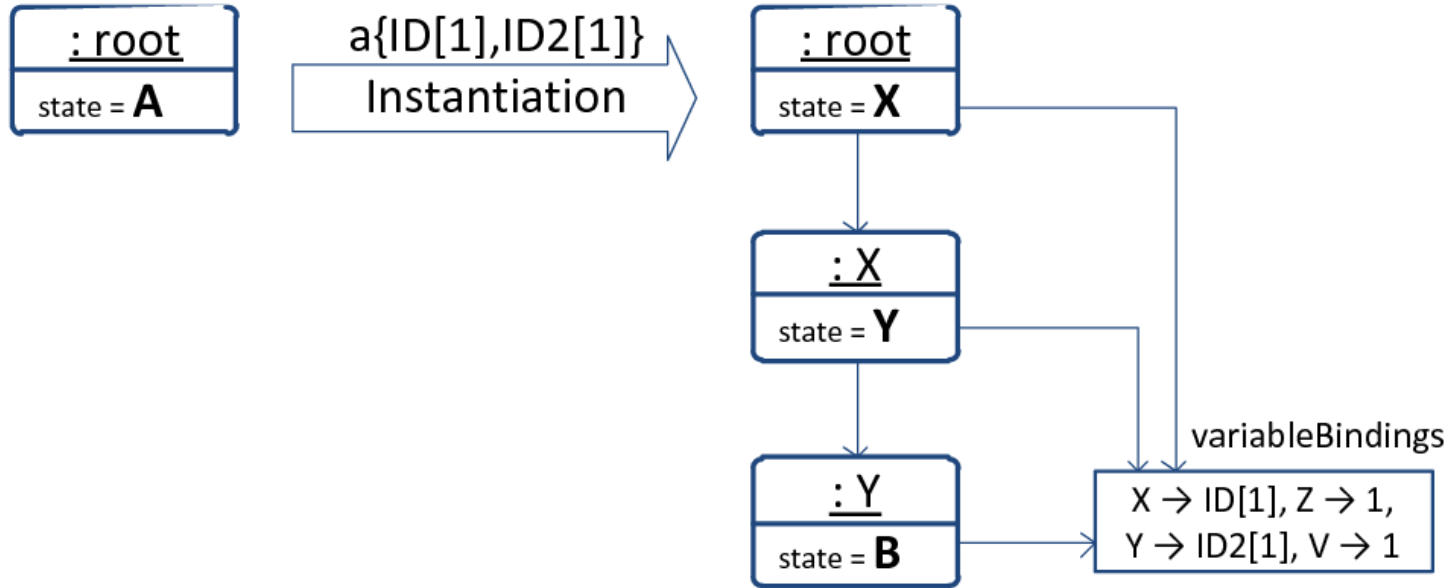
IHTA Configuration

- Dynamical tree of instances:

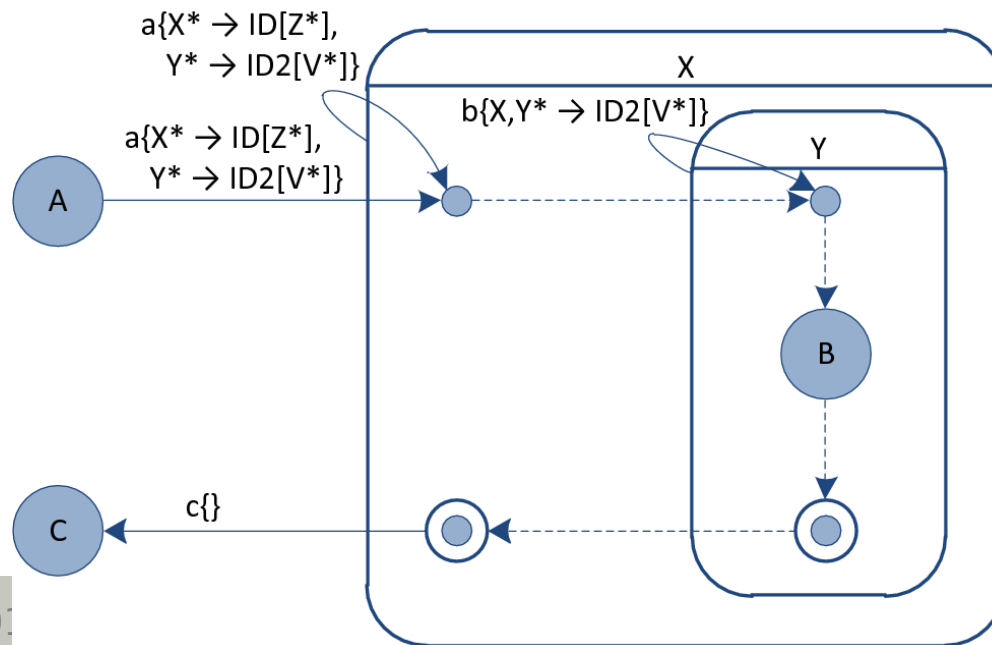
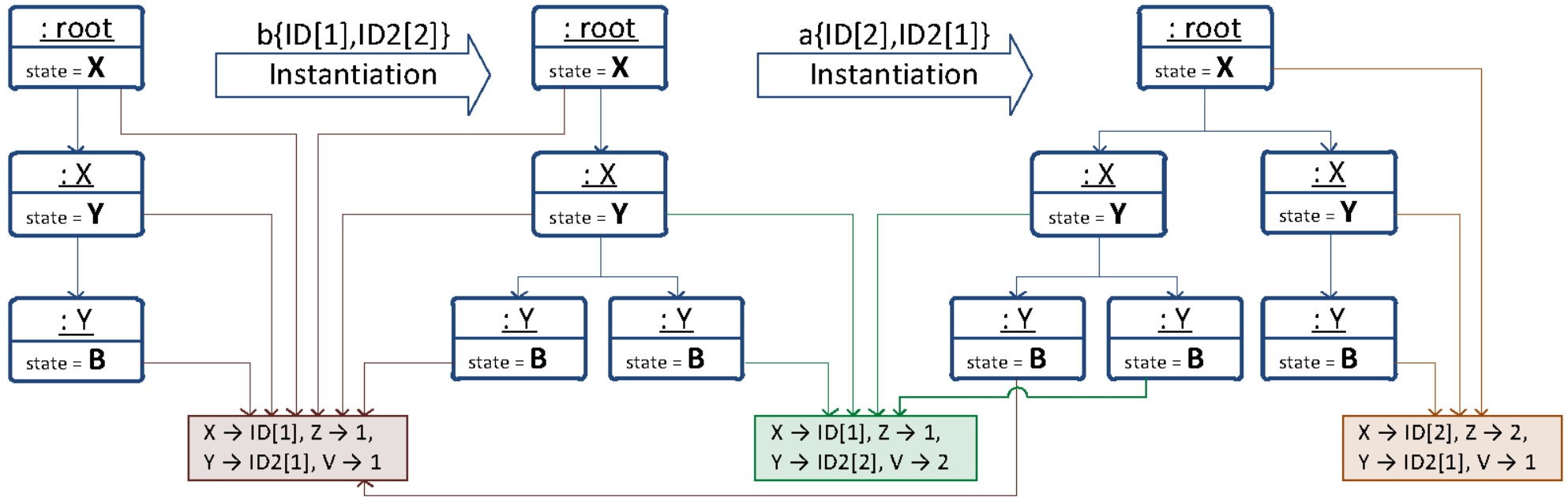
Representation of an automaton during its run



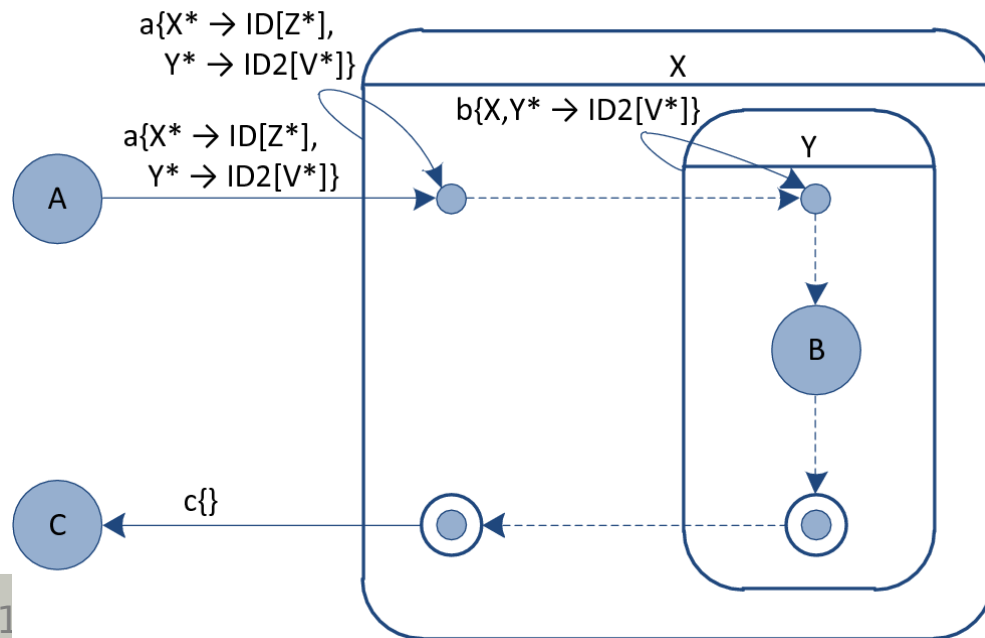
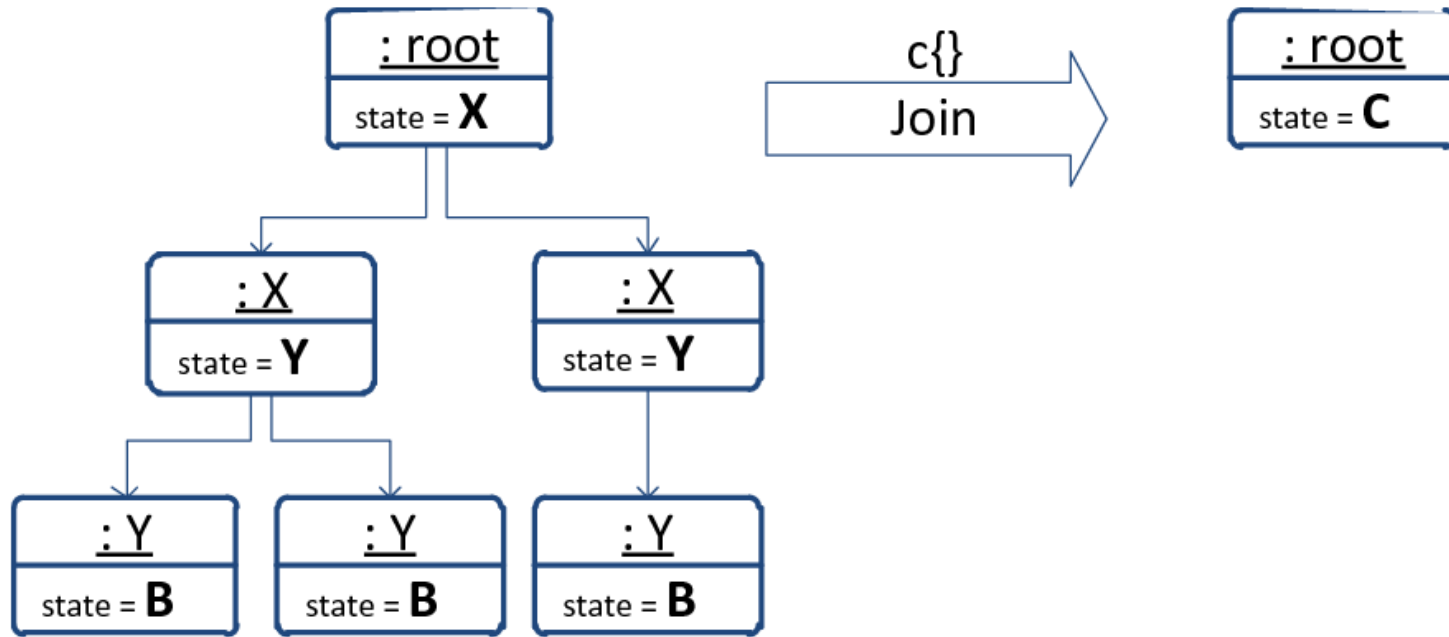
IHTA Configuration: Instantiation



IHTA Configuration: Concurrency

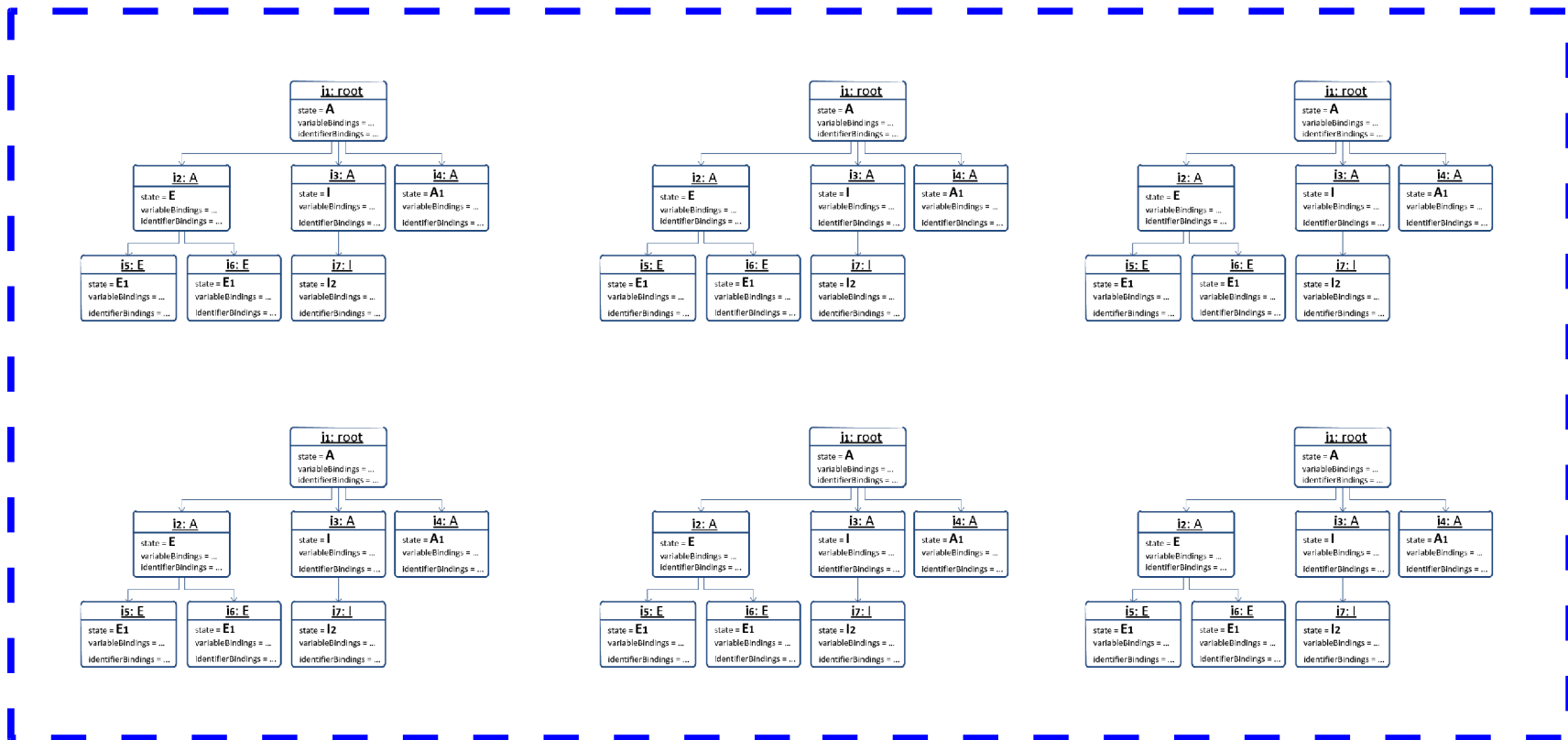


IHTA Configuration: Termination

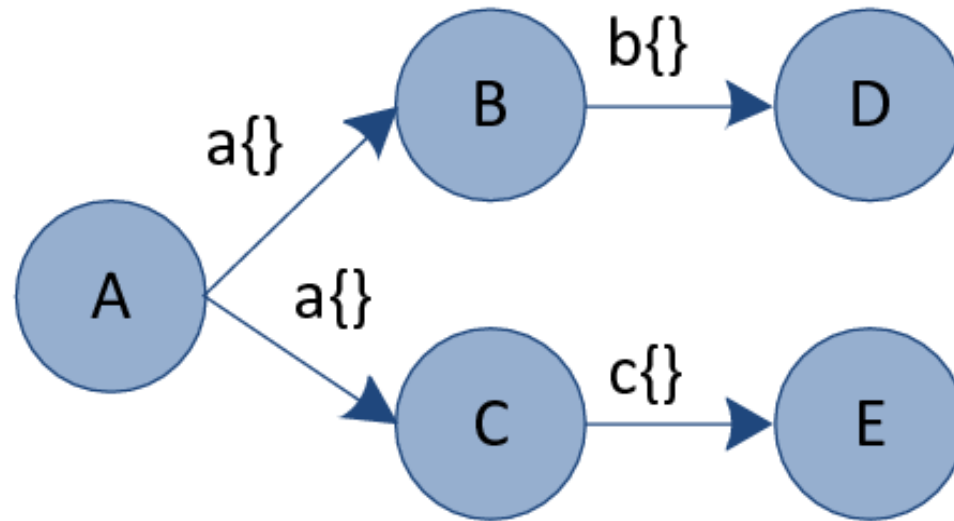
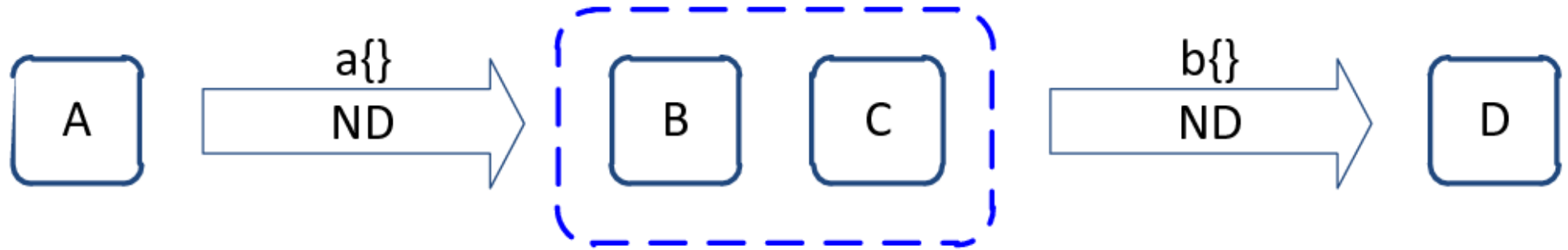


IHTA Configuration: Nondeterminism

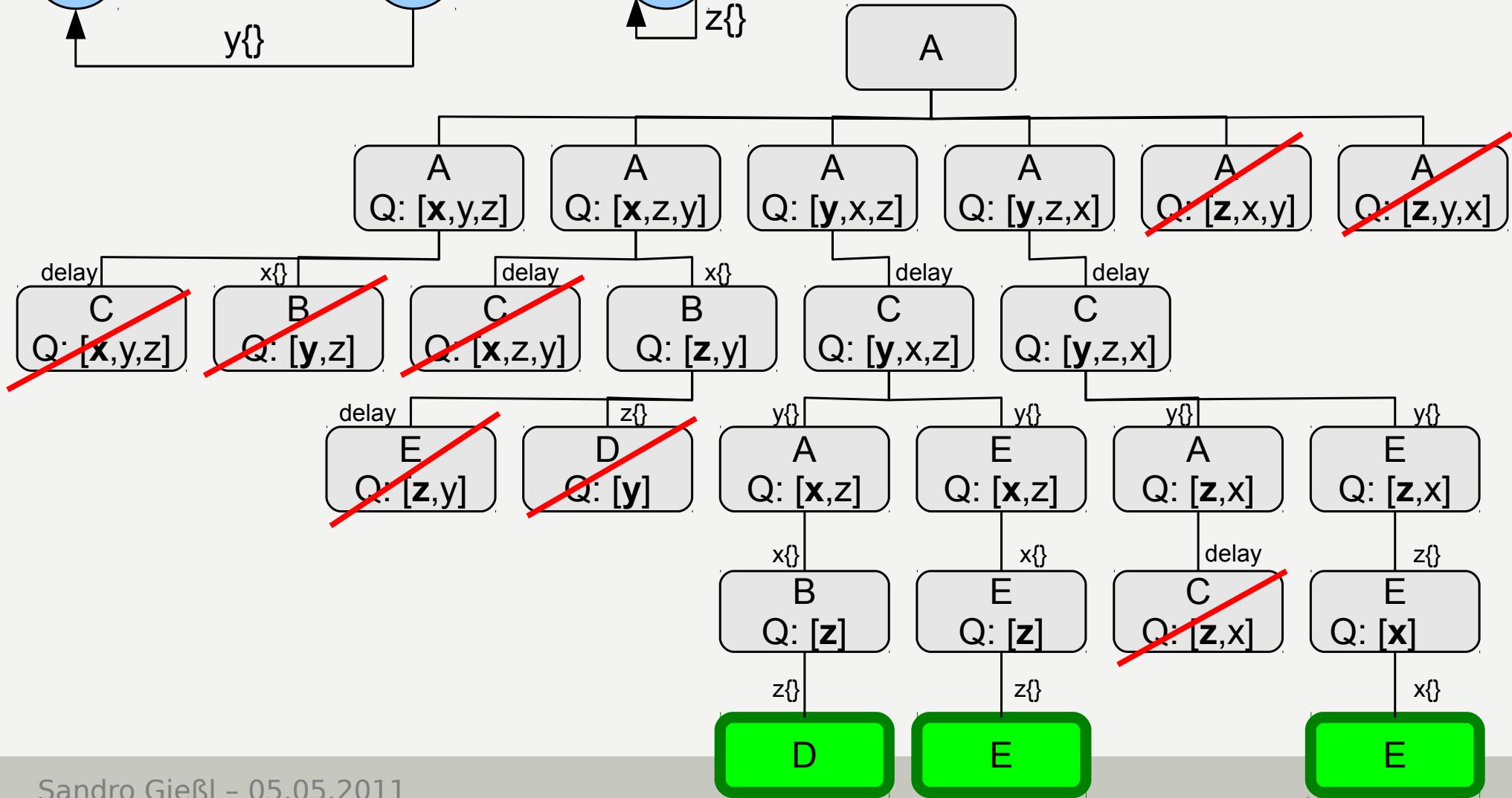
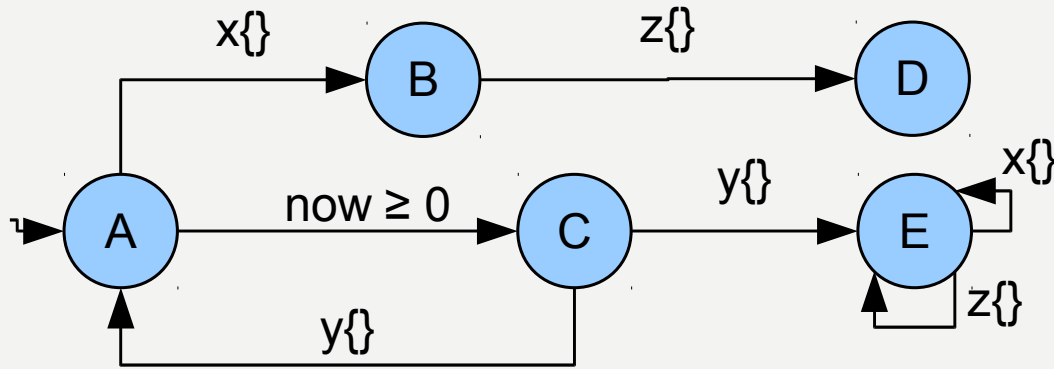
- The configuration can represent multiple non-deterministic runs



IHTA Configuration: Nondeterminism



Partial-Order Event Stream $y\{^{[1,1]}, x\{^{[1,1]}, z\{^{[1,1]}$



Conclusion

- Formal semantics of IHTA has been defined
- Each iteration of the algorithm at time point 'now' terminates

- Complexity of each iteration:

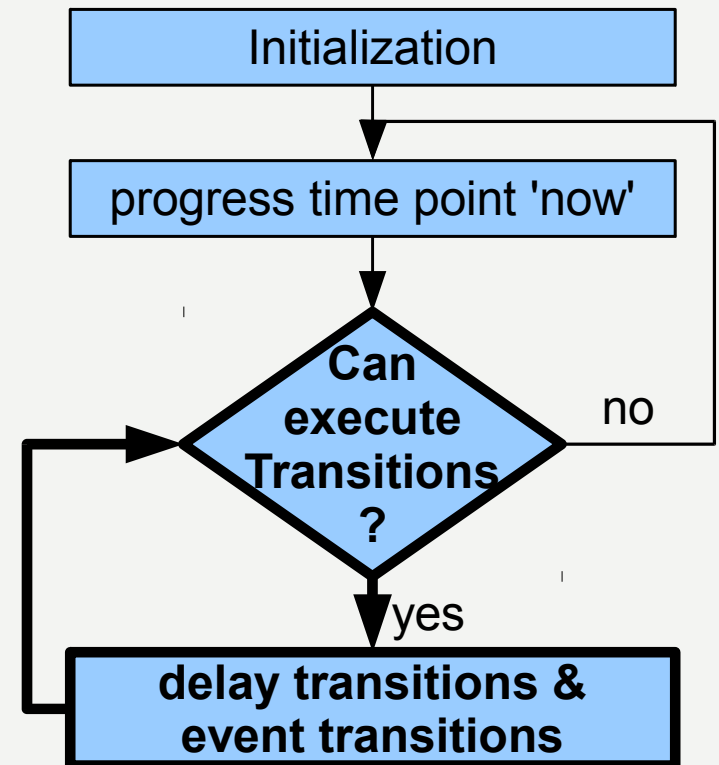
$$\in O(n_r * n_e! * n_i^2 * n_t^4 * (n_e + n_t))$$

n_r : Number of non-deterministic runs

n_e : Number of events with equal occurrence times

n_i : Max. number of instances in a run

n_t : Number of transitions (constant)



Conclusion

- Formal semantics of IHTA, coping with
 - Arbitrary number of processes
 - Non-determinism
 - Partially ordered events in a streams
- Kind of knowledge captured by IHTA:
 - Cardinality of events and states
 - Functional dependency between events and states
- IHTA are concise, modular and therefore readable and (ex-)changeable

Future Work

- Deriving states from IHTA at run-time
- Adapting SQO methods of database systems to CEP
 - Constraint language suitable for CEP
 - Deriving constraints from IHTA
- A visual editor for IHTA
- Additional use cases
- Events matched by many instances
- Extended event queries in transitions
- Improved efficiency of operational semantics of IHTA