CSL Cognitive Systems Laboratory

High-level Interpretation of the Visual Environment

Bernd Neumann

Department of Informatics
Hamburg University
Germany

kogs-www.informatik.uni-hamburg.de
neumann@informatik.uni-hamburg.de

Presentation at the ICRA Workshop "Semantic Information in Robotics" on 10.4.2007 in Rome

Agenda

- Compositional Hierarchies for Scene Interpretation
  - Aggregates as Conceptual Units
  - Part-Whole Reasoning
- Logics of Scene Interpretation
  - Model Construction
  - Abduction
- Using Description Logics for Scene Interpretation
- Probabilistic Inferences

Compositional Hierarchies for Scene Interpretation

At the core of scene interpretation:

Finding meaningful aggregates of entities in space and time

- **object constellations**
  e.g. living room, laid table, parking ground, building facade
- **activities, events, episodes**
  e.g. garbage collection, traffic situations, laying a table

Aggregates as Conceptual Structures

Generic frame-based representation of an aggregate concept:

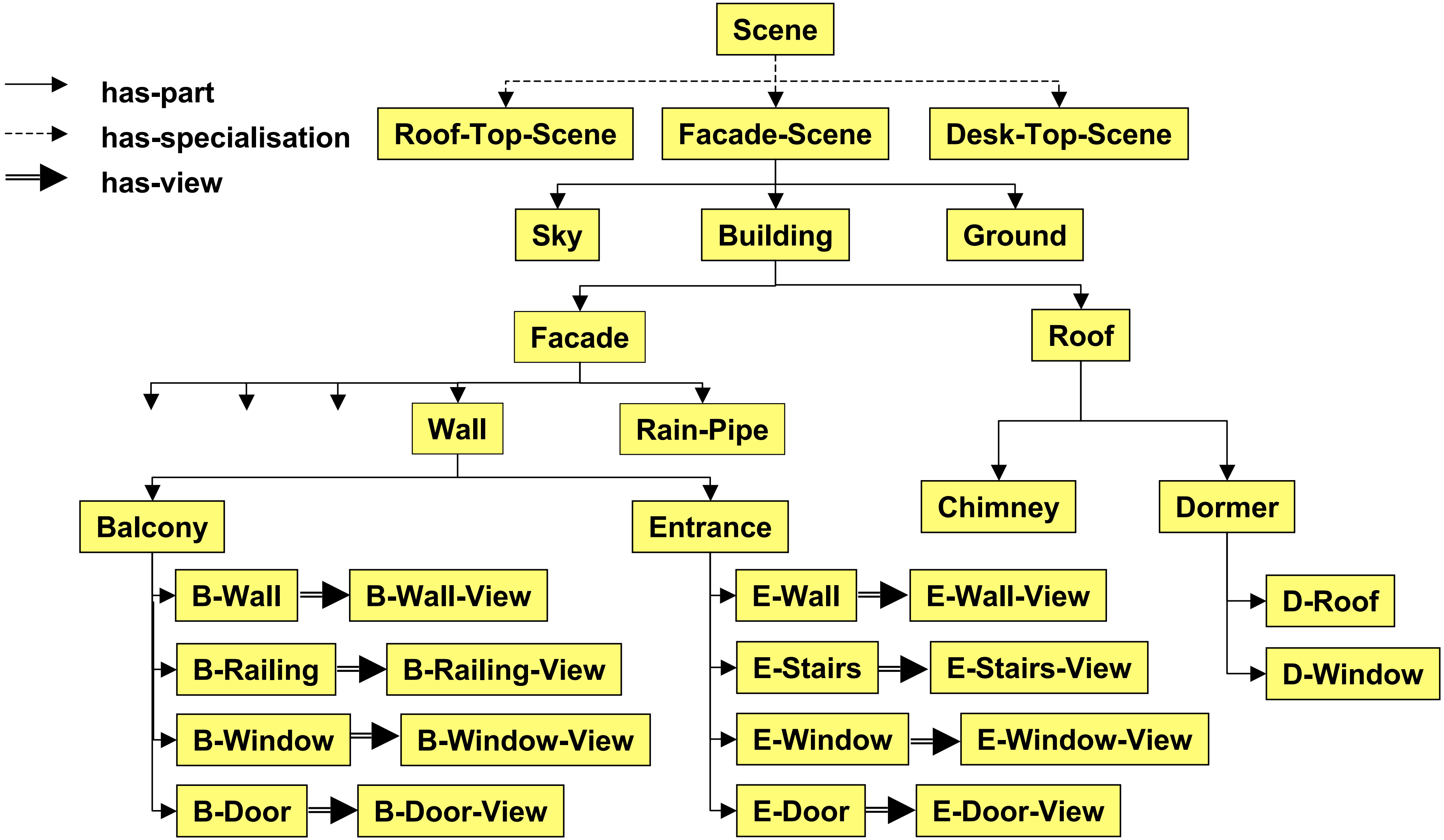
aggregate name
parent concepts
external properties
parts
constraints between parts

Restriction:

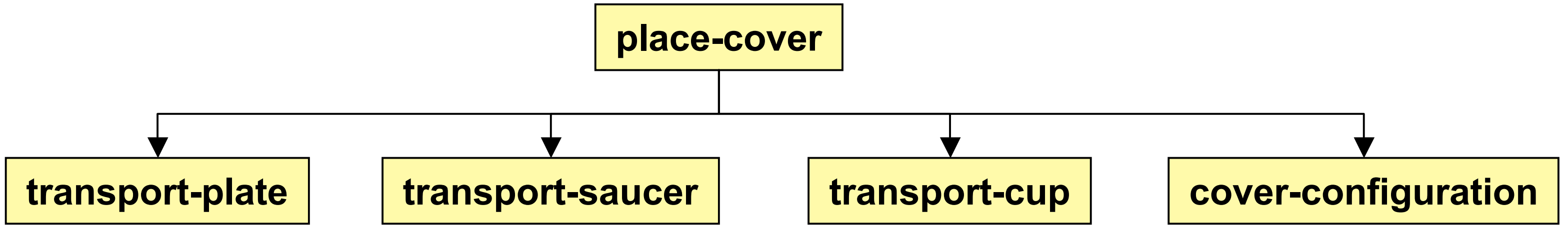
No constraints between components of different aggregates

- *aggregate name* contains a symbolic ID
- *parent concepts* contains IDs of taxonomical parents
- *external properties* provide a description of the aggregate as a whole
- *parts* refer to the subunits out of which an aggregate is composed
- *constraints* specify which relations must hold between the parts

Conceptual Facade Hierarchy



Occurrence Model for Placing a Cover

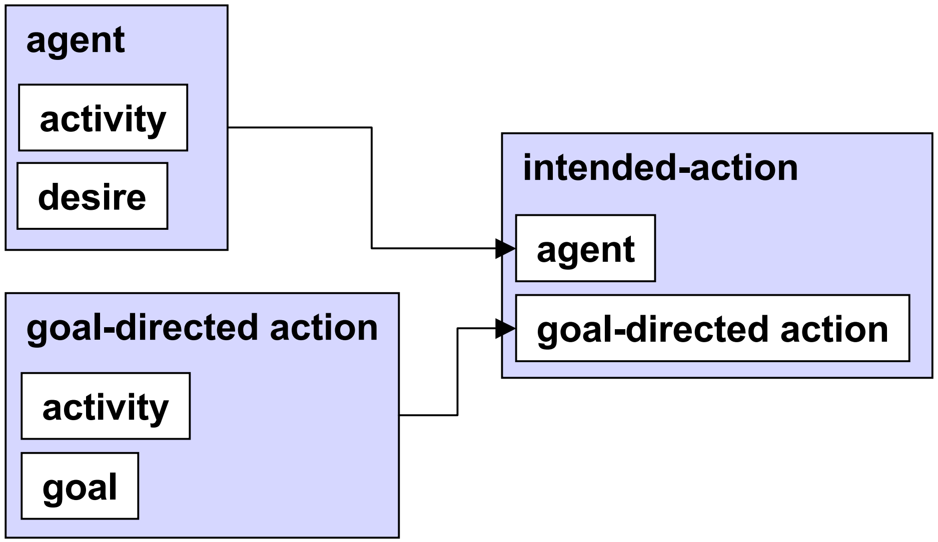


name:	place-cover	
parents:	:is-a agent-activity	
parts:	pc-tp1 :is-a (transport with (tp-obj :is plate))	%transport of a plate
	pc-tp2:is-a (transport with (tp-obj :is saucer))	%transport of a saucer
	pc-tp3 :is-a (transport with (tp-obj :is cup))	%transport of a cup
	pc-cv :is-a cover	%cover configuration
properties:	tb, te :is-a timepoint	%begin and end timepoint of place-cover
constraints:	pc-tp1.tp-ob = pc-cv.cv-pl	%transport-plate object same as cover-plate
	pc-tp2.tp-ob = pc-cv.cv-sc	%transport-saucer object same as cover-saucer
	pc-tp3.tp-ob = pc-cv.cv-cp	%transport-cup object same as cover-cup
	pc-cv.tb ≥ pc-tp1.te	%cover begins after plate transport
	pc-cv.tb ≥ pc-tp2.te	%cover begins after saucer transport
	pc-cv.tb ≥ pc-tp3.te	%cover begins after cup transport
	pc-tp3.tp-te ≥ pc-tp2.tp-te	%cup transport ends after saucer transport
	tb = pc-tp1.tb min pc-tp2.tb min pc-tp3.tb	
	te = pc-tp1.te max pc-tp2.te max pc-tp3.te	
	te ≤ tb + 80Δt	%place-cover may not last more than 80 time units

A Simple Model for Intention Recognition

Intended actions may be described by aggregates which connect observable actions with (unobservable) intentions of an actor.

name:	intended-place-cover
parents:	:is-a intended-action
parts:	ipc-pc :is-a place-cover
	ipc-ag :is-a agent with (ipc-ag.desire = ipc-pc.goal)
properties:	tb, te :is-a timepoint
constraints:	(temporal, spatial and other constraints on parts)



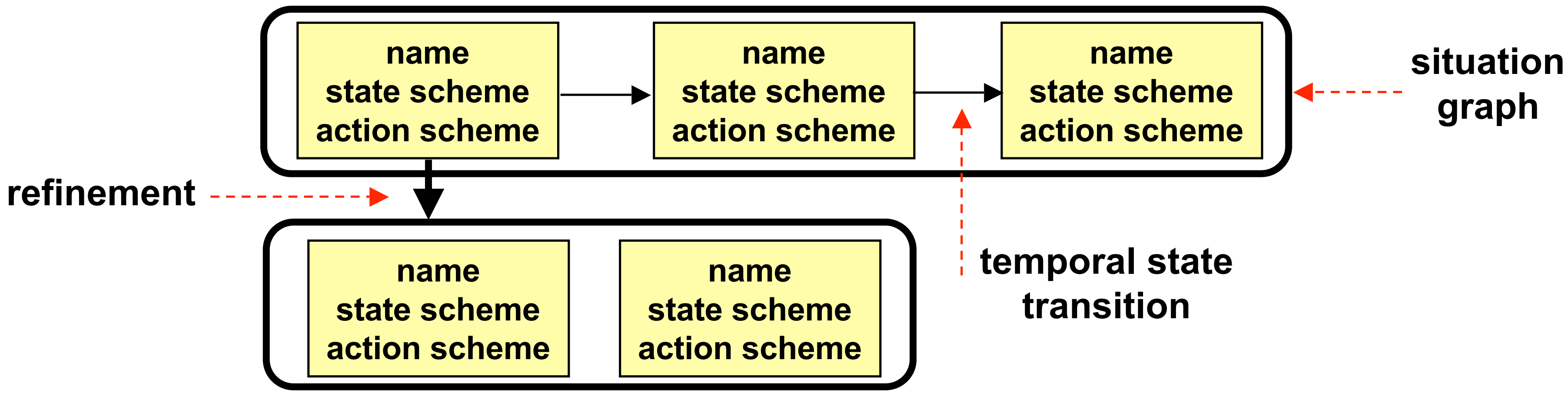
If an action is known to be goal-directed and an agent performs such an action, the agent is ascribed the intention to attain the goal.

Cognitive Systems Laboratory

Situation Graph Trees

Situation Graph Tree - Situation Scheme - State Scheme - Action Scheme

(Nagel et al. 99)

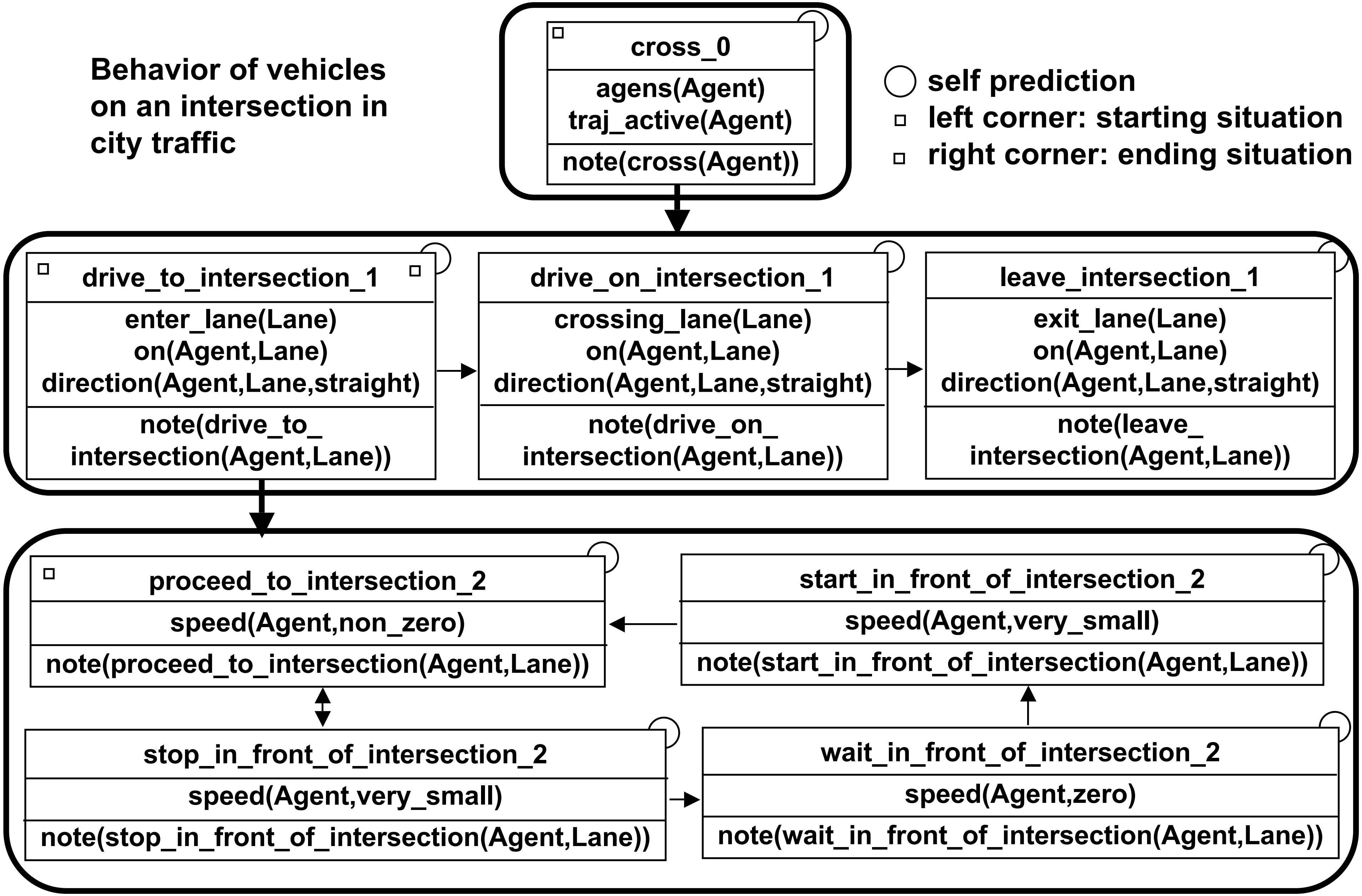


Example of Situation Graph Tree

CSL Cognitive Systems Laboratory

Behavior of vehicles on an intersection in city traffic

○ self prediction
▫ left corner: starting situation
▫ right corner: ending situation

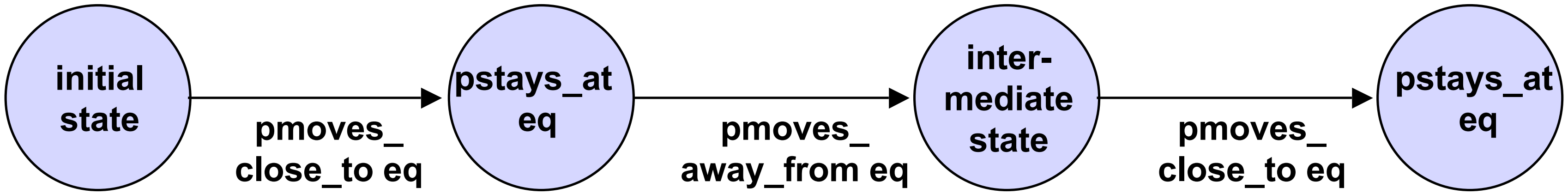


Scenarios

Scenario - Events - States (Thonnat et al. 2006)

```
Scenario(vandalism_against_ticket_machine,
        Physical_objects((p : Person),
                                (eq : Equipment, Name= “Ticket_Machine”) )
        Components(      (event s1: pmoves_close_to eq)
                                (state s2: pstays_at eq)
                                (event s3: pmoves_away_from eq)
                                (event s4: pmoves_close_to eq)
                                (state s5: pstays_at eq) )
        Constraints(       (s1 != s4) (s2 != s5)
                                (s1 before s2) (s2 before s3)
                                (s3 before s4) (s4 before s5) ) ) )
```

Notation as state transition graph:



Interpretation Steps

Aggregation

Inferring an aggregate from (incomplete) parts

Part-whole reasoning

Obtaining an interpretation at a higher abstraction level

Expansion

Inferring parts from an aggregate

Hypothesising occluded objects

Filling in for missing evidence

Predicting future events or reconstructing past events

Classification

Inferring objects from evidence

Specialisation

Assigning objects to specialised concepts along taxonomical hierarchies

Merging

Merging partial interpretations of a distributed interpretation process

Logics of Image Interpretation

Scene interpretation can be formalised as:

- ***Partial Model Construction***

  Construct a partial mapping of the symbols of your formal knowledge about the world into a real-world domain.

  An interpretation is a partial instantiation of formal knowledge consistent with evidence about the real-world domain.

- ***Abduction***

  Construct an explanation of real-world evidence from your formal knowledge about the real-world domain.

  An interpretation is an instantiation of formal knowledge which allows to deduce the evidence.

Scene Interpretation by Partial Model Construction

Given a knowledge base with
- general domain knowledge,
- specific context information,
- specific sensory evidence

construct a mapping of
- constant symbols into scene elements D,
- predicate and relation symbols into predicate and relation functions over D

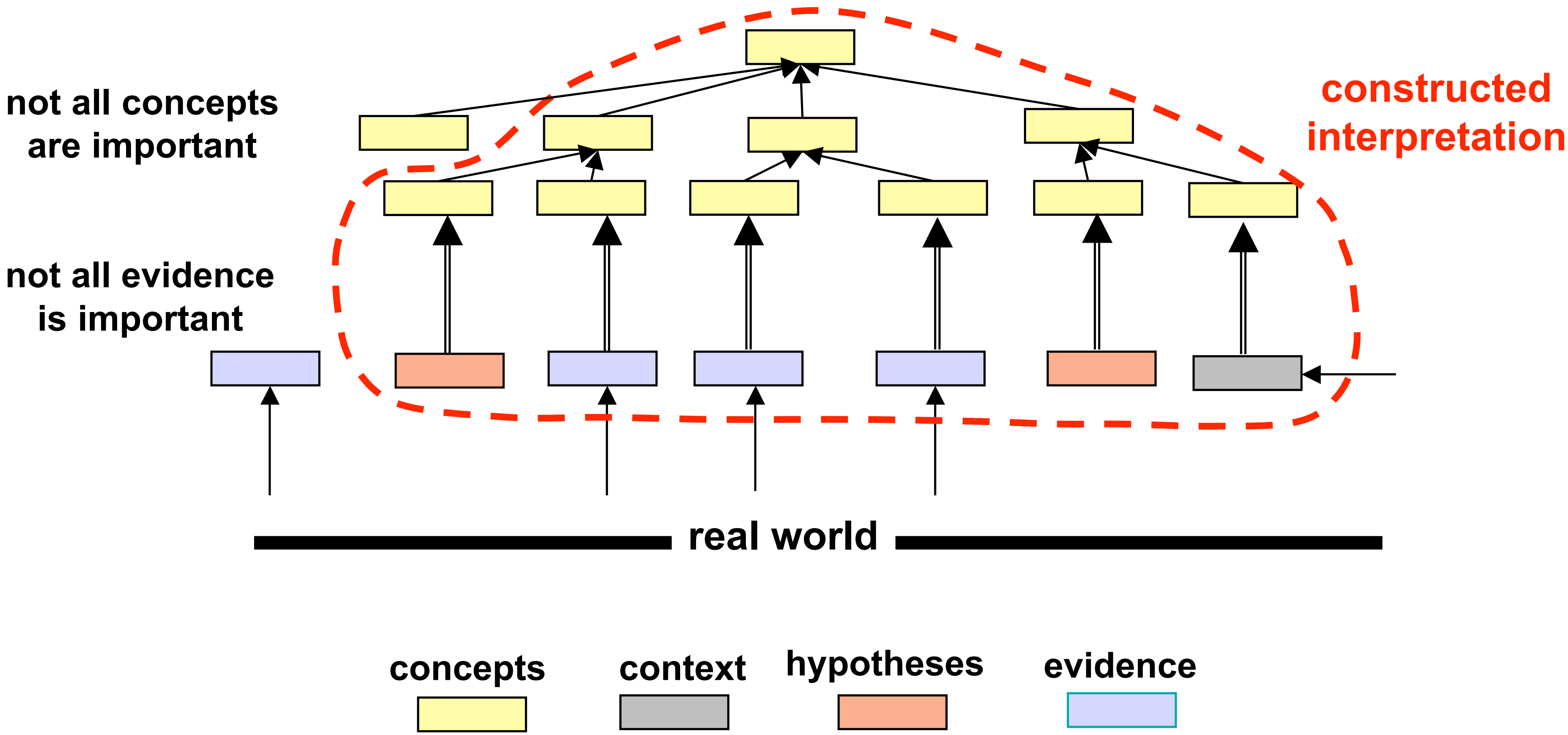
such that all predicates and relations are true.

- Operational semantics of low-level vision provide mapping into primitive constant and predicate/relation symbols.
- Hypotheses need no evidence.

Clowes: "Vision is controlled hallucination"

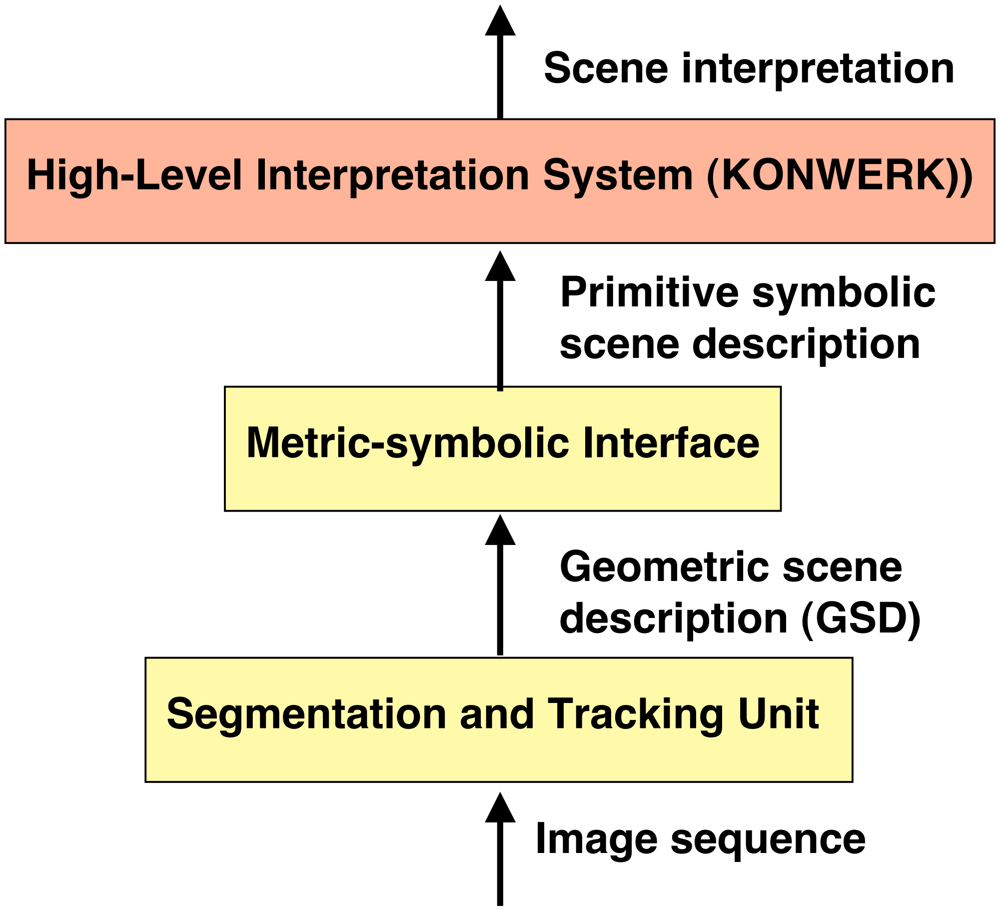
Consistent Interpretations in Compositional Hierarchies

A scene interpretation is a scene description in terms of instantiated aggregate concepts consistent with evidence and context information.



Scene Interpretation as Configuration

Model Construction is also the basis of knowledge-based configuration

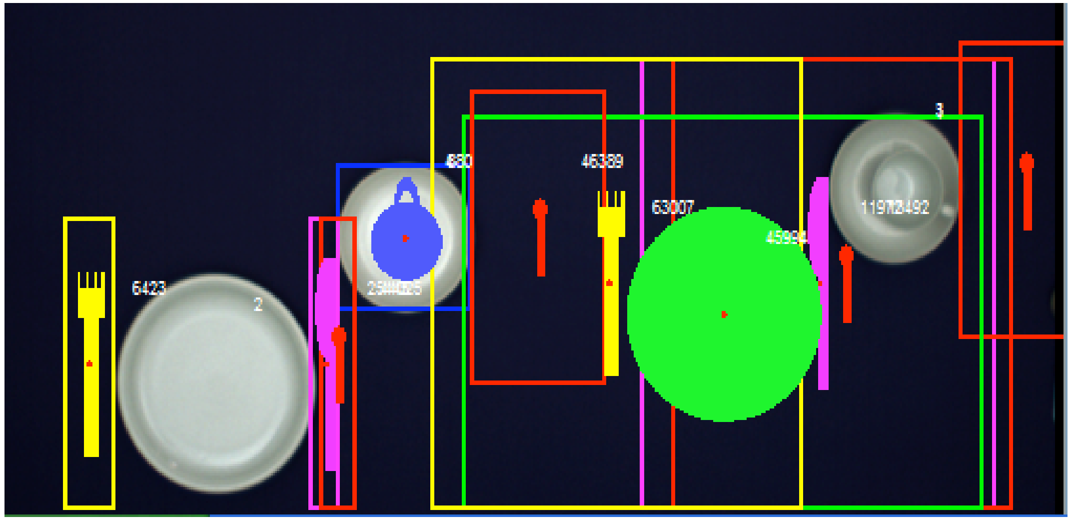


SCENIC uses a configuration system framework for scene interpretation

Hotz & Neumann 2005
Scene Interpretation as a Configuration Task
Künstliche Intelligenz, 3/2005, BöttcherIT Verlag, Bremen, 59-65

CSL Cognitive Systems Laboratory

Experimental Results (1)



natural views = evidence

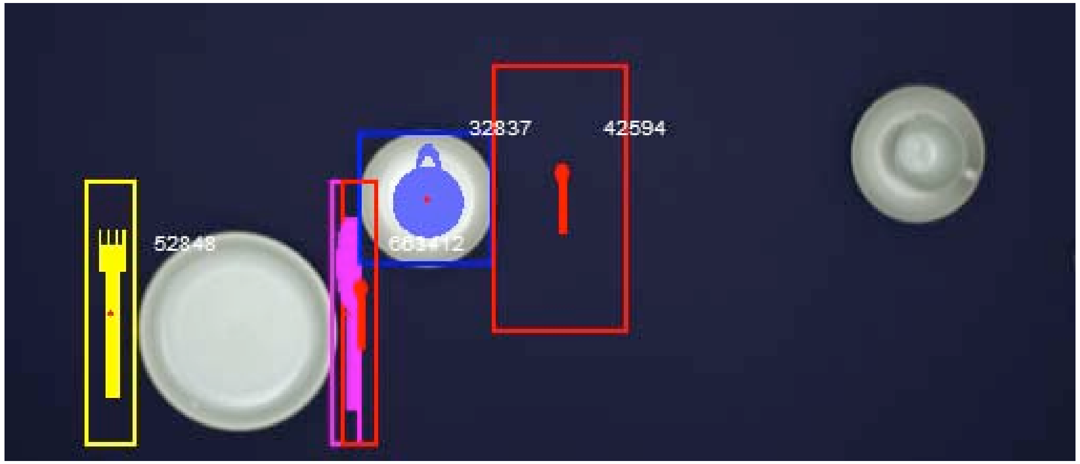
coloured shapes = hypotheses

boxes = expected locations

Intermediate state of interpretation after 51 interpretation steps:

- **"lay-dinner-for-2" hypothesis based on partial evidence**
- **predictions about future actions and locations**
- **high-level disambiguation of low-level classification**
- **influence of context**

Experimental Results (2)



- **alternative interpretation in terms of "dinner-for-one" and "cluttered-table" (after backtracking)**

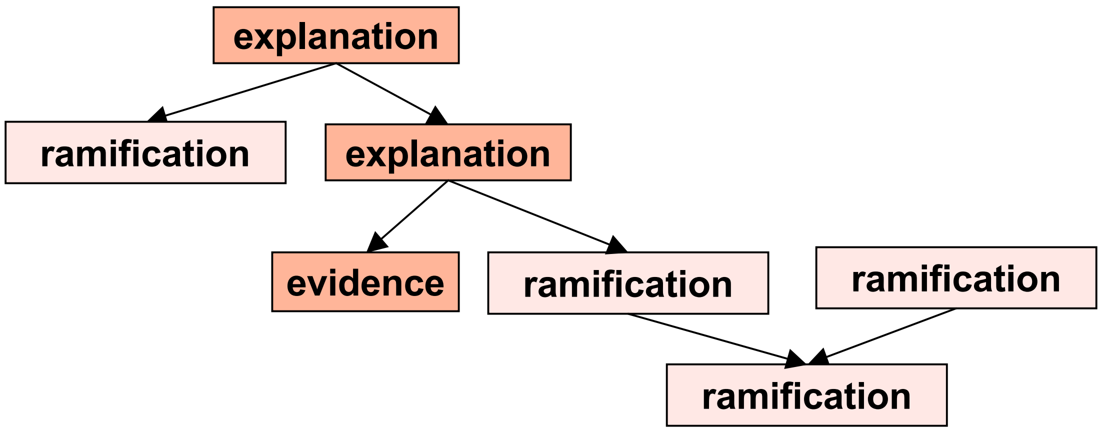
Scene Interpretation by Abduction

Shanahan, M. (2005): Perception as abduction: Turning sensor data into meaningful representation. Cognitive Science 29, 104-134

Compute Δ such that $\Sigma \cup \Delta \mapsto \Gamma$ with

- Σ **background knowledge**
- Γ **evidence**
- Δ **explanation**

Abduction focusses on evidence and does not provide additional ramifications.



Abduction in Description Logics (DLs)

- **Abduction has only recently been introduced as a "non-standard inference service" in DLs.**
- **Growing interest in media interpretation for the Semantic Web.**

First implementation in the commercial DL system RacerPro:

Solve $\Sigma \cup \Gamma_1 \cup \Delta \mapsto \Gamma_2$

Σ **= ABox + TBox**
Γ_1 **= facts not needing an explanation**
Γ_2 **= facts needing an explanation**
Δ **= explanation**

TBox for Table-Laying Domain

```
(implies plate dish)
(implies saucer dish)
(implies cup dish)
(implies napkin (or paper cloth))
(equivalent  cover
     (and  configuration
        (exactly  1  has-plate     plate)
        (exactly  1  has-saucer  (and saucer (near plate)))
        (exactly  1  has-cup       (and cup (on saucer)))
        (atmost 1 has-napkin   napkin)))
        (same-as   has-saucer o near   has-cup)
```

```
(X Y near)   <=    (and  (Z cover)
                         (Z X has-plate)(X plate)
                         (Z Y has-saucer)(Y saucer))
(X Y on)     <=    (and  (Z cover)
                         (Z X has-cup)(X cup)
                         (Z Y has-saucer)(Y saucer))
```

DL-safe rules for representing constraints

Providing Rules for Explanations

```
(equivalent  cover
        (and  configuration
            (exactly  1  has-plate    plate)
            (exactly  1  has-saucer saucer)
            (exactly  1  has-cup      cup)
            (atmost 1 has-napkin   napkin)))
```

⇓ automatic conversion of all conjuncts of an aggregate definition

```
(X configuration)  <=   (and  (X cover)(X configuration)
(Y plate)          <=   (and  (X cover)
                              (X Y  has-plate)(Y plate)
(Y saucer)         <=   (and  (X cover)
                              (X Y has-saucer)(Y cup)
(Y cup)            <=   (and  (X cover)
                              (X Y has-cup)(Y cup)
```

DL-safe rules to allow abduction by backward-chaining

Abduction Example

Calling compute_explanations(Σ, Γ_1, Γ_2) in RacerPro for the table-laying knowledge base:

Γ_2 = {(plate1 plate)(saucer1 saucer)(plate1 saucer1 near)}

$\Downarrow$

Δ = {(cover1 cover)(cover1 plate1 has-plate)(cover1 saucer1 has-saucer)}

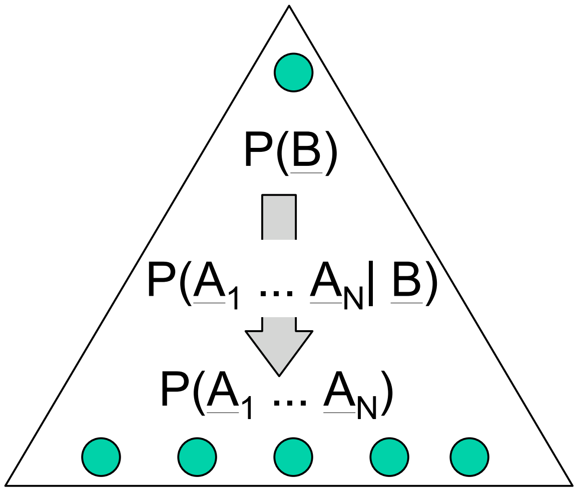
Interpretation Issues Left Open by Logical Framework

- **Task-dependent scope and abstraction level**
  - no need for checking all predicates
  e.g. propositions outside a space and time frame may be uninteresting
  - no need for maximal specialization
  e.g. geometrical shape of "thing" suffices for obstacle avoidance
- **Ambiguous choices for interpretation steps**
  - evidence classfication is naturally ambiguous
  - bad choices may cause inconsistency and backtracking
- **Real-world agents need single "best" scene interpretation**
  - requires uncertainty rating for evidence and context (propositions)
  - requires preference measure for scene interpretations

➡ **Logical model property provides only loose frame for possible scene interpretations.**

Probabilistic Aggregate Structure

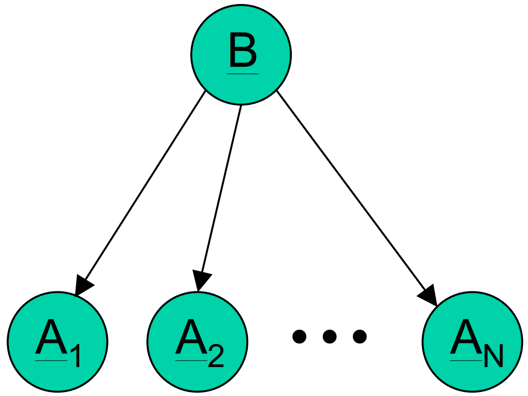
external representation in terms of aggregate properties



internal representation in terms of component properties

Rimey 93:

Tree-shaped part-of nets, is-a trees, expected-area nets, and task nets



unrealistic conditional independence:

$$P(\underline{A}_1 \ldots \underline{A}_N | \underline{B}) = P(\underline{A}_1|\underline{B})\, P(\underline{A}_2|\underline{B}) \ldots P(\underline{A}_N|\underline{B})$$

Probabilities in an Abstraction Hierarchy

Conditional-independence requirements for a compositional hierarchy to be an "abstraction hierarchy":

- ***Aggregate properties do not depend on details below the part properties.***
- ***Part properties depend only on the properties of the corresponding mother aggregate.***
- ***Parts of different aggregates are statistically independent given their mother aggregates.***

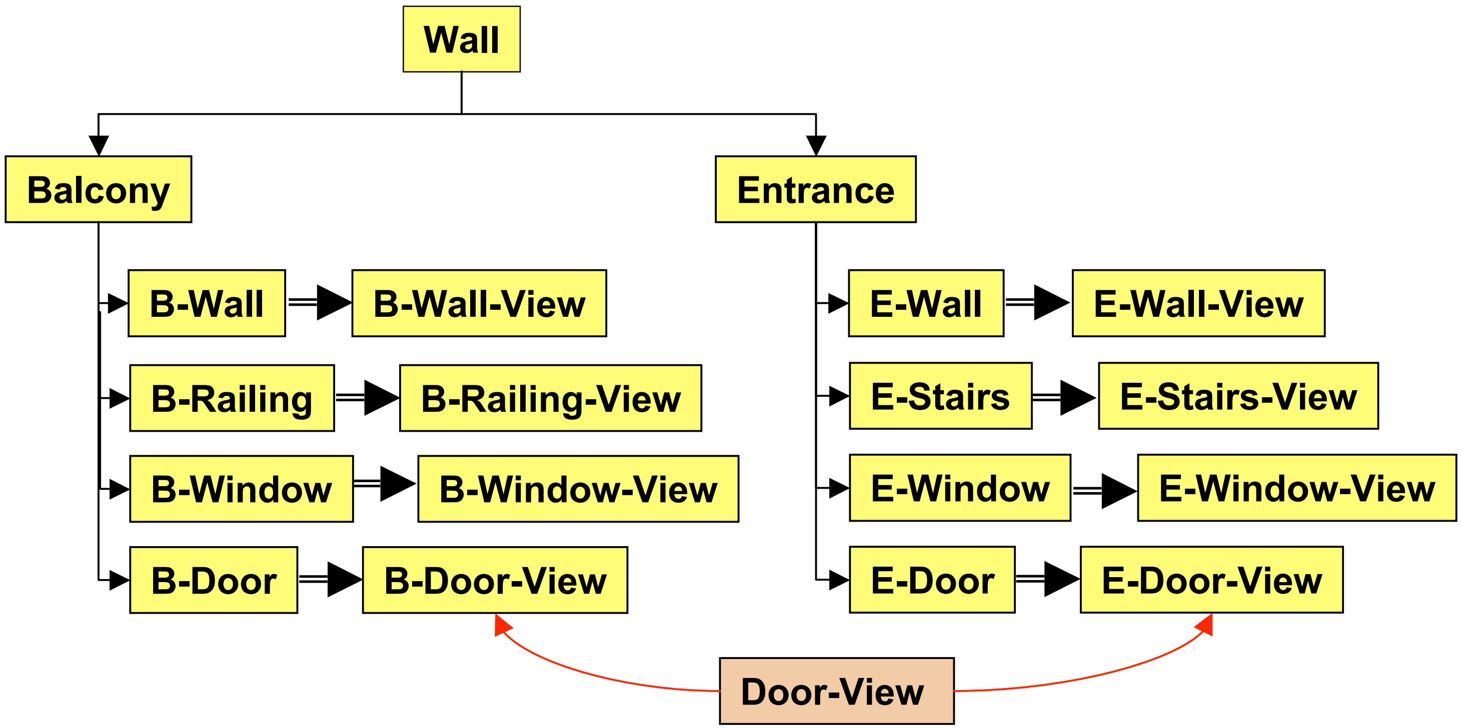
$$P(\underline{Z}_0 \,..\, \underline{Z}_M) = P(\underline{Z}_0) \prod_{i=0 \dots M} P(\text{parts}(\underline{Z}_i) \mid \underline{Z}_i)$$

Z_0 is a node and Z_i, $i = 1 \,..\, M$ are its successors.

The complete JPD of an abstraction hierarchy can be computed from the conditional aggregate JPDs.

Preference Computation

- Probabilities within a branch may be compared without considering the rest of the compositional hierarchy
- Probability updating can be performed with a simple propagation procedure between aggregates



Summary

- Recognising aggregates is the main task in high-level scene interpretation
- Useful inferences can be obtained by stepwise navigation in the aggregate hierarchy
- Partial model construction and abduction provide a logical basis for scene interpretation
- Abduction is available as a non-standard inference service in an optimised DL system
- A probabilistic preference measure can be combined with the logical framework