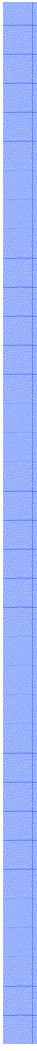


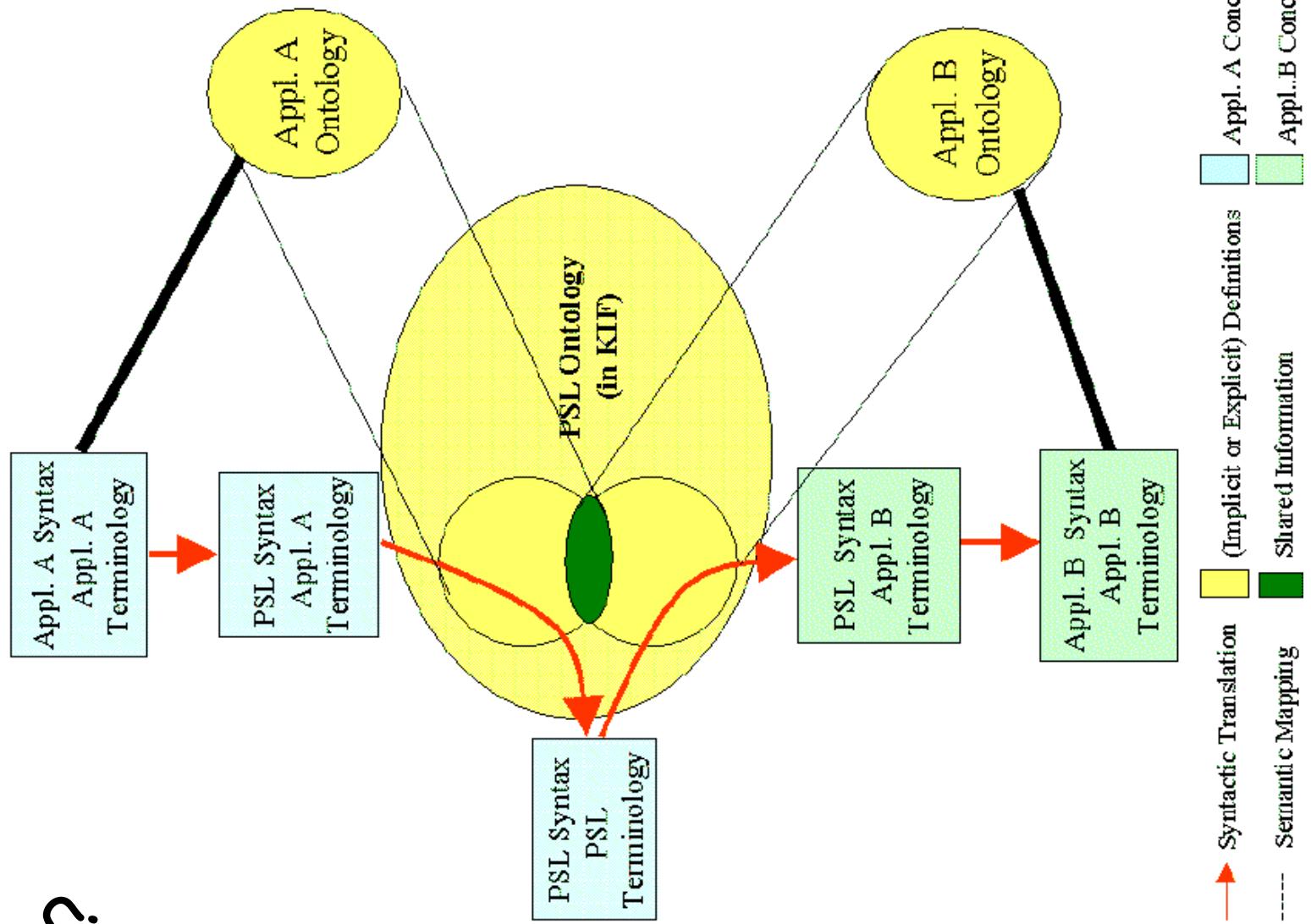
Process Specification Language (PSL)



PSL: Process Specification Language

- A standard developed by NIST
- Representing manufacturing processes
 - Process as data
 - Semantics
- Aims at interoperability
 - n^2 translators $\rightarrow 2n$ translators

How Does PSL Work?



Process Ontology

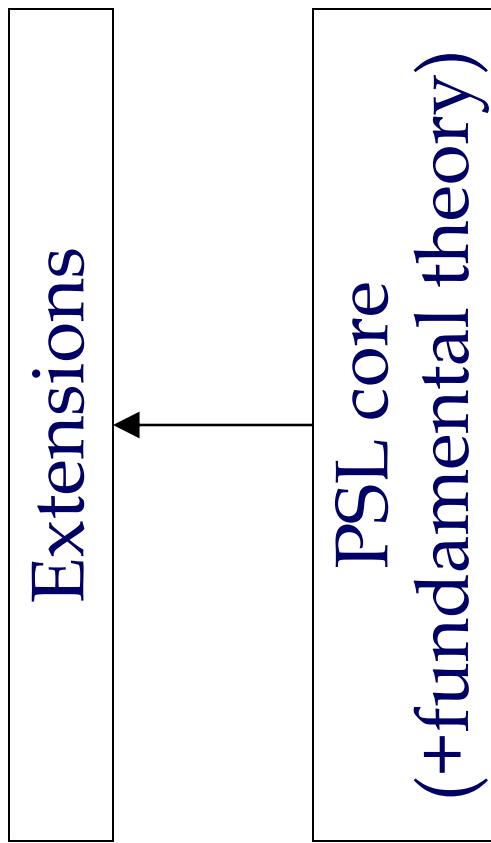
- An ontology is a set of logic sentences:
 - Fundamental theories
 - Definitions over the theories

- A process ontology needs:
 - Language – syntax
 - Model theory – meaning of the syntax
 - Proof theory – axioms to tailor to the specifics of processes

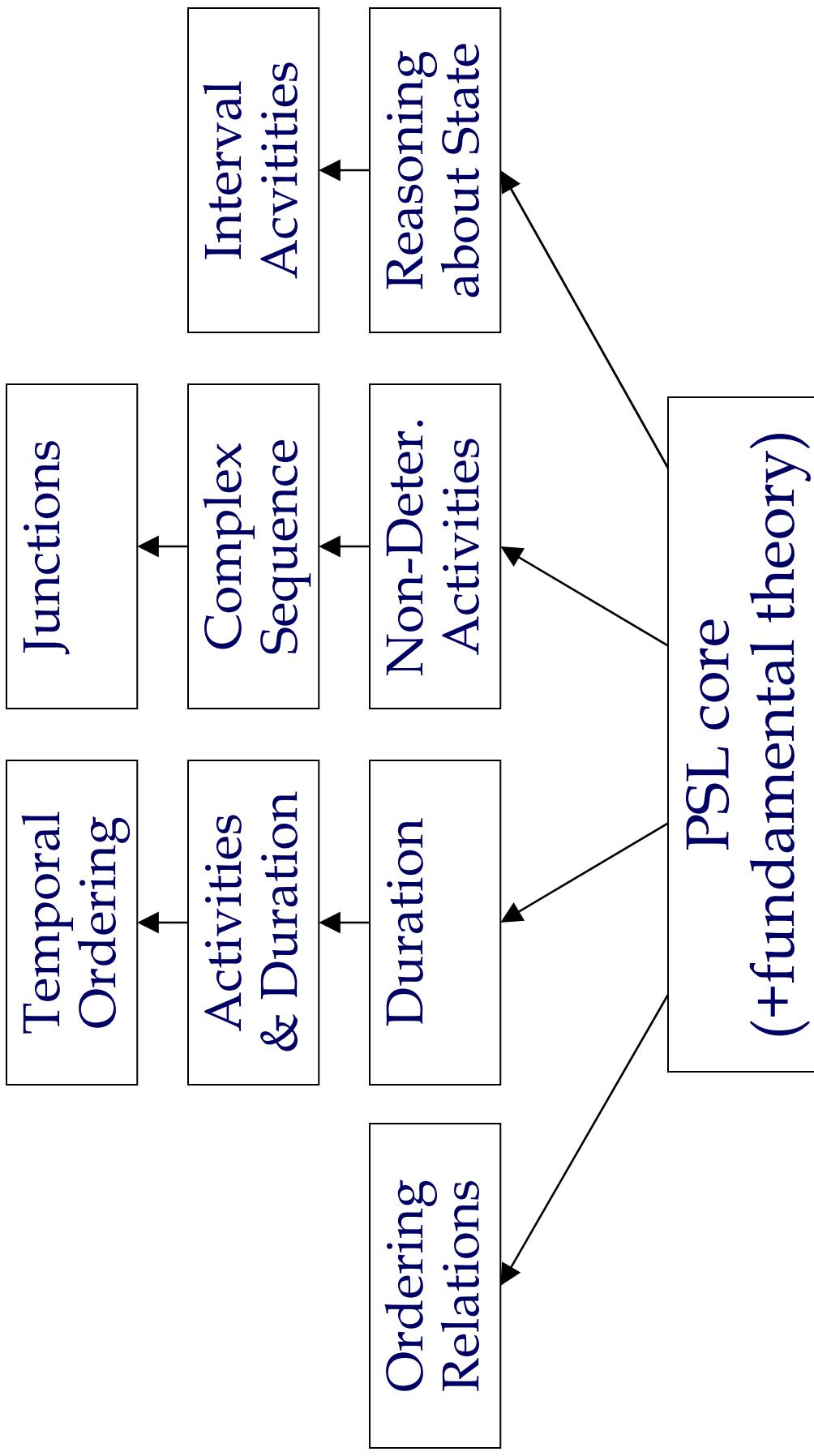
PSL Ontology

- PSL core: intuitive semantic primitives that is adequate for describing the fundamental concepts of manufacturing processes
- Three families of extensions: outer core, generic activities, and schedules
- Outer core: still very general
 - Subactivity, activity-occurrence, state
- Generic activities: process modeling and ordering
- Schedules: motivated from a pilot implementation

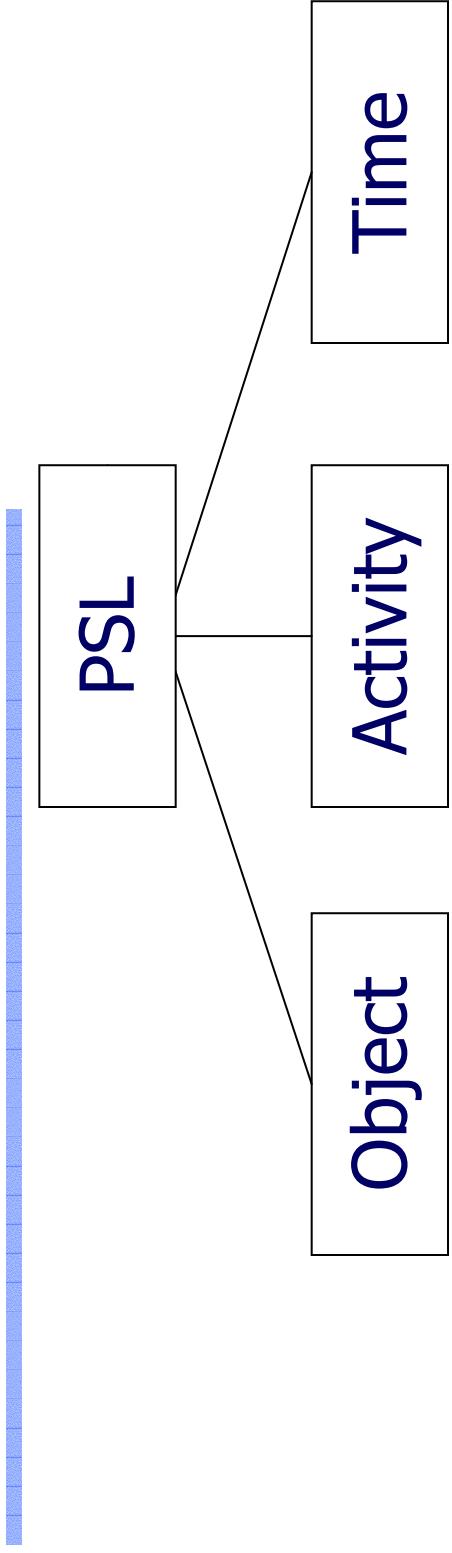
PSL Semantic Architecture



Models for Generic Activities and Ordering



Elements of PSL



- A process is one or more **activities** that occur over a a period of time in which **objects** participate
- Four (disjoint) classes/concepts:
 - **Activity** — a type of action
 - **Activity-Occurrence** — an event or action that takes place at a specific place and time
 - **Timepoint** — a time instant
 - **Object** — anything but not a timepoint nor an activity

PSL Core

- There are four kinds of entities required for reasoning about processes – **activities, activity occurrences, timepoints, and objects**
- Activities may have multiple occurrences, or there may exist activities that do not occur at all
- Timepoints are linearly ordered, forwards into the future, and backwards into the past
- Activity occurrences and objects are associated with unique timepoints that mark the begin and end of the occurrence or object

PSL Formalism

- Based on situation calculus
 - first-order logic plus time
- Provides a way to represent process information, i.e., processes as data
 - Similar to “process tables” and “context switch” in OS
- Enables analysis and optimization of web service and/or executions

Primitives in PSL Core: Classes

- Four classes (sets):

- *Activity*: reusable behaviors (e.g., programs)
 - *Activity_occurrence*: specific instances of activities, uniquely associated with activities
 - *Timepoint*: time instants for objects and activity occurrences
 - *Object*: anything not activities, activity occurrences, nor timepoints
- Treated as unary relations
 - $\text{Activity}(x)$ means the statement " x is an activity"
 - Denoted as O , A , AO , and T , respectively

Primitives in PSL Core: Relations

■ Three relations (predicates):

- $\text{Participates_In} \subseteq O \times A \times T$
 $\text{Participates_In}(x, y, z) : x$ plays some role in an
occurrence of the activity y at the timepoint z
- $\text{Before} \subseteq T \times T$
 $\text{Before}(x, y) : \text{the timepoint } x \text{ is earlier than } y \text{ in the}$
linear ordering over timepoints
- $\text{Occurrence_Of} \subseteq Ao \times A$
 $\text{Occurrence_Of}(x, y) : x$ is a particular occurrence of
the activity y

Primitives in PSL Core: Functions

- Two unary functions:

- **$Begin_Of : O \cup Ao \rightarrow T$**
 $Begin_Of(x)$: returns the starting time of an object x
or an activity occurrence x
- **$End_Of : O \cup Ao \rightarrow T$**
 $End_Of(x)$: returns the end time of an object x or an activity occurrence x

Primitives in PSL Core: Timepoint Constants

- Two constants:
 - `inf+`: the timepoint that is after all timepoints
 - `inf-`: the timepoint that is before all timepoints

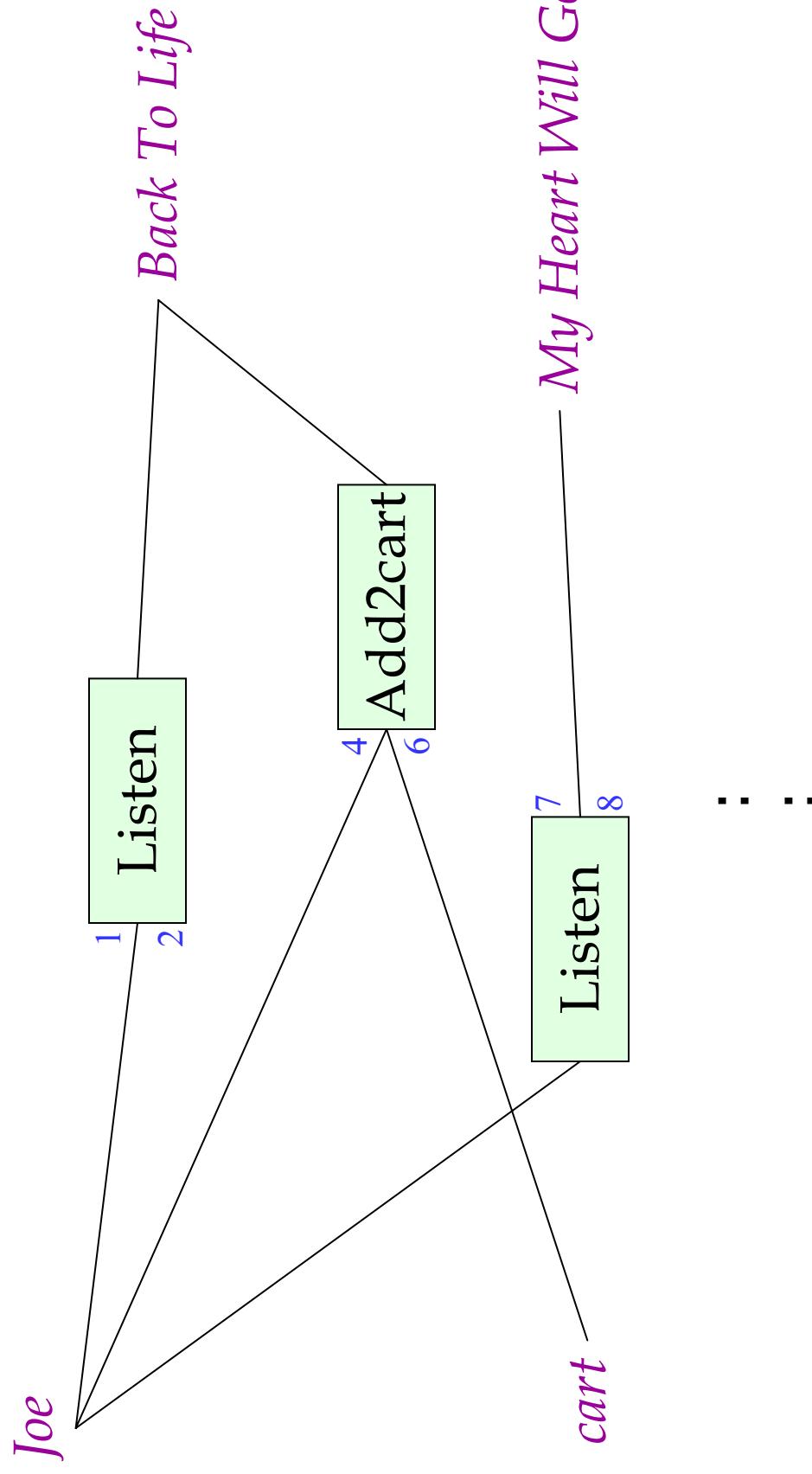
Primitives in PSL Core: The Language

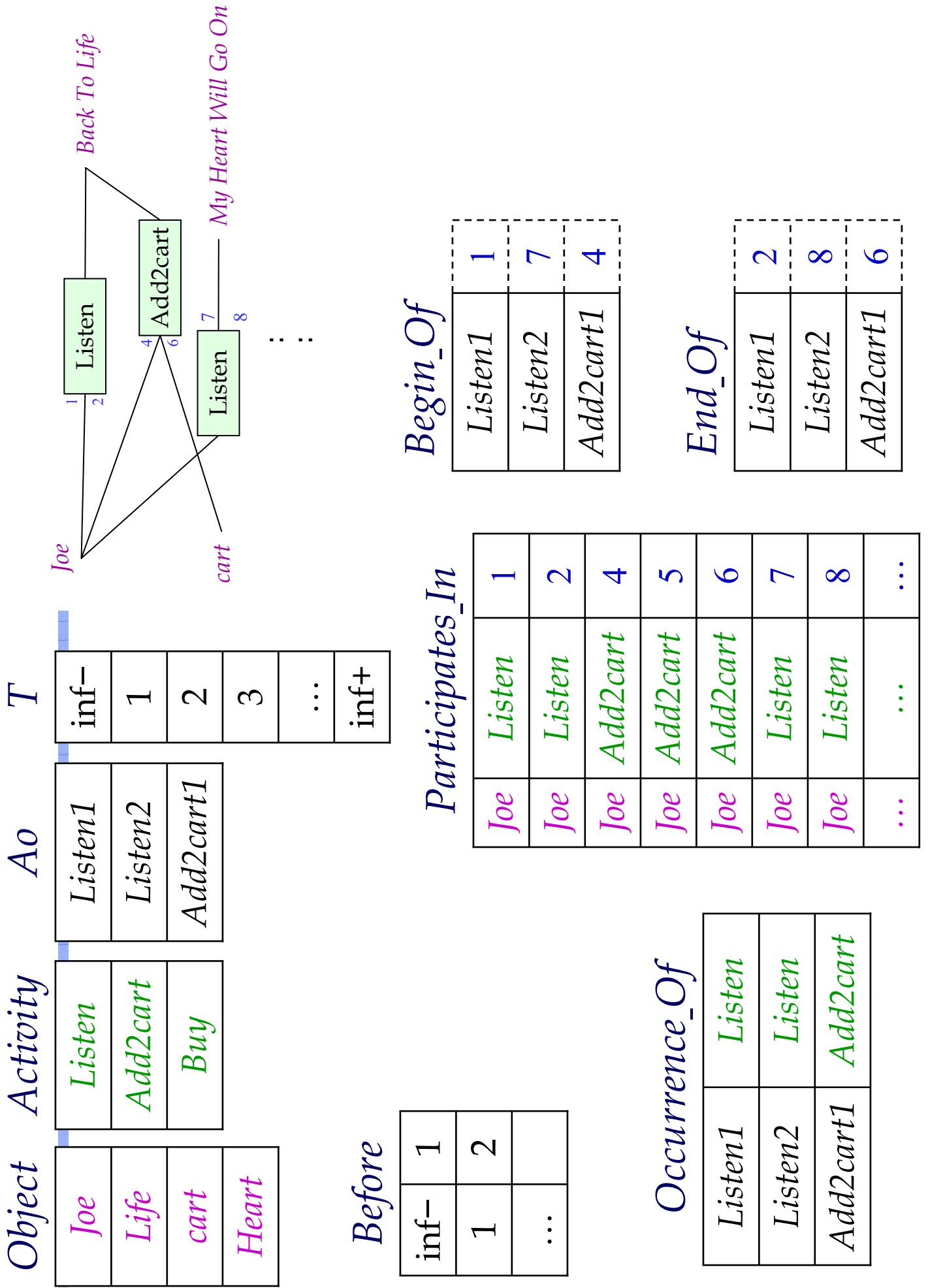
■ Relations

- **Unary** : $A(\cdot)$, $Ao(\cdot)$, $T(\cdot)$, $O(\cdot)$
- **Binary** : $Before(\cdot, \cdot)$, $Occurrence_Of(\cdot, \cdot)$
- **Ternary** : $Participates_In(\cdot, \cdot, \cdot)$
- **Functions** : $Begin_Of(\cdot)$, $End_Of(\cdot)$
- **Constants** : inf+ , inf-

An Example

- PSL can be used to represent what happened and what is happening during service execution





Object *Activity* *Ao* *Occurrence_Of*

<i>Joe</i>	<i>Listen</i>	<i>Listen1</i>
<i>Life</i>	<i>Add2cart</i>	<i>Listen2</i>
<i>cart</i>	<i>Buy</i>	<i>Add2cart1</i>
<i>Heart</i>		

<i>Joe</i>	<i>Listen1</i>	<i>Listen</i>
<i>Life</i>	<i>Listen2</i>	<i>Listen</i>
<i>cart</i>	<i>Add2cart1</i>	<i>Add2cart</i>

<i>T</i>	<i>Before</i>	<i>inf-</i>	<i>inf-</i>	<i>inf-</i>
		1	1	1
		2	2	2
	
		3	3	3
				<i>inf+</i>

Begin_Of

<i>Listen1</i>	<i>1</i>
<i>Listen2</i>	<i>7</i>
<i>Add2cart1</i>	<i>4</i>

```
SELECT P.1
      FROM Participates_In P
     WHERE P.3=4
```

End_Of

<i>Listen1</i>	<i>2</i>
<i>Listen2</i>	<i>8</i>
<i>Add2cart1</i>	<i>6</i>

<i>Joe</i>	<i>Listen</i>	<i>1</i>
<i>Joe</i>	<i>Listen</i>	<i>2</i>
<i>Joe</i>	<i>Add2cart</i>	<i>4</i>
<i>Joe</i>	<i>Add2cart</i>	<i>5</i>
<i>Joe</i>	<i>Add2cart</i>	<i>6</i>
<i>Joe</i>	<i>Listen</i>	<i>7</i>
<i>Joe</i>	<i>Listen</i>	<i>8</i>
<i>...</i>	<i>...</i>	<i>...</i>

Relations and Functions in PSL

Before

inf-	1
1	2
1	4
4	1
...	

- What if *Before* has a cycle?
- What if *Before* does not have (2,5)?
- What if the begin time is later than the end time?
- What if some values occur in both *Activity* and *Activity_Occurrence* ?
- ...
- Solution: define axioms that are conditions to be always satisfied by the relations and functions
 - The conditions can be defined in the Logic

Begin_Of

Listen1	5
Listen2	7
Add2cart1	4

End_Of

Listen1	2
Listen2	8
Add2cart1	6

Axioms for PSL Core

Axiom 1: The *Before* relation only holds between timepoints

Axiom 2: The *Before* relation is a total ordering

Axiom 3: The *Before* relation is irreflexive

Axiom 4: The *Before* relation is transitive

Axiom 5: The timepoint inf- is before all other timepoints

Axiom 6: Every other timepoint is before inf+

Axiom 7: Given any timepoint t other than inf- , there is a timepoint between inf- and t

Axiom 8: Given any timepoint t other than inf+ , there is a timepoint between t and inf+

Axioms for PSL Core

Axiom 9: Everything is either an activity, activity occurrence, timepoint, or object

Axiom 10: Objects, activities, activity occurrences, and timepoints are all distinct kinds of things

Axiom 11: The occurrence relation only holds between activities and activity occurrences

Axiom 12: Every activity occurrence is the occurrence of some activity

Axiom 13: An activity occurrence is associated with a unique activity

Axioms for PSL Core

Axiom 14: The begin and end of an activity occurrence or object are timepoints

Axiom 15: The begin point of every activity occurrence or object is before or equal to its end point

Axiom 16: The *participates_in* relation only holds between objects, activities, and timepoints, respectively

Axiom 17: An object can participate in an activity only at those timepoints at which both the object exists and the activity is occurring

PSL Core: Universe

- (Axiom 9) Everything is either an activity, activity occurrence, timepoint, or object

$$\forall x (A(x) \vee Ao(x) \vee T(x) \vee O(x))$$

- (Axiom 10) Objects, activities, activity occurrences, and timepoints are all distinct kinds of things

$$\forall x ((A(x) \rightarrow \neg(Ao(x) \vee T(x) \vee O(x))) \wedge \\ (Ao(x) \rightarrow \neg(T(x) \vee O(x))) \wedge \\ (T(x) \rightarrow \neg O(x)))$$

PSL Core: Activities

- (Axiom 12) Every activity occurrence is the occurrence of some activity

$$\forall x \ (Ao(x) \rightarrow \exists y \ A(y) \wedge Occurrence_Of(x, y))$$

- (Axiom 14) The begin and end of an activity occurrence or object are timepoints

$$\forall x \forall y \ (Occurrence_Of(x, y) \vee O(x) \rightarrow \\ T(Begin_Of(x)) \wedge T(End_Of(x)))$$

- (Axiom 13) An activity occurrence is associated with a unique activity

$$\forall x \forall y \forall z \ (Occurrence_Of(x, y) \wedge Occurrence_Of(x, z) \\ \rightarrow y = z)$$

PSL Core: Time Instant

- (Axiom 5) The timepoint inf- is before all other timepoints

$$\forall x (T(x) \wedge \neg x = \text{inf-} \rightarrow \text{Before}(\text{inf-}, x))$$

- (Axiom 6) Every other timepoint is before inf+

$$\forall x (T(x) \wedge \neg x = \text{inf+} \rightarrow \text{Before}(x, \text{inf+}))$$

- (Axiom 7) Given any timepoint t other than inf- , there is a timepoint between inf- and t

$$\forall x (T(x) \wedge \neg x = \text{inf-} \rightarrow \exists y \text{Between}(\text{inf-}, y, x))$$

$$\text{Between}(x, y, z) \equiv \text{Before}(x, y) \wedge \text{Before}(y, z)$$

- (Axiom 8) Given any timepoint t other than inf+ , there is a timepoint between t and inf+

$$\forall x (T(x) \wedge \neg x = \text{inf+} \rightarrow \exists y \text{Between}(x, y, \text{inf+}))$$

PSL Core: Relation Before

- (Axiom 1) The *Before* relation only holds between timepoints
- (Axiom 2) The *Before* relation is a total ordering
- (Axiom 3) The *Before* relation is irreflexive
- (Axiom 4) The *Before* relation is transitive

PSL Core: Relation Occurrence_Of

- (Axiom 11) The *Occurrence relation* only holds between activities and activity occurrences
- (Axiom 12) Every activity occurrence is the occurrence of some activity
- (Axiom 17) An object can participate in an activity only at those timepoints at which both the object exists and the activity is occurring

PSL Core: Relation *Participates_In*

- (Axiom 16) The *participates_in* relation only holds between objects, activities, and timepoints, respectively

- (Axiom 17) An object can participate in an activity only at those timepoints at which both the object exists and the activity is occurring

PSL Core: Functions *Begin_Of* and *End_Of*

- (Axiom 14) The begin and end of an activity occurrence or object are timepoints

$$\forall x \forall y (Occurrence_Of(x, y) \vee O(x) \rightarrow \\ T(Begin_Of(x)) \wedge T(End_Of(x)))$$

- (Axiom 15) The begin point of every activity occurrence or object is before or equal to its end point

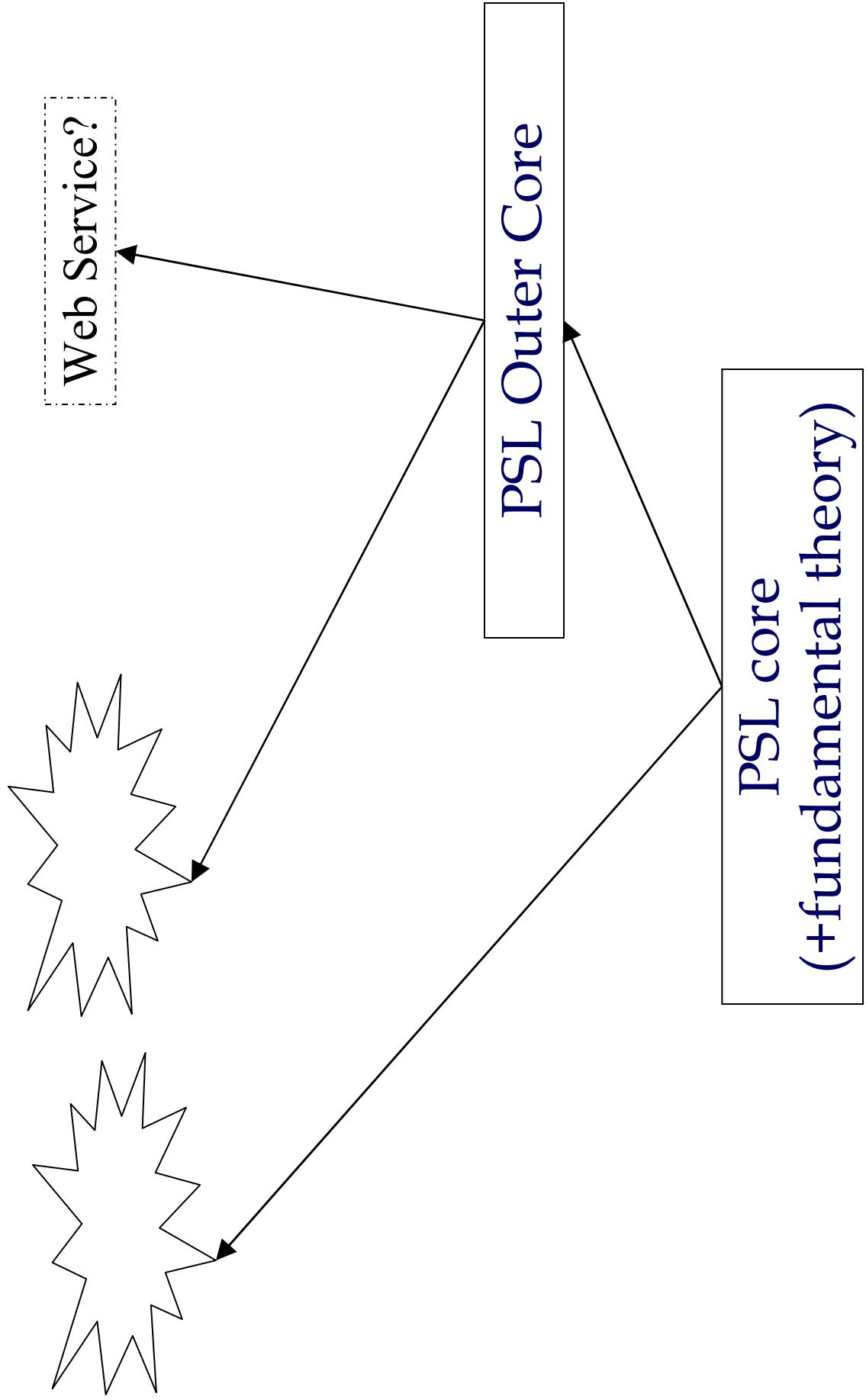
PSL Core: Supporting Relations

- $\text{Between}(x, y, z) \equiv \text{Before}(x, y) \wedge \text{Before}(y, z)$
- $\text{BeforeEq}(x, y) \equiv \text{Before}(x, y) \vee x = y$
- $\text{BetweenEq}(x, y, z) \equiv \text{BeforeEq}(x, y) \wedge \text{BeforeEq}(y, z)$
- $\text{Exists}_\text{At}(x, y) \equiv \text{BetweenEq}(\text{Begin}_\text{Of}(x), y, \text{End}_\text{Of}(x))$
- $\text{Is}_\text{Occurring}_\text{At}(x, y) \equiv \exists z \text{ Occurrence}_\text{Of}(z, x) \wedge \text{BetweenEq}(\text{Begin}_\text{Of}(z), y, \text{End}_\text{Of}(z))$

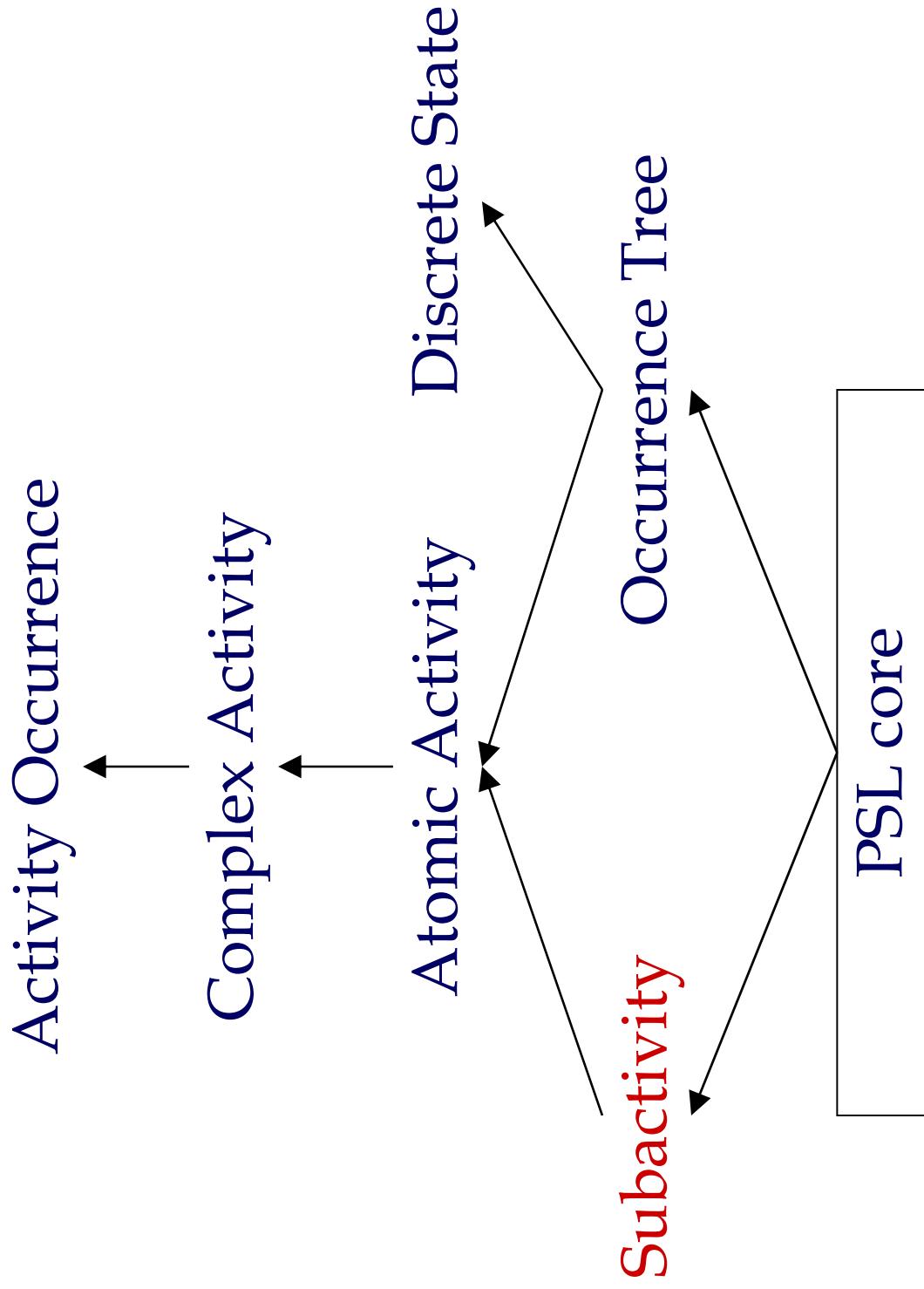
Summary of PSL Core

- The core theory is very limited
 - No composition
 - No concurrency
- PSL outer core attempts to address this issue with six extensions:
 - Subactivity
 - Occurrence trees
 - Discrete states
 - Atomic activities
 - Complex activities
 - Activity occurrence

Further Extensions



PSL Outer Core



Subactivity Extension

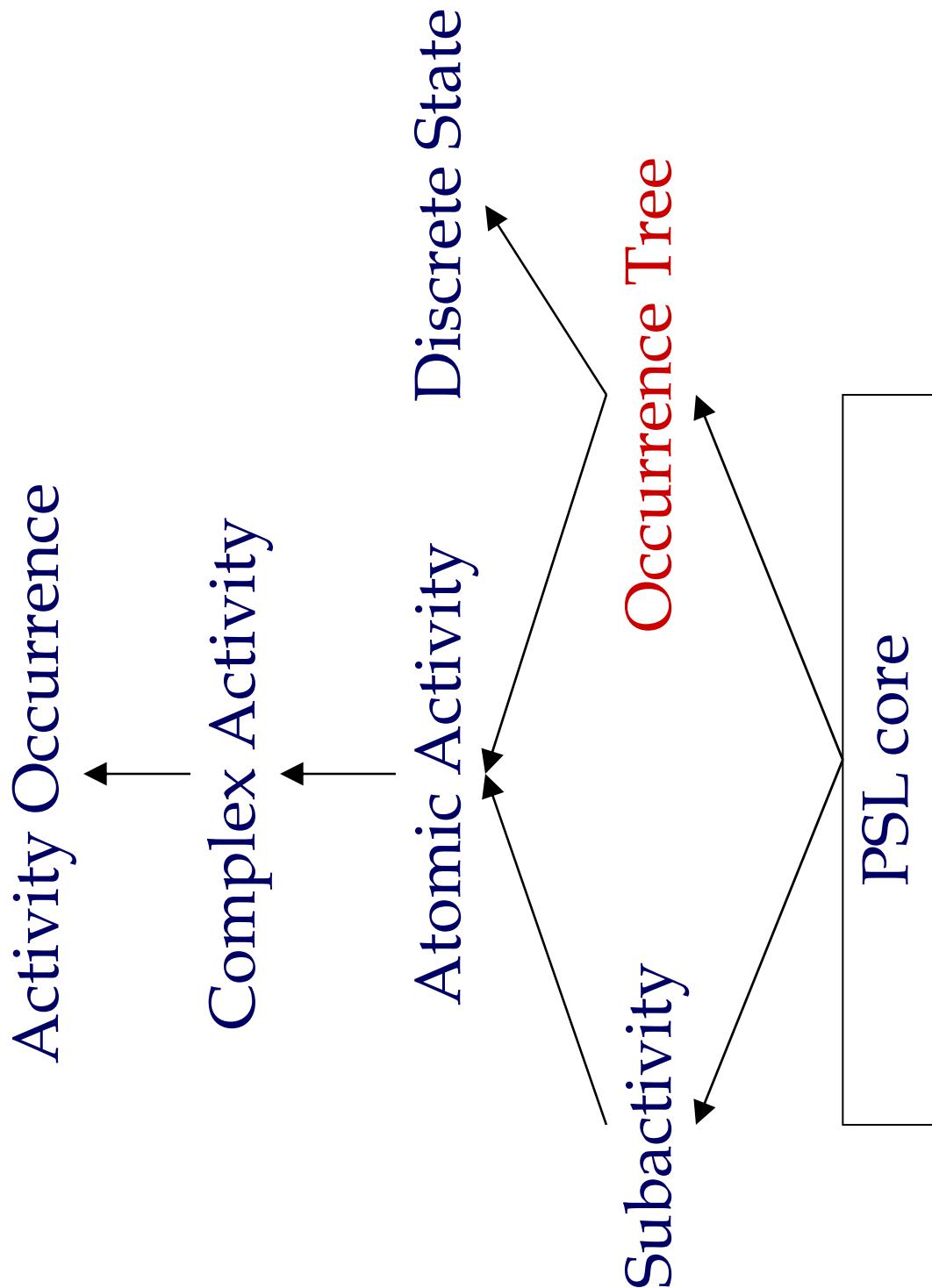
- Express the **logical** relationships of “composition”
 - But not about “how”
- The composition relation is a discrete partial ordering,
in which primitive activities are the minimal elements

- New relation: *Subactivity* (x, y)
 - x is a subactivity of y
- New class: *Primitive* (x)
 - x is a primitive activity
- Can be defined using *Subactivity*:
$$\text{Primitive}(x) \equiv \forall y (\text{Subactivity}(y, x) \rightarrow y = x)$$

Axioms

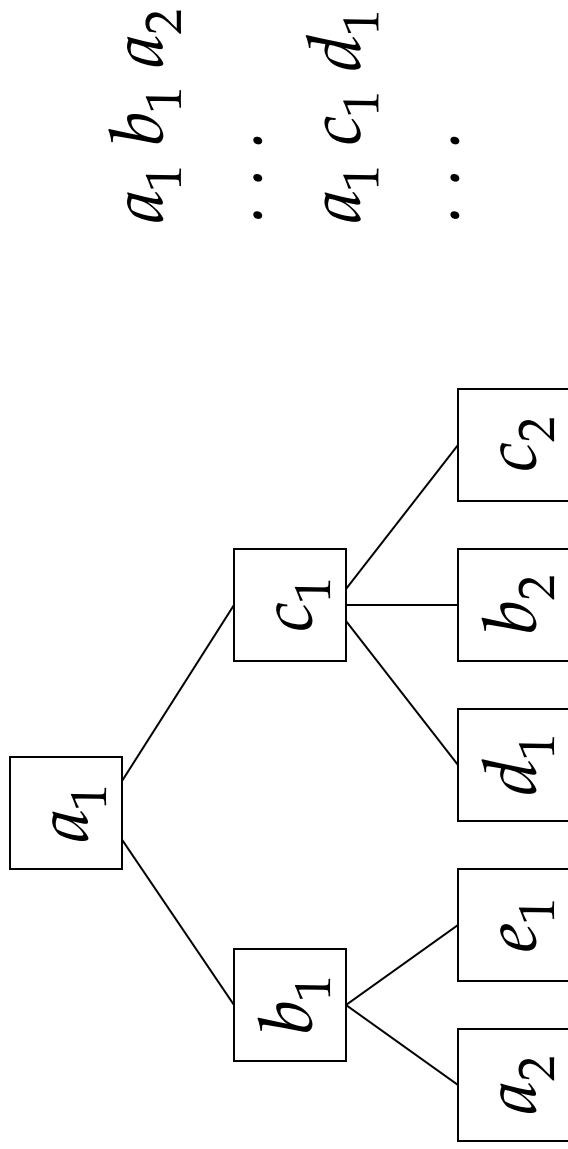
- Axiom 1: Subactivity is a relation over activities
- Axiom 2: The subactivity relation is reflexive
- Axiom 3: The subactivity relation is anti-symmetric
- Axiom 4: The subactivity relation is transitive
- Axiom 5: The subactivity relation is a discrete ordering,
so every activity has an upwards successor in the
ordering
- Axiom 6: The subactivity relation is a discrete ordering,
so every activity has a downwards successor in the
ordering

PSL Outer Core

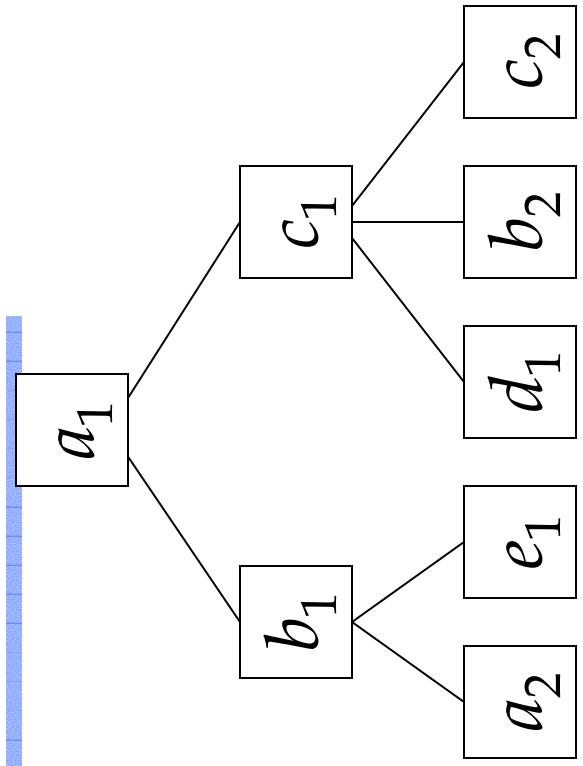


Occurrence Tree Extension

- Capture the set of all discrete sequences of activity occurrences
- An occurrence tree is a partially ordered set of activity occurrences
 - for a given set of activities, all discrete sequences of their occurrences are branches of the tree



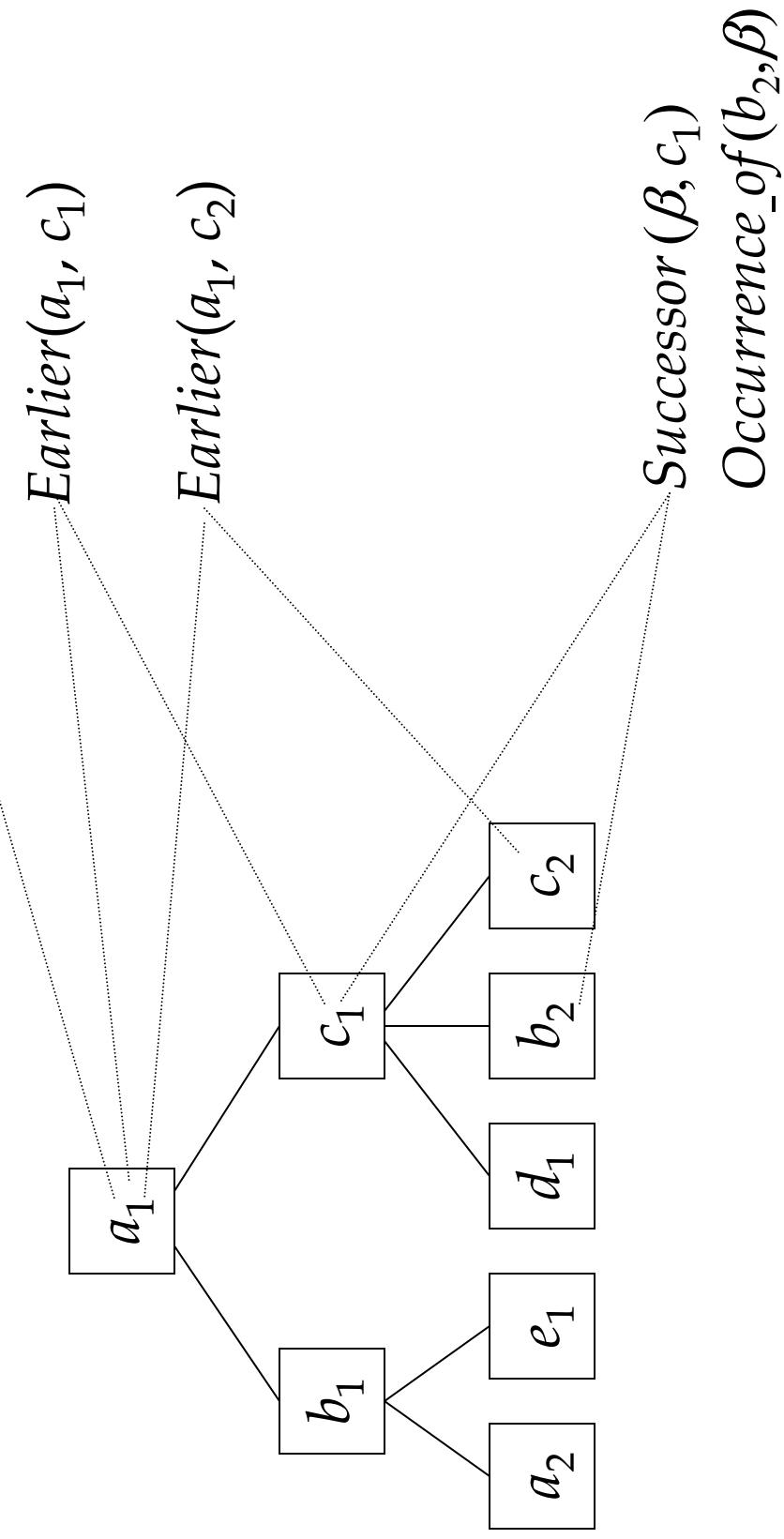
Occurrence Tree



- There are constraints on which activities can possibly occur in some domain
- Every sequence of activity occurrences has an initial occurrence (which is the root of an occurrence tree)
- The ordering of activity occurrences in a branch of an occurrence tree respects the temporal ordering

Concepts

- Relations: *Earlier, Initial, Legal*
- Functions: *Successor*



Semantics of Occurrence Tree

- *Earlier* (x, y) :
 - two activity occurrences x and y are on the same branch of the tree and x is closer to the root than y
- *Initial* (x) :
 - the activity occurrence x is a root of the occurrence tree
- *Legal* (x) :
 - the activity occurrence x is an element of the legal occurrence tree
- *Successor* (x, y) :
 - returns the occurrence of activity x that follows consecutively after the activity occurrence y in the occurrence tree

Axioms

Axiom 1: The *Earlier* relation is restricted to activity occurrences

Axiom 2: *Earlier* is irreflexive

Axiom 3: *Earlier* is transitive

Axiom 4: A branch in the occurrence tree is totally ordered

Axiom 5: No occurrence is earlier than an initial occurrence

Axioms

Axiom 6: Every branch of the occurrence tree has an initial occurrence

Axiom 7: There is a unique initial occurrence for each activity

Axiom 8: The successor of an activity occurrence is an occurrence of the activity

Axiom 9: Every non-initial activity occurrence is the successor of another activity occurrence

Axiom 10: An occurrence x is earlier than the successor occurrence of y if and only if the occurrence y is later than x

Axioms

Axiom 11: The *legal* relation restricts activity occurrences

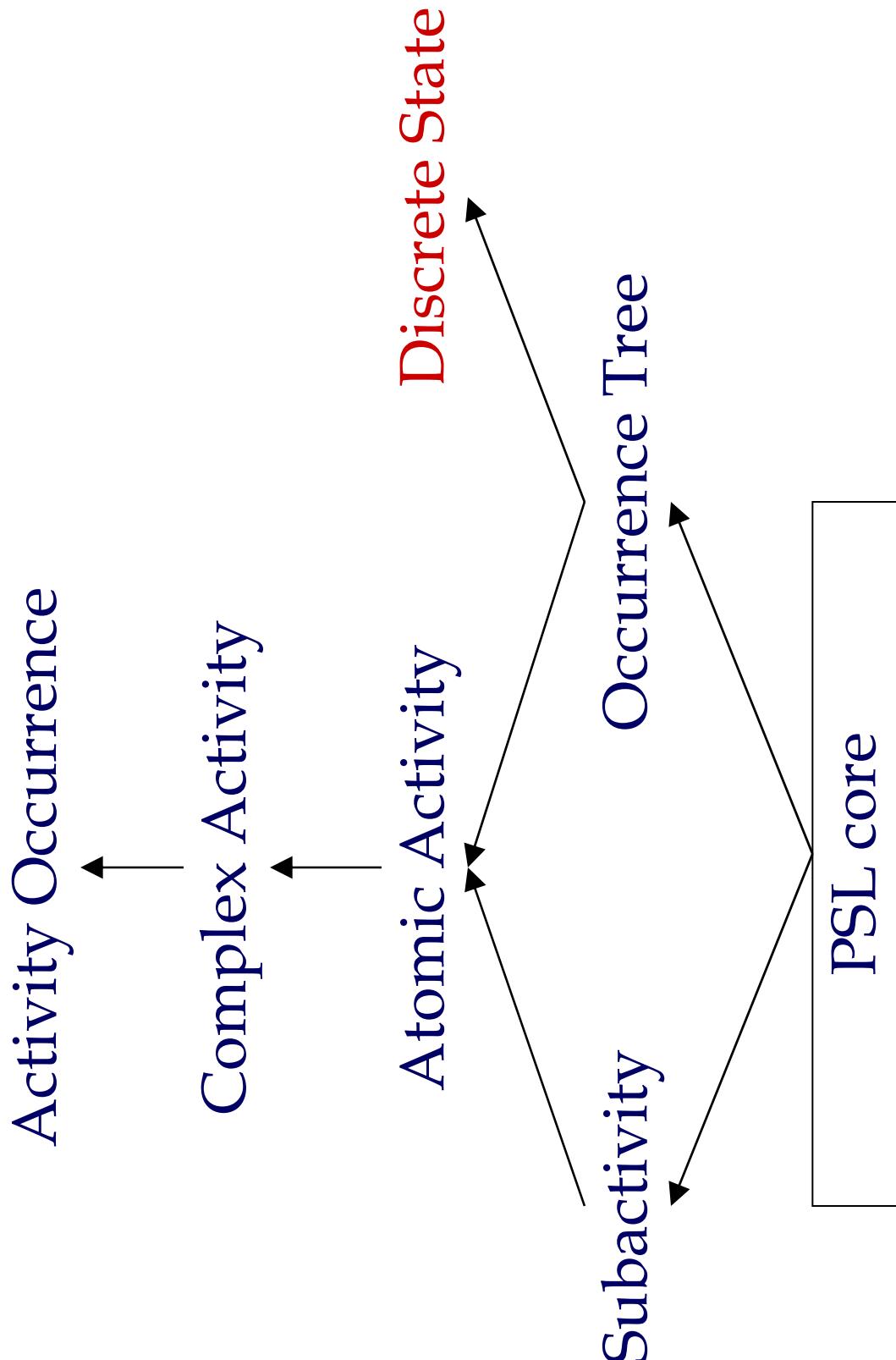
Axiom 12: If an activity occurrence is legal, all earlier activity occurrences in the occurrence tree are also legal

Axiom 13: The end of an activity occurrence is before to the begin of the successor of the activity occurrence

Semantics of Occurrence Tree

- $\text{poss}(x, y)$:
the activity x can possibly occur after the activity occurrence y
- $\text{precedes}(x, y)$:
the activity occurrence x is earlier than the activity occurrence y in the occurrence tree and such that all activity occurrences between them correspond to activities that are possible

PSL Outer Core



Theory of Discrete States

- Capture the basic intuitions about states and their relationship to activities
- State is changed by the occurrence of activities
- State can only be changed by the occurrence of activities
- State does not change during the occurrence of an activity in the occurrence tree

Semantics of Discrete States

- $state(x)$:
 - x is a member of the set of states in the universe of discourse of the interpretation
 - States are a subclass of object
 - $holds(x, y)$:
 - the state x is true after the activity occurrence y
 - $prior(x, y)$:
 - the state x is true prior to the activity occurrence y
- 7 axioms

Axioms for Discrete States

Axiom 1: States are objects

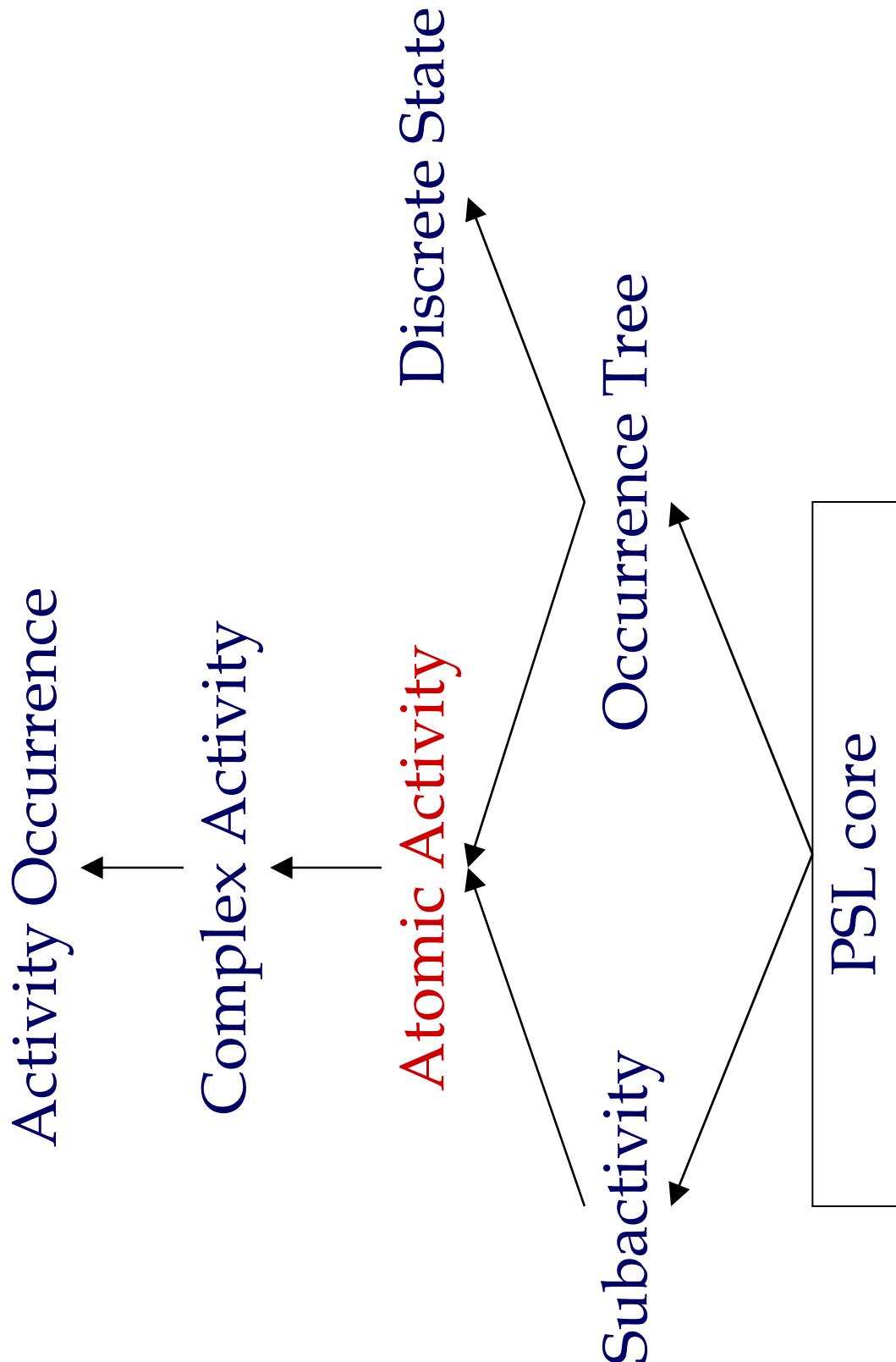
Axioms 2, 3: The *holds* and *prior* relations are only between states and activity occurrences

Axiom 4: All initial occurrences agree on the states that hold prior to them

Axiom 5: A state *holds* after an occurrence if and only if it holds prior to the successor occurrence

Axioms 6, 7: If a state holds (resp. does not hold) after some activity occurrence, then there exists an earliest activity occurrence along the branch where the state holds (resp. does not hold)

PSL Outer Core



Theory of Atomic Activities

- Concurrent aggregation of primitive activities
- Concurrency is represented by the occurrence of one concurrent activity rather than multiple concurrent occurrences
- Every concurrent activity is equivalent to the composition of a set of primitive activities

Semantics

- $atomic(x)$:
 x is either primitive or the concurrent superposition of a set of primitive activities
- $conc(x, y)$:
returns the atomic activity that is the concurrent superposition of the two atomic activities x and y
- 9 axioms

Axioms for Atomic Activities

Axiom 1: Primitive activities are atomic

Axiom 2: The function conc is idempotent

Axiom 3: The function conc is commutative

Axiom 4: The function conc is associative

Axiom 5: The concurrent aggregation of atomic action is an atomic action

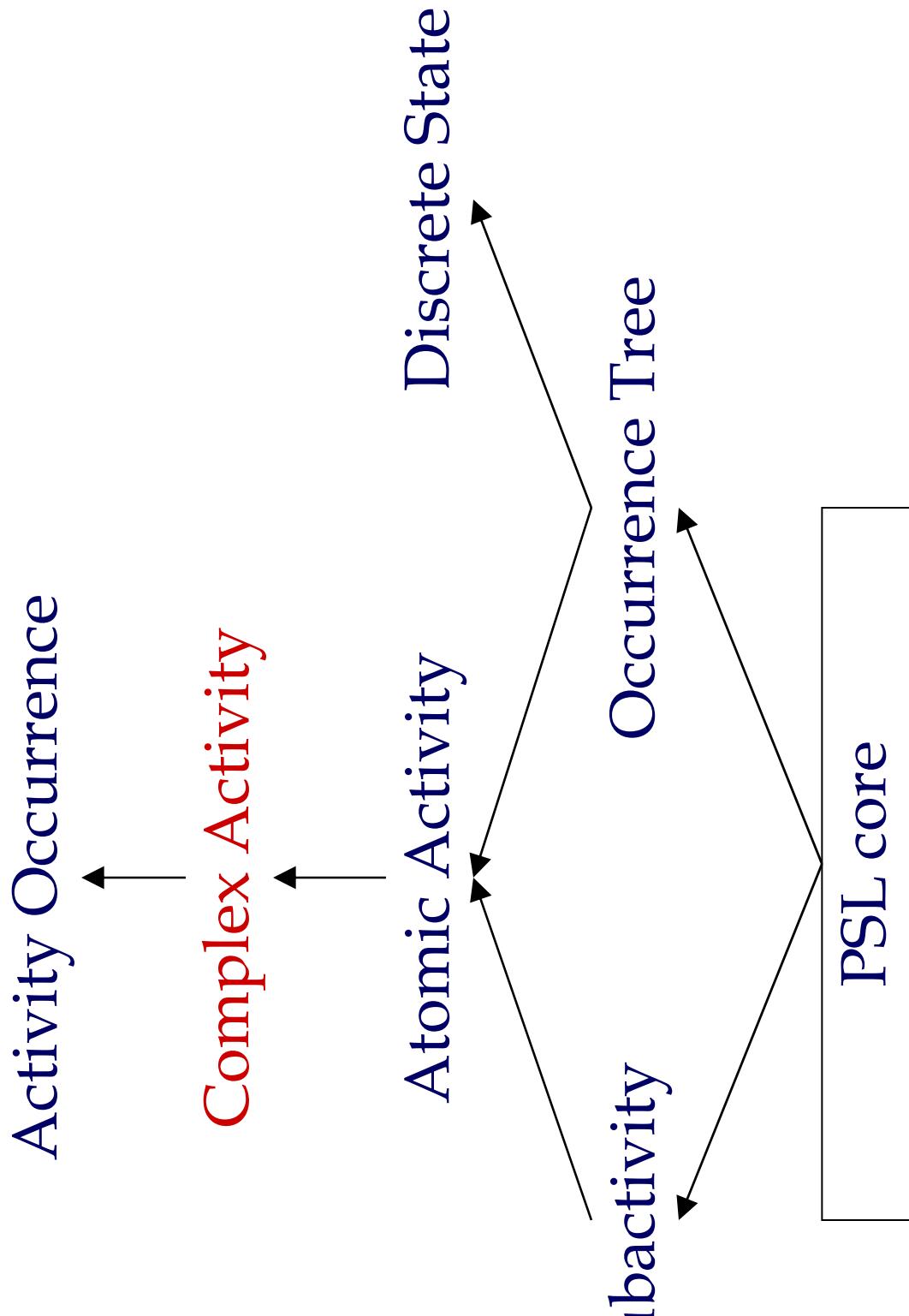
Axiom 6: An atomic activity x is a subactivity of an atomic activity y if and only if y is an idempotent for x

Axiom 7: An atomic action has a subactivity if and only if there exists another atomic activity which can be concurrently aggregated

Axiom 8: The semi-lattice of atomic activities is distributive

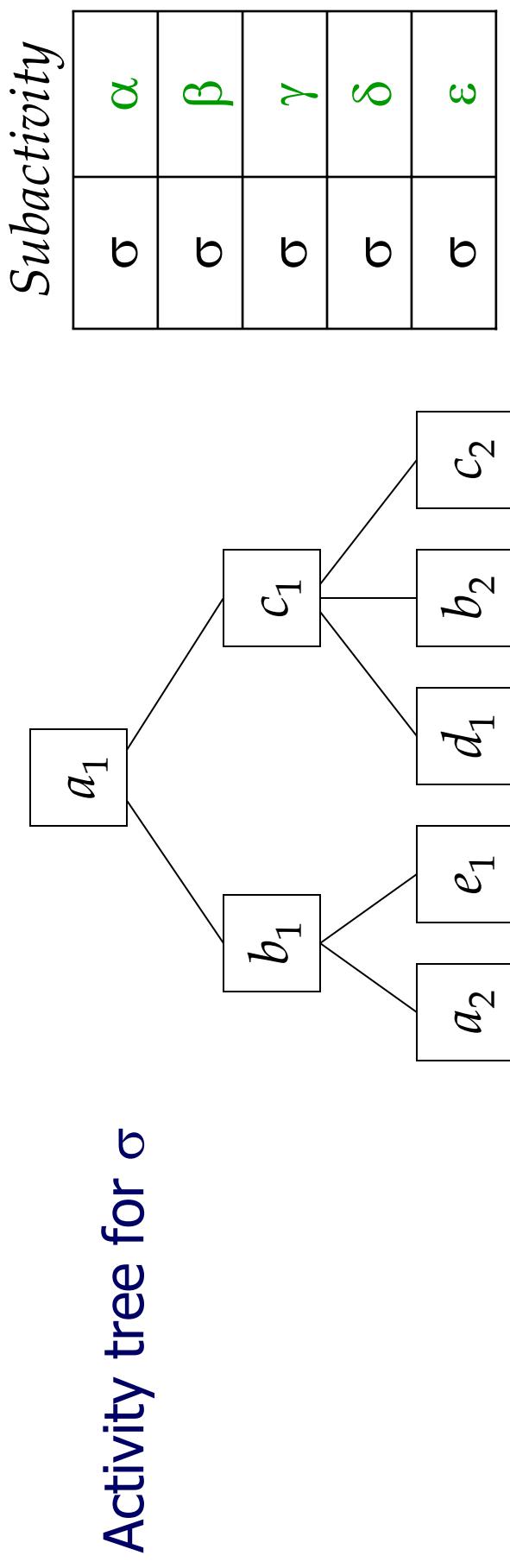
Axiom 9: Only atomic activities can be elements of the legal occurrence tree

PSL Outer Core



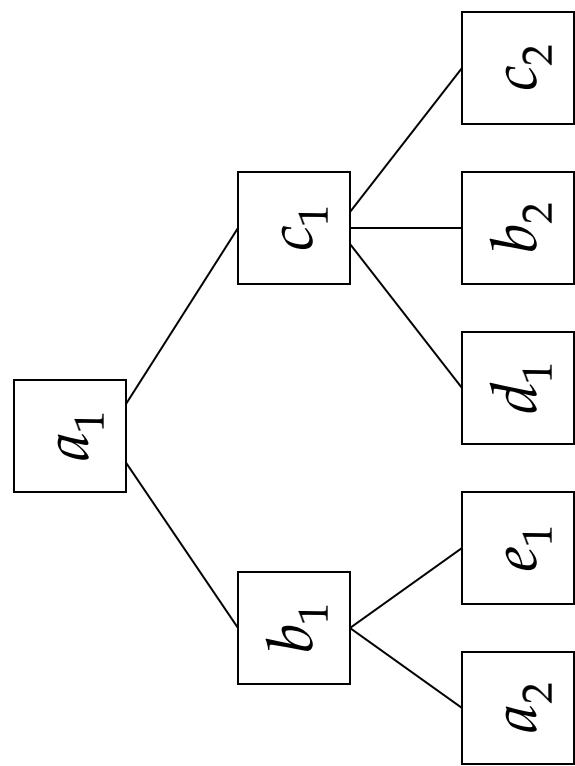
Theory of Complex Activities

- Representing complex activities and the relationship between occurrences of an activity and occurrences of its subactivities
- An **activity tree** consists of all possible sequences of atomic subactivity occurrences beginning from a root subactivity occurrence



Activity Trees

- Different subactivities may occur on different branches of the activity tree
- An activity will in general have multiple activity trees within an occurrence tree, and not all activity trees for an activity need be isomorphic
- Not every occurrence of a subactivity is a subactivity occurrence. There may be other external activities that occur during an occurrence of an activity



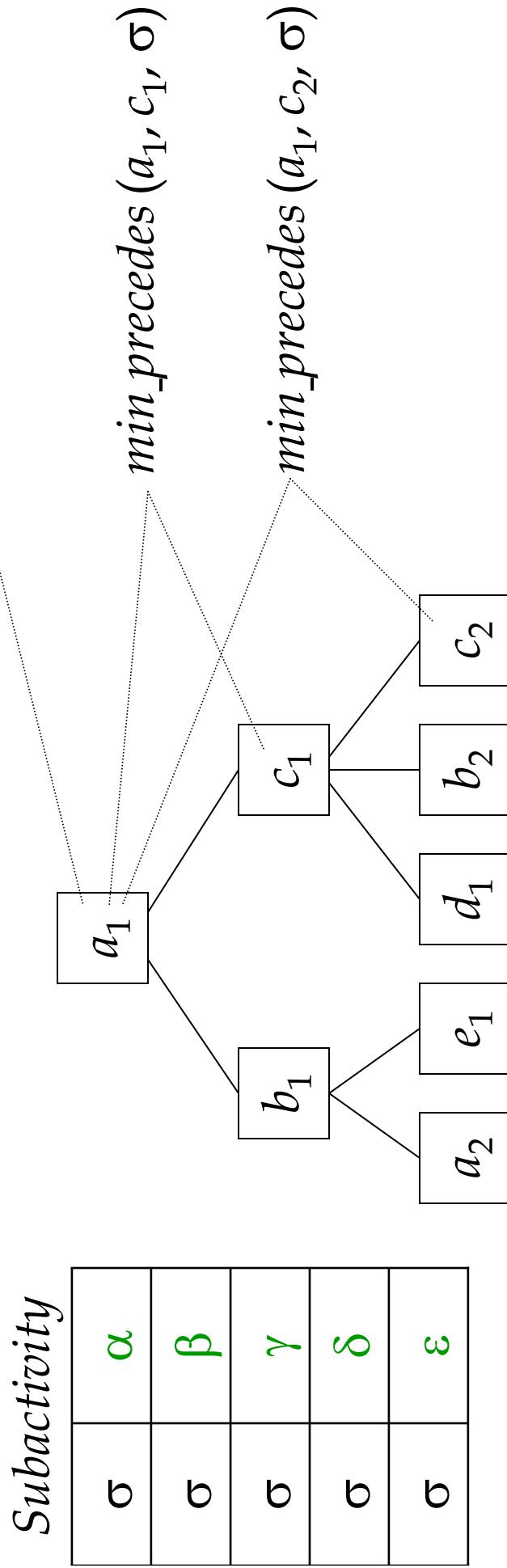
Subactivity	
σ	α
σ	β
σ	γ
σ	δ
σ	ϵ

Semantics of Complex Activities

- $\text{min_precedes}(x, y, z)$:
 - x and y are subactivity occurrences in the activity tree for z , and x precedes y
 - Any occurrence of an activity z corresponds to an activity tree. The activity occurrences within this tree are the subactivity occurrences of the occurrence of z
- $\text{root}(x, y)$:
 - the activity occurrence x is the root of an activity tree for y

Semantics of Complex Activities

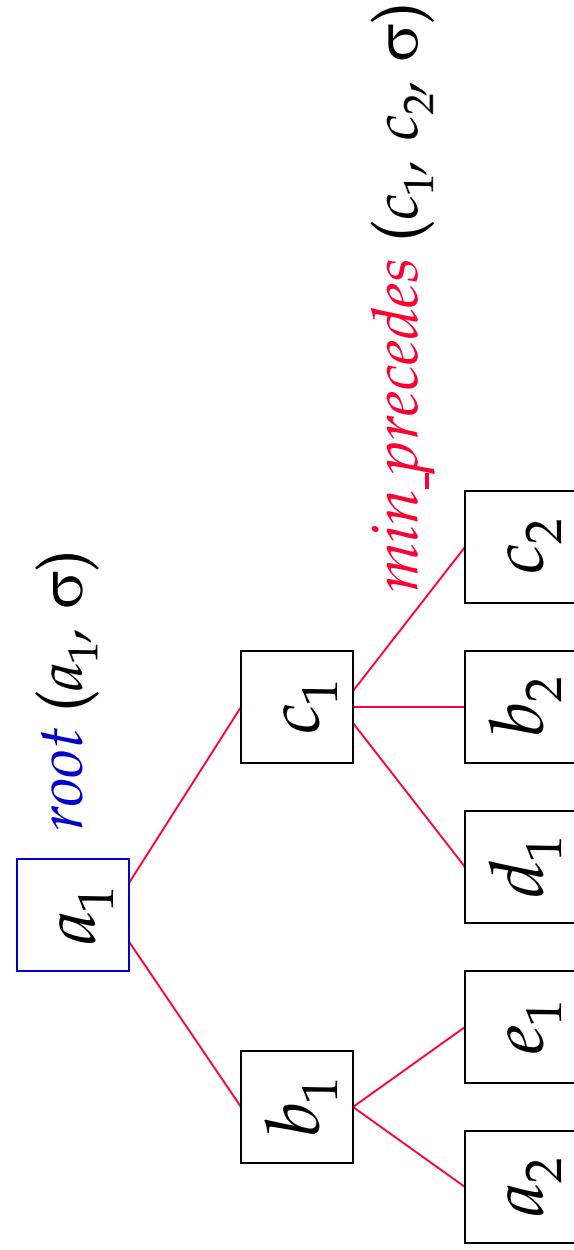
- $\text{min_precedes}(x, y, z) : x \text{ and } y \text{ are subactivity occurrences in the activity tree for } z, \text{ and } x \text{ precedes } y$
- $\text{root}(x, y) : \text{the activity occurrence } x \text{ is the root of an activity tree for } y$



Nodes in An Activity Tree

Axioms 1-2: Occurrences in the activity tree for an activity correspond to atomic subactivity occurrences of the activity

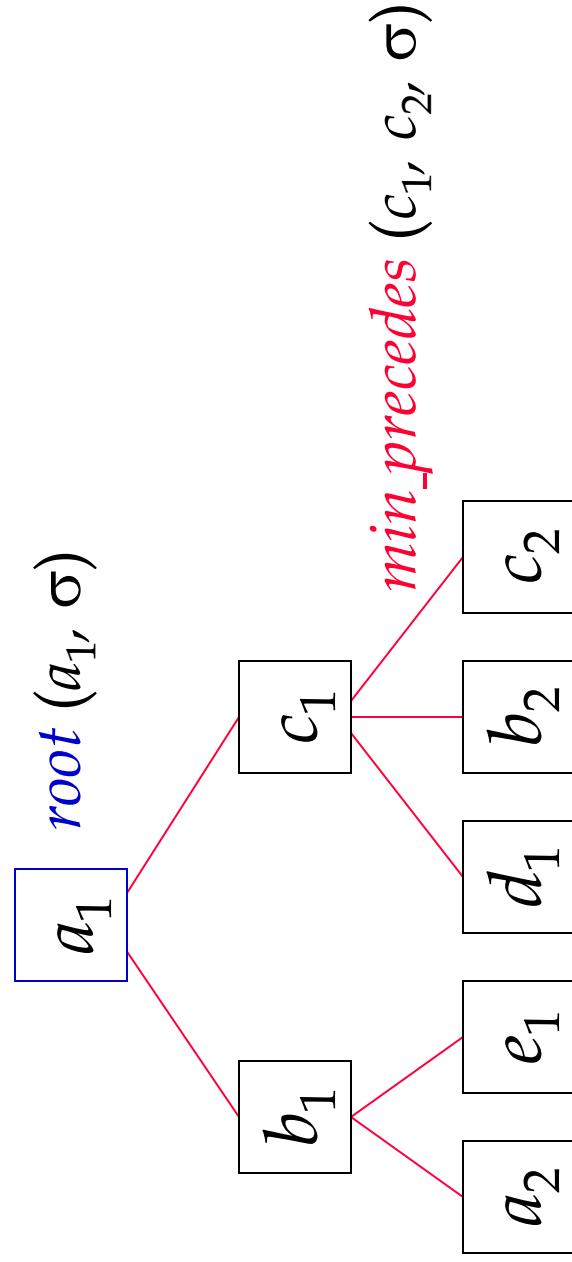
Axiom 3: Root occurrences in the activity tree correspond to atomic subactivity occurrences of the activity



The Root of An Activity Tree

Axiom 4: All activity trees have a root subactivity occurrence

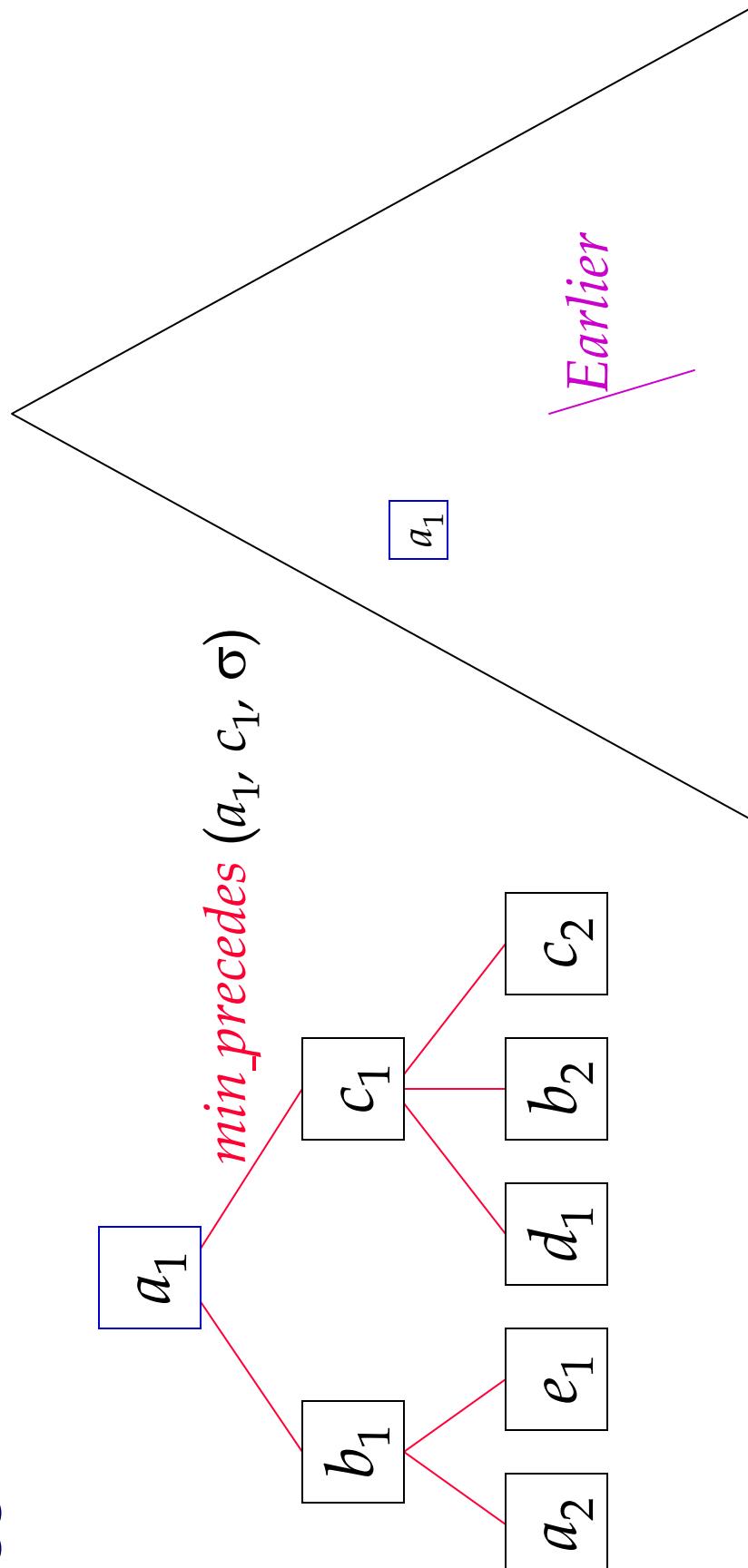
Axiom 5: No subactivity occurrences in an activity tree occur earlier than the root subactivity occurrence



Activity Tree and Occurrence Tree

Axiom 6: An activity tree is a subtree of the occurrence tree

Axiom 7: Root occurrences are elements of the occurrence tree



Properties of An Activity Tree

Axiom 8: Every atomic activity occurrence is an activity tree containing only one occurrence

Axiom 9: Activity trees are discrete

Axiom 10 & 11: Subactivity occurrences on the same branch of the occurrence tree are on the same branch of the activity tree

Axiom 12: The activity tree for a complex subactivity occurrence is a subtree of the activity tree for the activity occurrence

Semantics of Complex Activities

- $\text{subtree}(x, y)$:
every atomic subactivity occurrence in the activity tree
for x is an element of the activity tree for y
- $\text{do}(x, y, z)$:
 y is the root of an activity tree for x , z is a leaf of the
same activity tree, both activity occurrences are
elements of the same branch of the activity tree
- $\text{leaf}(x, y)$:
the activity occurrence x is the leaf of an activity tree
for y
- $\text{next_subocc}(x, y, z)$:
 x precedes y in the tree for z and there does not exist a
subactivity occurrence between them in the tree

PSL Outer Core

Activity Occurrence

Complex Activity

Atomic Activity

Subactivity

Occurrence Tree

Discrete State

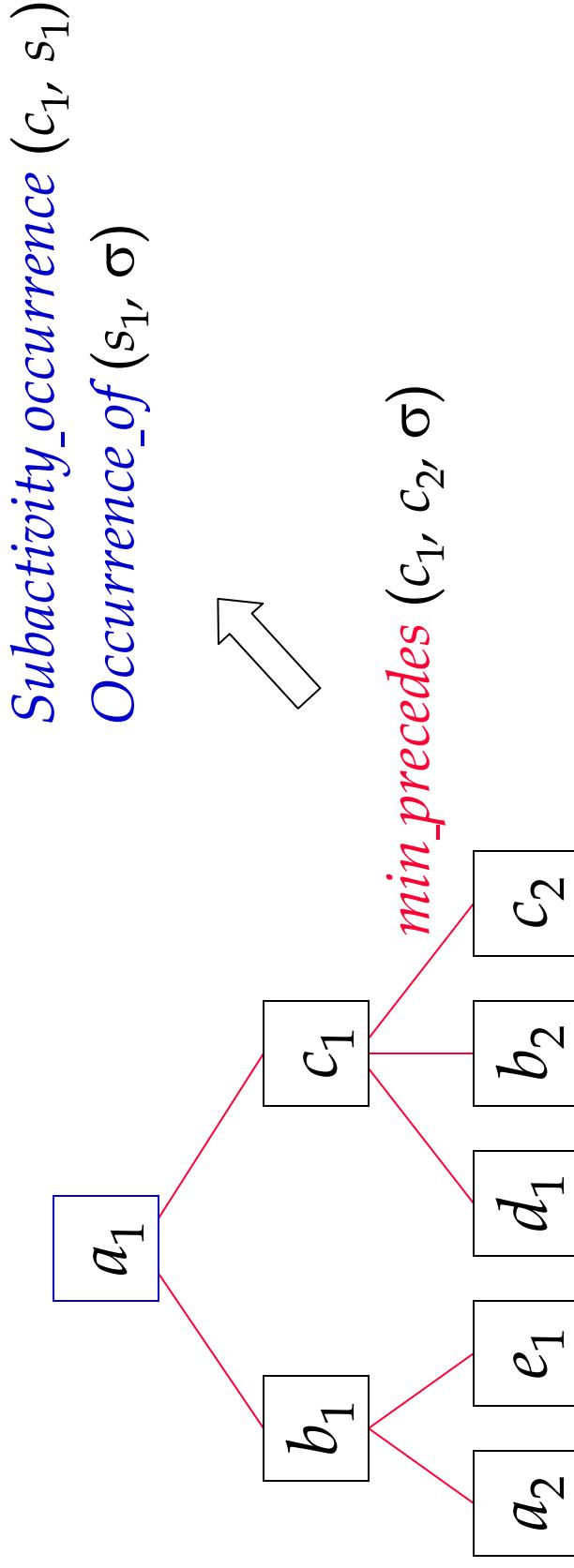
PSL core

Complex Activity Occurrences

- Complex activity occurrences correspond to activity trees
 - Not elements of the legal occurrence tree
 - This theory ensure that complex activity occurrences correspond to branches of activity trees
 - Each complex activity occurrence has a unique atomic root occurrence
 - Each finite complex activity occurrence has a unique atomic leaf occurrence
 - A subactivity occurrence corresponds to a sub-branch of the branch corresponding to the complex activity occurrence

Semantics for Activity Occurrences

- $\text{subactivity_occurrence}(x, y)$:
 - the branch corresponding to the activity occurrence x is a subset of the branch corresponding to activity occurrence y



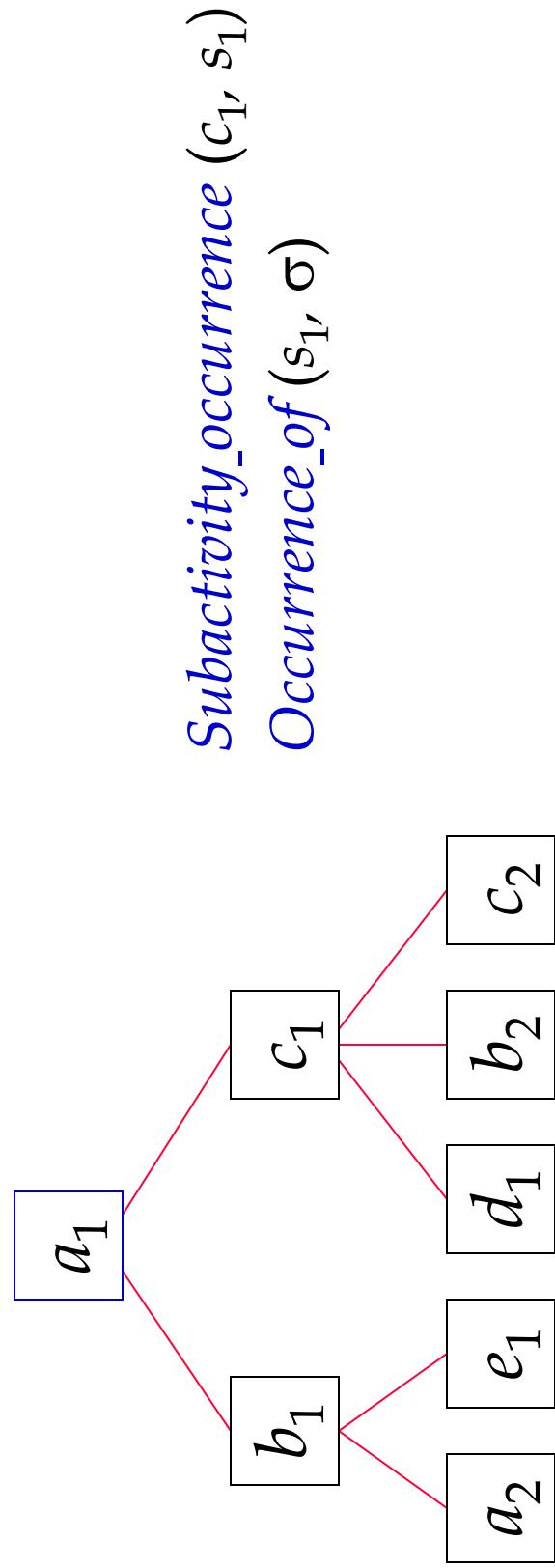
Axioms for Complex Activity Occurrences

- Axiom 1:** There exists an occurrence of an activity x for every branch of an activity tree for x . All atomic subactivity occurrences on the branch are subactivity occurrences of the occurrence of x
- Axiom 2:** There exists an occurrence of an activity x for every branch of an activity tree for x . All root subactivity occurrences on the branch are subactivity occurrences of the occurrence of x

Axioms for Complex Activity Occurrences

Axiom 3: All atomic subactivity occurrences of a complex activity occurrence are elements of the same branch of the activity tree

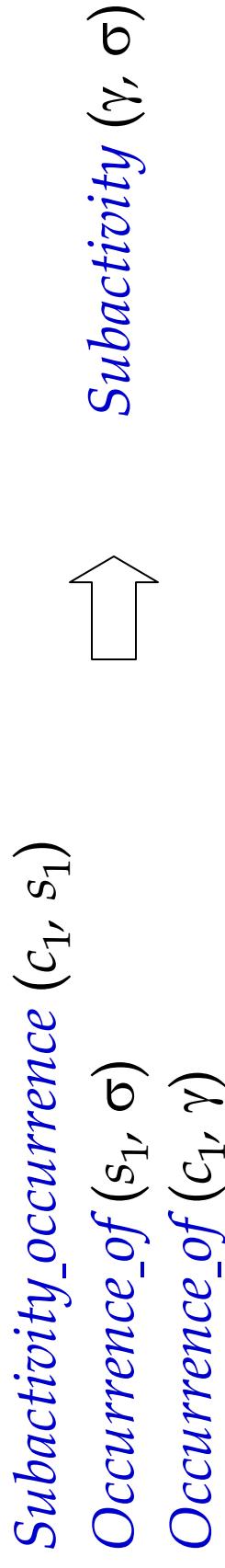
Axiom 4: All elements of the same branch of an activity tree are atomic subactivity occurrences of the same activity occurrences



Axioms for Complex Activity Occurrences

Axiom 5: The *subactivity_occurrence* relation preserves the subactivity relation

Axiom 6: The *subactivity_occurrence* relation is transitive



Axioms for Complex Activity Occurrences

Axiom 7: Occurrences of subactivities are subactivity occurrences if the occurrences satisfy branch containment

Axiom 8: The begin of timepoint for a complex activity occurrence is equal to the begin of timepoint of its root occurrence

Axiom 9: The end of timepoint for a complex activity occurrence is equal to the end of timepoint of its leaf occurrence

Semantics for Activity Occurrences

- $\text{mono}(x, y, z)$:
 - there is a one-to-one mapping between branches of an activity tree for z that maps the atomic subactivity occurrence x to the atomic subactivity occurrence y

Axioms for Complex Activity Occurrences

Axiom 10: The mono relation is a branch homomorphism

Axiom 11: If an atomic subactivity occurrence is mapped in a branch homomorphism, then there exists another atomic subactivity occurrence that is mono with it

Axiom 12: The mono relation is restricted to one-to-one homomorphisms between different branches of the activity tree

PSL: Summary

- PSL core + outer core provides a good starting point for describing activities and executions
 - Capable of expressing FSA based formalism
- Formalism with well defined semantics based on situation calculus (first order logic plus time)
 - Linear time logic?
- Messages: needs additional theory (theories)