HIGH-LEVEL VISION

Slides for the High-level Vision Course at the ECVision Summerschool on Cognitive Vision August 2003

Bernd Neumann

Cognitive Systems Laboratory
Hamburg University
Germany

neumann@informatik.uni-hamburg.de
http://kogs-www.informatik.uni-hamburg.de

Sections

What is High-level Scene Interpretation?

Some Application Scenarios for High-level Scene Interpretation

- street traffic observations (long history)
- cameras monitoring parking lots, railway platforms, supermarkets, nuclear power plants, ...
- video archiving and retrieval
- soccer commentator
- smart room cameras
- autonomous robot applications
  (eg robot watchmen, playmate for children)

Characteristics of High-level Scene Interpretation Tasks

- interpretations typically involve several interrelated objects
- spatial and temporal relations are important
- interpretations may build on common sense knowledge
- application scenarios are highly diverse
- domains may be very large
- learning and adaptation may be required
- reliability and complexity management may become important issues
- economical application development requires generic approach

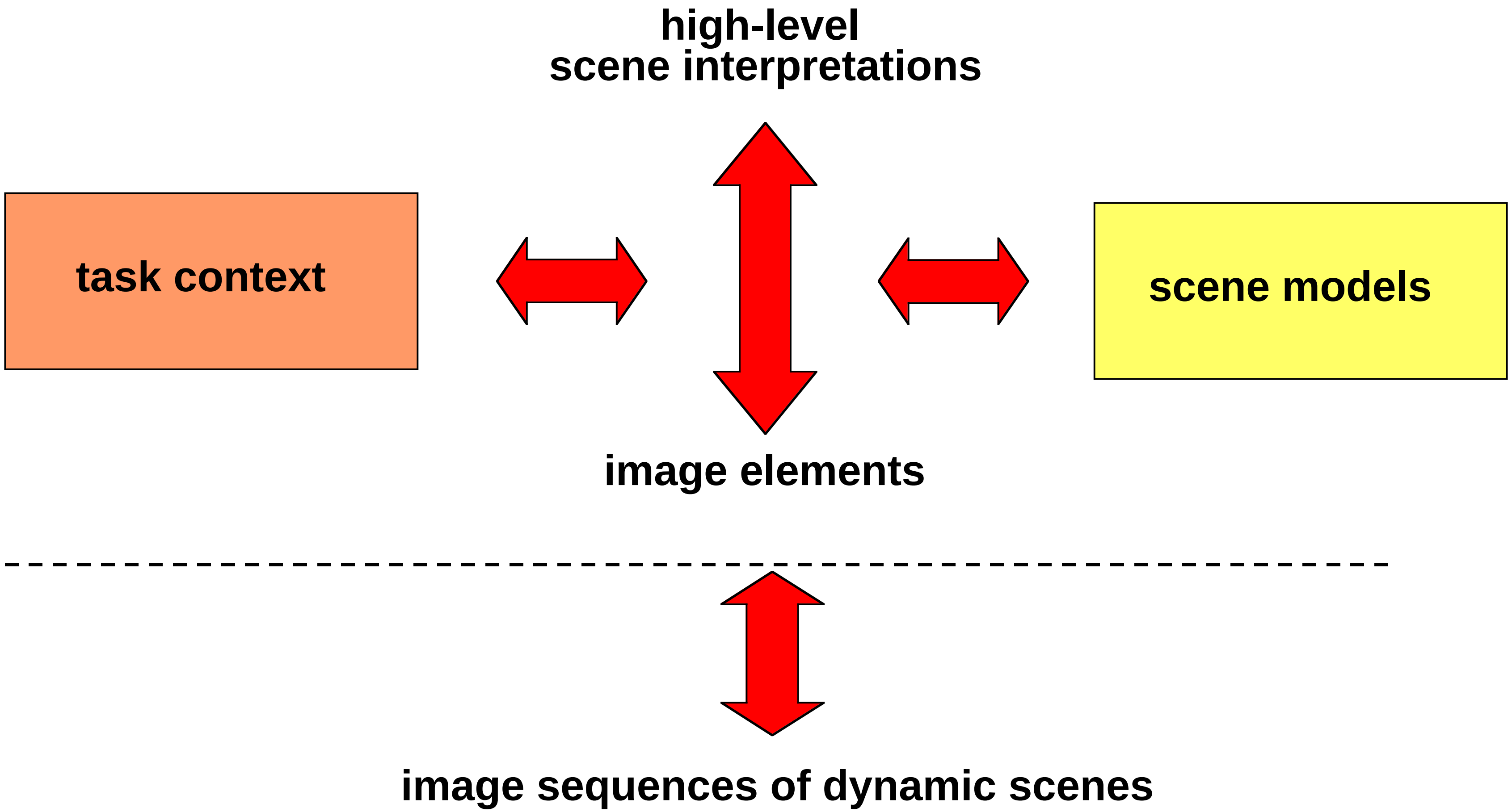
Context and Task Dependence

Interpretations may depend on

- **domain context**
- **spatial context**
- **temporal context**
- **intentional context**
- **task context**
- **communicative context**
- **focus of attention**
- **a priori probabilities**

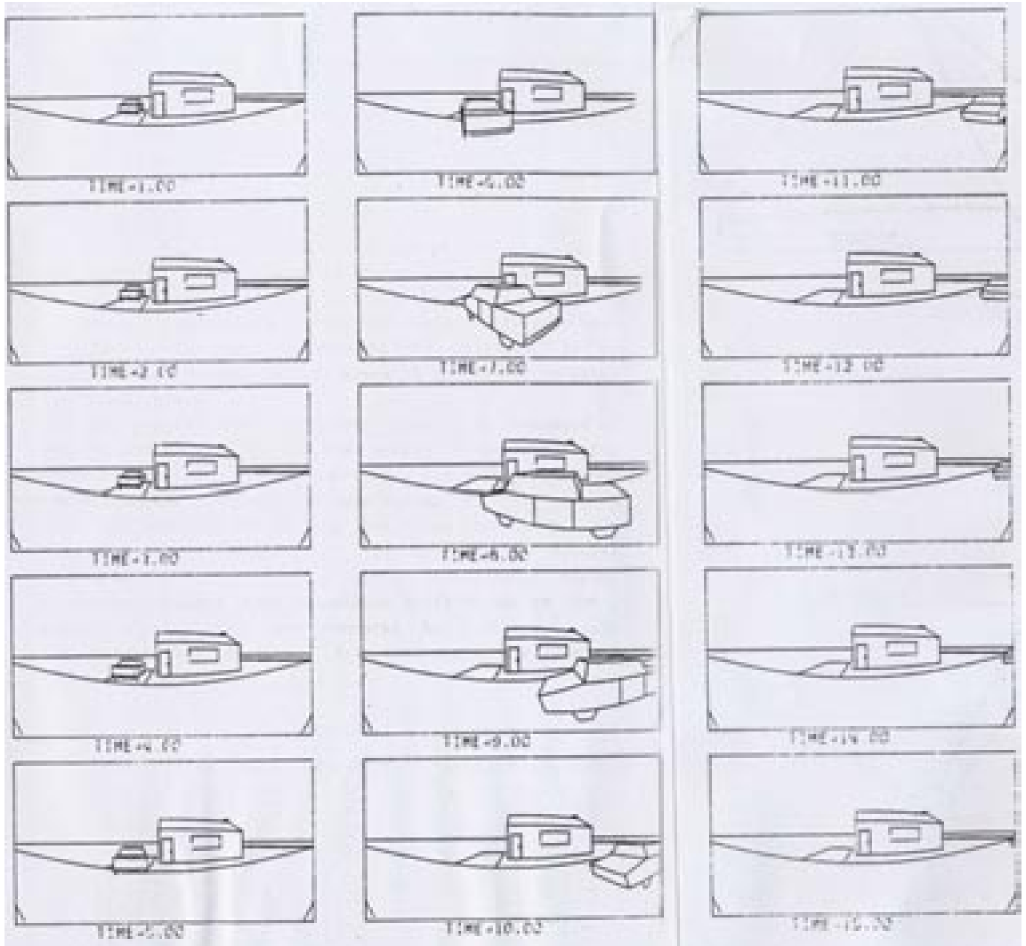
Constructing an interpretation is <u>not</u> a mapping from image data into interpretation space.

Basic Structure of High-level Scene Interpretation



Historical Examples

Early Traffic Scene Analysis (Badler 75)



15 "snapshots" of a car leaving the driveway of a house

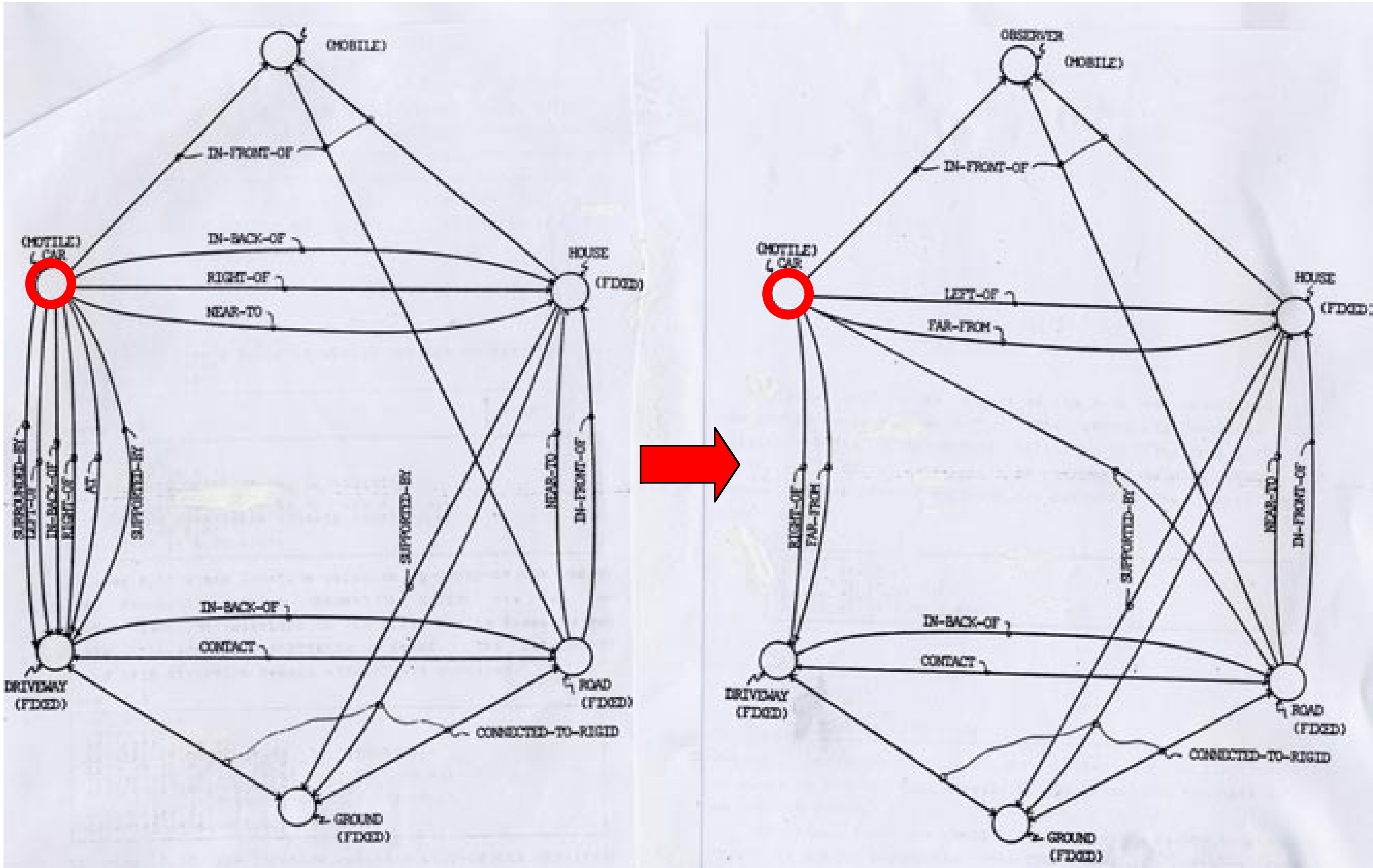
Directional Adverbials for Motion Description (Badler 75)

ACROSS
AFTER
AGAINST
AHEAD-OF
ALONG
APART
AROUND
AWAY
AWAY-FROM
BACK
BACK-AND-FORTH
BACKWARD
BEHIND
BY

CLOCKWISE
COUNTERCLOCKWISE
DOWN
FORWARD
FROM
IN
IN-THE-DIRECTION-OF
INTO
INWARD
OFF
OFF-OF
ON
ONTO
ONWARD

OUT
OUT-OF
OUTWARD
OVER
SIDEWAYS
THROUGH
TO
TO-AND-FRO
TOGETHER
TOWARD
UNDER
UP
UP-AND-DOWN
UPWARD
WITH

Changing Scene Graph for Car Scene (Badler 75)



Demon Representation of "ACROSS" Motion (Badler 75)

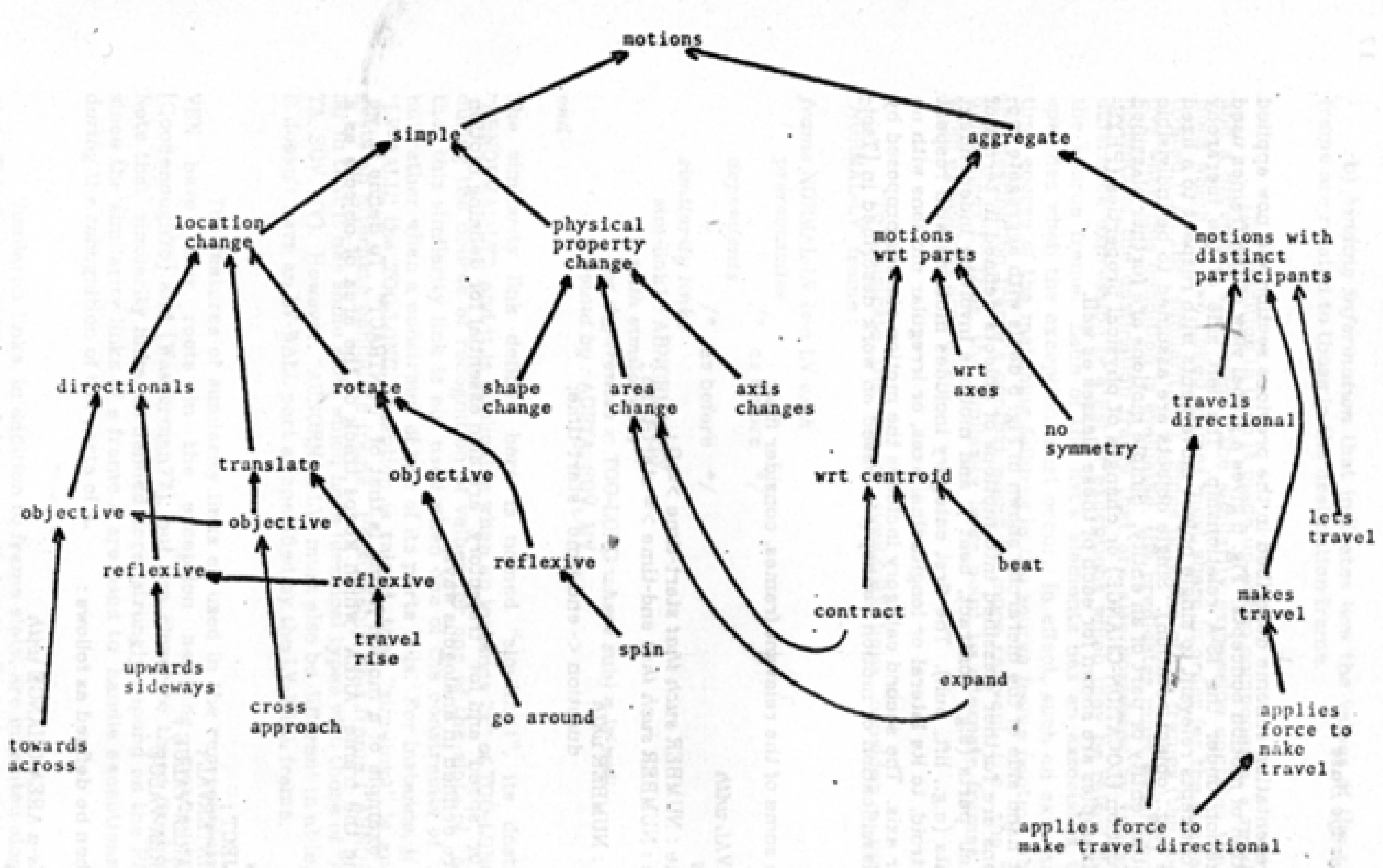
A NEAR-TO relation with one side of an object is broken and replaced by a similar relation with the other side. There is an implicit sense of passage ABOVE the object.

Precondition 1
NEAR-TO(X S1).
SUB-PART(Y S1) for some object Y and SUB-PART [chain] to object S1.
FRONT or BACK or LEFT-SIDE or RIGHT-SIDE(Y S1).
ACROSS remains active as long as NEAR-TO(X Y) and A BOVE(X Y) hold.

Precondition 2
NEAR-TO(X S2).
SUB-PART(Y S2) for a SUB-PART [chain] to object S2.
FRONT or BACK or LEFT-SIDE or RIGHT-SIDE(Y S2) where S1 ≠ S2 and at least one of the ORIENTATION relations to S1 (from Precondition 1) no longer holds.

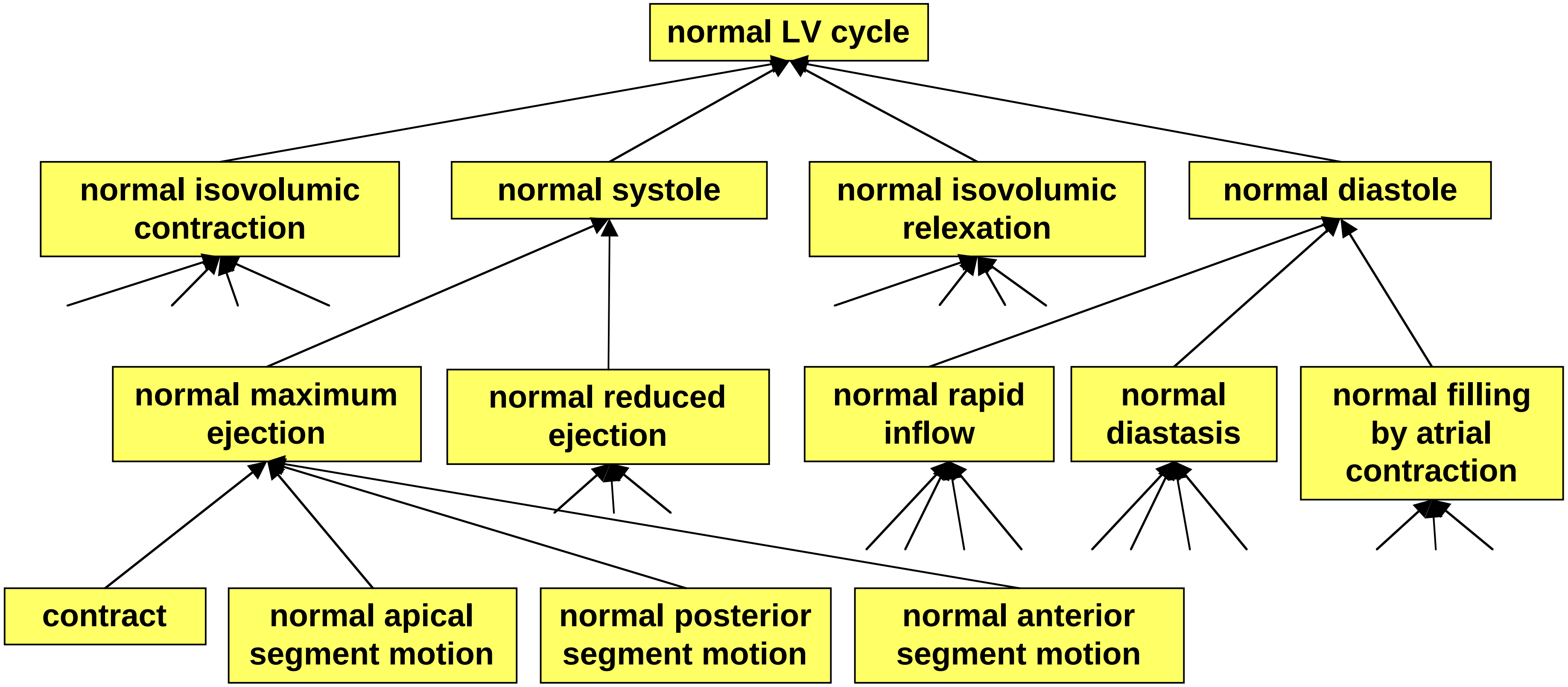
Postcondition
SUBJECT X
DIRECTION PCONS((ACROSS Y), DIRECTION)

Motion IS-A Hierarchy (Tsotsos 79)



Left-ventricular Motion PART-OF Hierarchy (Tsotsos 79)

PART-OF structure supports part-whole reasoning in recognition processes

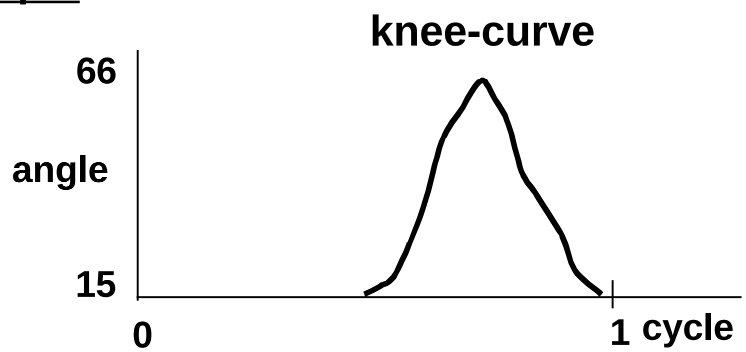


Model-based Prediction for Tracking a Jointed Moving Object (Hogg 84)

The case of highly coordinated motion of parts

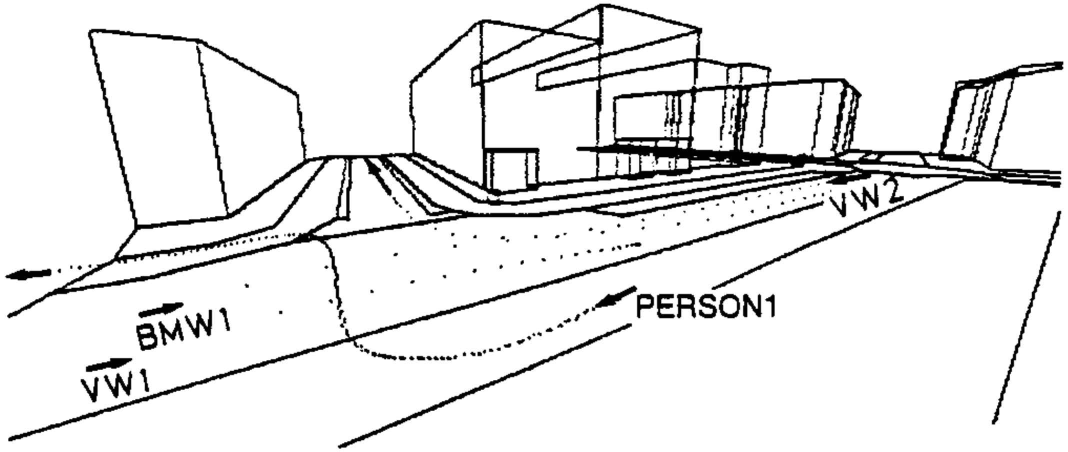
Posture curves + constraints represent coordinated motion of joints of walker.

Example:



NAOS - Natural Language Description of Object Motions in Traffic Scenes

(Neumann & Novak 1986)



English paraphrase of automatically generated description:

The scene contains four moving objects: three cars and a pedestrian.

A VW drives from the Alte-Post to the front of the FBI. It stops.

Another VW drives towards Dammtor. It turns off Schlueterstrasse. It drives on Bieberstrasse towards Grindelhof.

A BMW drives towards Hallerplatz. While doing so, it overtakes the VW which has stopped, before Bieberstrasse. The BMW stops in front of the traffic lights.

The pedestrian walks towards Dammtor. While doing so, he crosses Schlueterstrasse in front of the FBI.

From Scene Data to a Natural-language Scene Description (NAOS)

natural-language scene description

↑

case frames

↑

occurrences

↑

primitive occurrences

↑

perceptual primitives

↑

geometrical scene description (GSD)

Geometrical Scene Description (GSD) in NAOS

Quantitative description of all objects in a time-varying scene:

- **name of all objects (class or identity)**
- **position of all objects at all times (location and orientation)**
- **illumination (if required for high-level description)**

Example of a synthesized GSD in NAOS:

```
(LAGE VW2 (779. 170. 0.) (-1.0 0.0 0.0) 0)
(LAGE VW2 (753. 170. 0.) (-1.0 0.0 0.0) 1)
(LAGE VW2 (727. 170. 0.) (-1.0 0.0 0.0) 2)
(LAGE VW2 (701. 170. 0.) (-1.0 0.0 0.0) 3)
(LAGE VW2 (675. 170. 0.) (-1.0 0.0 0.0) 4)
(LAGE VW2 (649. 170. 0.) (-1.0 0.0 0.0) 5)
(LAGE VW2 (623. 170. 0.) (-0.999 0.037 0.0) 6)
(LAGE VW2 (596. 171. 0.) (-1.0 0.0 0.0) 7)
(LAGE VW2 (570. 171. 0.) (-1.0 0.0 0.0) 8)
(LAGE VW2 (544. 171. 0.) (-1.0 0.0 0.0) 9)
•
•
•
```

location   orientation   time

Occurrence Model for "OVERTAKE" (NAOS)

(OVERTAKE OBJ1 OBJ2 T1 T2) <=>
    (MOVE OBJ1 T1 T2)
    (MOVE OBJ2 T1 T2)
    (BEHIND OBJ1 OBJ2 T1 T3)
    (BESIDE OBJ1 OBJ2 T3 T4)
    (BEFORE OBJ1 OBJ2 T4 T2)
    (APPROACH OBJ1 OBJ2 T1 T3)
    (DIS-APPROACH OBJ1 OBJ2 T4 T2)

- temporal constraint satisfaction for occurrence recognition
- principled definition of primitive occurrences

Temporal Relations in NAOS

- Observations provide begin and end time points of occurrences
- Models express qualitative constraints on time points

Unary temporal constraints: $t_{min} \leq t \leq t_{max}$

Binary temporal constraints: $t_1 \geq t_2 + c_{12}$

Convex interval relations may be expressed by inequalities:

$$I_1 \text{ during } I_2 \quad \Rightarrow \quad \begin{array}{l} I_2.tb \leq I_1.tb \\ I_1.te \leq I_2.te \end{array}$$

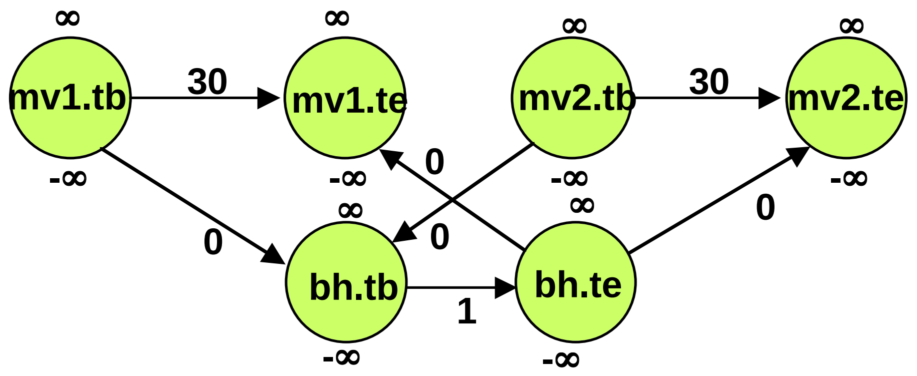
NAOS temporal constraint propagation was later identified as a convex time point algebra [Vila 94].

Constraint Propagation for Occurrence Verification (1)

Example:

Verify occurrence "two moving objects, one behind the other"

1. **Initialize constraint net of occurrence model**



2. **Compute primitive events for scene**

ID:	move1
instance:	move
parts:	mv-ob = obj1
	mv-tr = trj1
times:	mv-tb = 13
	mv-te = 47

ID:	behind1
instance:	behind
parts:	bh-ob1 = obj1
	bh-obj2 = obj2
times:	bh-tb = 20
	bh-te = 33

(and many more)

Constraint Propagation for Occurrence Verification (2)

3. **Instantiate parts in occurrence model**

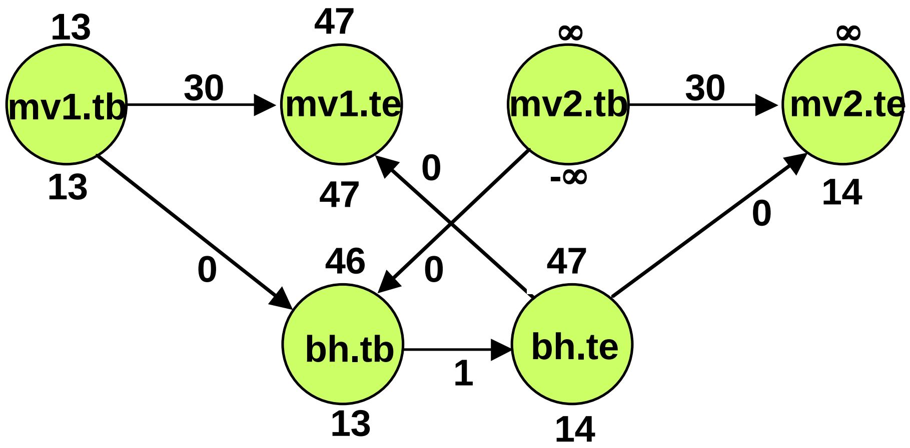
propagate minima and maxima of time points through constraint net:

- minima in edge direction $t_{2min}' = \max \{t_{2min}, t_{1min} + c_{12}\}$

- maxima against edge direction $t_{1max}' = \min \{t_{1max}, t_{2max} - c_{12}\}$

<u>Example</u>: move1 in scene instantiates mv1 of model

ID:	move1
instance:	move
parts:	mv-ob = obj1
	mv-tr = trj1
times:	mv-tb = 13
	mv-te = 47



Constraint Propagation for Occurrence Verification (3)

4. Consistency and completeness test

A (partially) instantiated model is inconsistent, if for any node T one has: Tmin > Tmax

=> search for alternative instantiations or terminate with failure

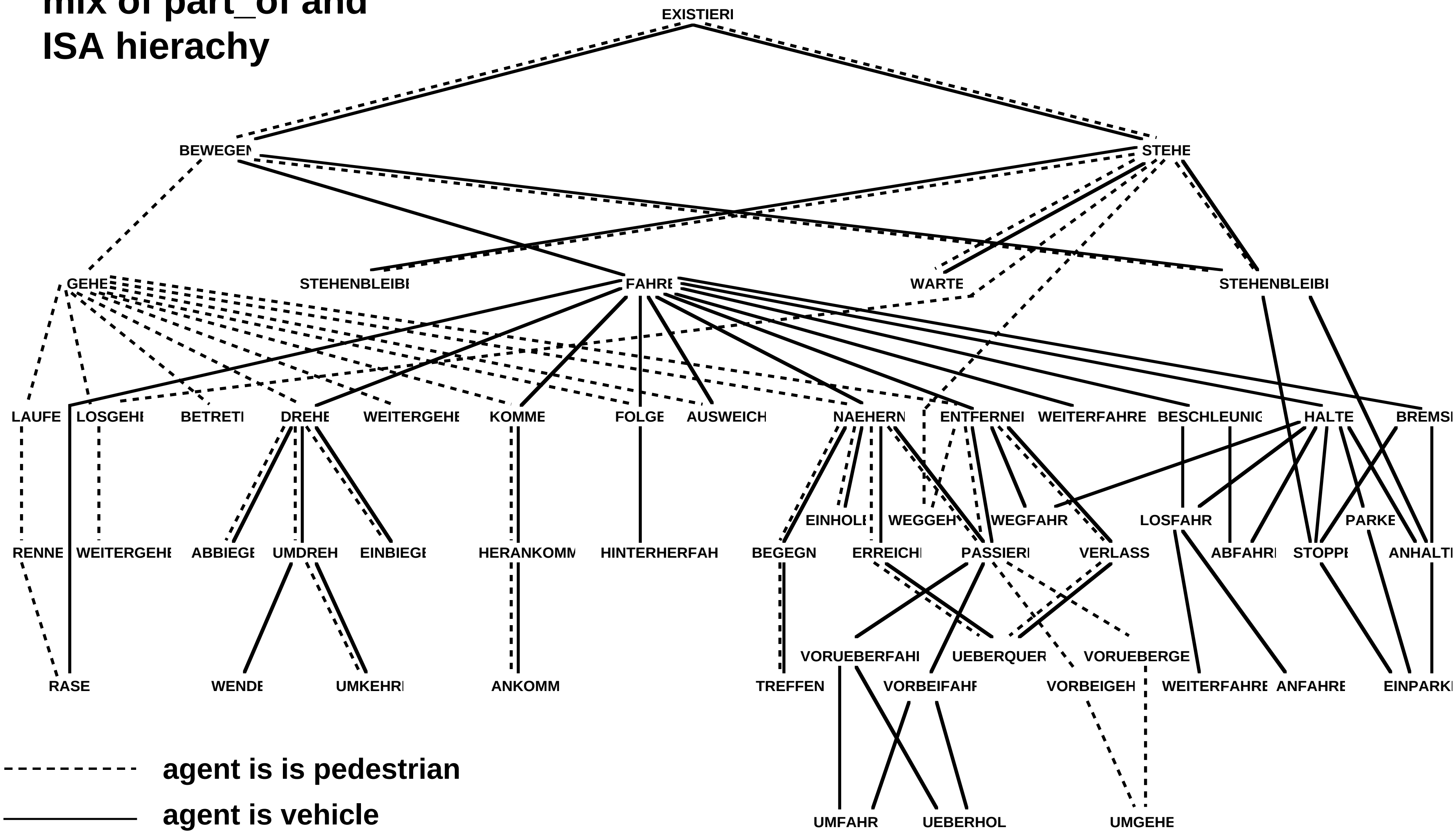
An occurrence has been recognized if the occurrence model is instantiated with sufficient completeness and the instantiation is consistent.

Note:

- **<u>Incremental</u> occurrence recognition follows an evolving scene**
- **<u>A-posteriori</u> occurrence recognition is carried out after observing a scene (choice of order!)**
- **Partially instantiated models may be used for scene <u>prediction</u>**

Hierachy for Object Motions in Street Traffic (NAOS)

mix of part_of and ISA hierachy



- - - - - - - - agent is is pedestrian

———————— agent is vehicle

Generating a Natural-language Description

Principle:

sentence

↑

case frame

↑

occurrence

language-oriented AI techniques

Problems:

- Which occurrences should be selected for verbalization?
- Which deep cases should be filled?
- Which addtitional time or location information is required?
- In which order should the information be presented?

Solution:

Speech planning based on hearer simulation

informing a hearer <=> enabling a hearer to imagine the scene

Standard Plan for Generating Natural-language Scene Descriptions in NAOS

- rules which assure that the hearer will be able to <u>imagine</u> the scene
- summary + descriptions of all object trajectories, each in chronological order
- no explicit hearer simulation

Description of an object trajectory

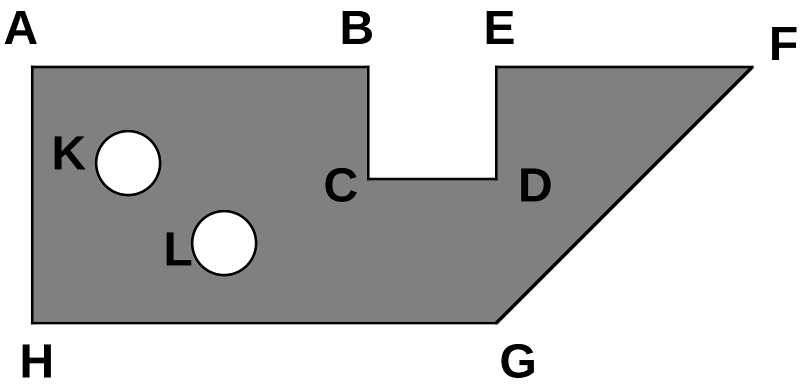
1. Each time interval is described by the <u>most special</u> occurrence
2. The first occurrence begins at the beginning of the scene
3. The next occurrence follows in temporal order
4. Location information is given by prepositional expressions as required
5. Temporal information is given by prepositional expressions or references to other occurrences as required

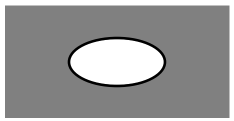
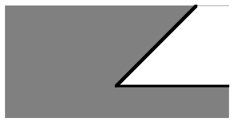
Relational Matching

Example of a Relational Model for Object Recognition (1)

(Bolles & Cain 83)

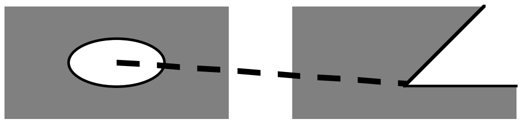
shape to be recognized:



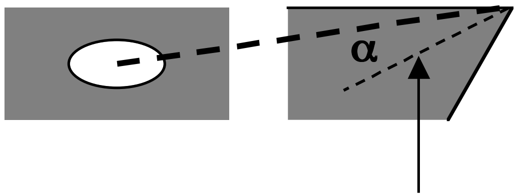
primitive descriptive elements (nodes)		properties	
	hole	t f a	type T1 area axes relation
	interior corner	t w	type T2 angle
	exterior corner	t w	type T3 angle

Example of a Relational Model for Object Recognition (2)

relations between primitive descriptive elements (edges)



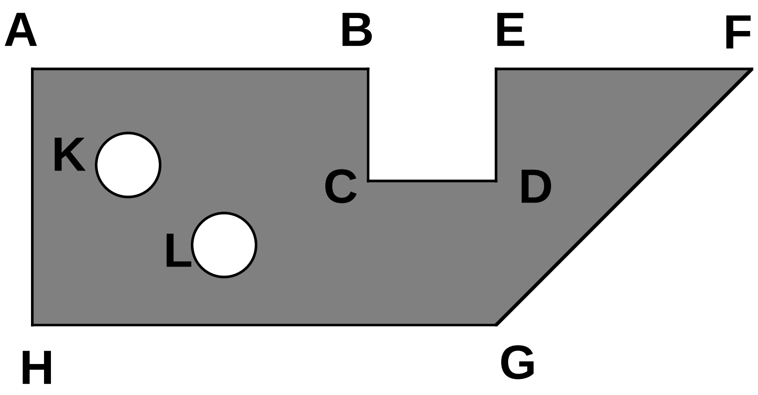
...
d10 distance 10 ± 1
d12 distance 12 ± 1
d14 distance 14 ± 1
...



bisector of angle

...
o10 orientation 10 ± 5
o20 orientation 20 ± 5
o30 orientation 30 ± 5
...

Example of a Relational Model for Object Recognition (3)



A	t	T3
	w	90

B	t	T3
	w	90

C	t	T2
	w	90

D	t	T2
	w	90

E	t	T3
	w	90

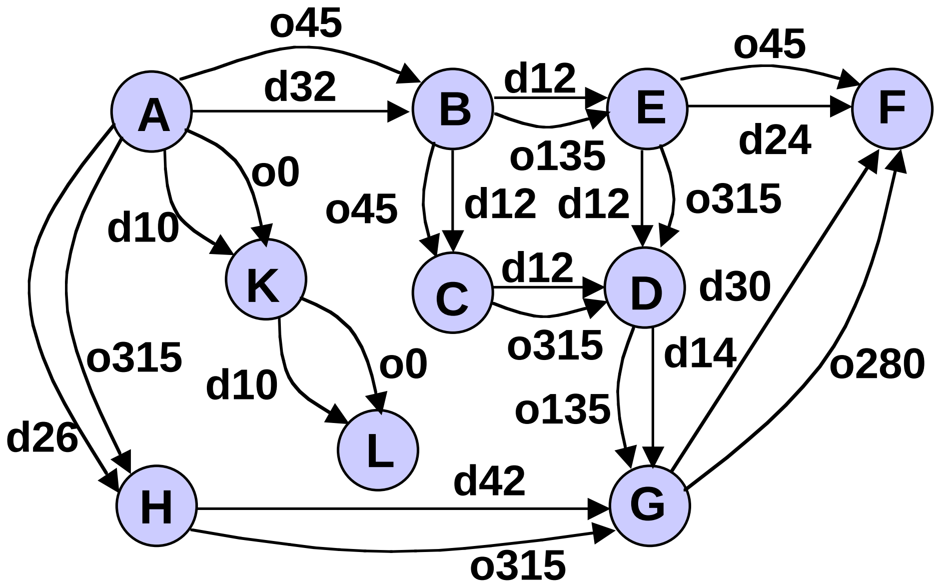
F	t	T3
	w	45

G	t	T3
	w	135

H	t	T3
	w	90

K	t	T1
	f	48
	a	1

K	t	T1
	f	48
	a	1



(not all edges are shown)

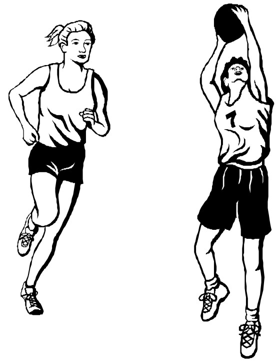
Relational Models for High-level Vision

Relational models describe objects (object classes) based on parts (components) and relations between the parts

A relational model can be represented as a structure with nodes and edges:

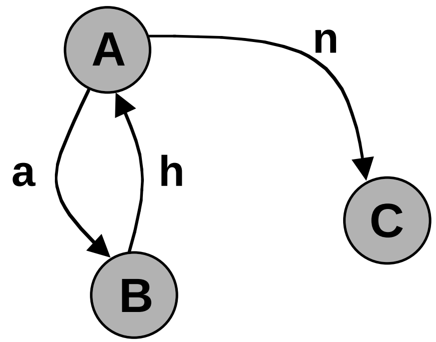
Nodes: parts with properties

A	B	C
is-a person	is-a person	is-a ball
state running	state jumping	colour black



Edges: relations between parts

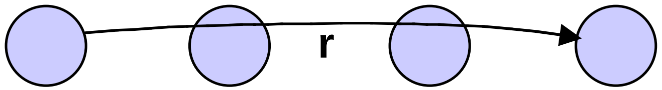
- approaches A B
- nearby B A
- holds B C



Representing N-ary Relations

Awkward graphical representation:

"hypergraph"



Reification:

(BETWEEN A B C) → **(INSTANCE BETW1 BETWEEN)**
(BETWEEN-ARG1 BETW1 A)
(BETWEEN-ARG2 BETW1 B)
(BETWEEN-ARG3 BETW1 C)

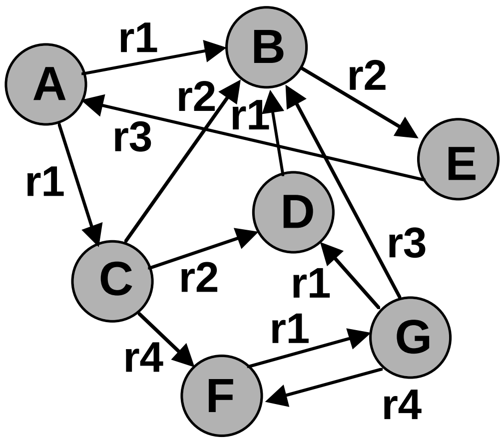
(OVERTAKE VEH1 VEH2 23 46) → **(INSTANCE OT1 OVERTAKE)**
(OVERTAKER OT1 VEH1)
(OVERTAKEE OT1 VEH2)
(TBEG OT1 23)
(TEND OT1 42)

Recognition by Relational Matching

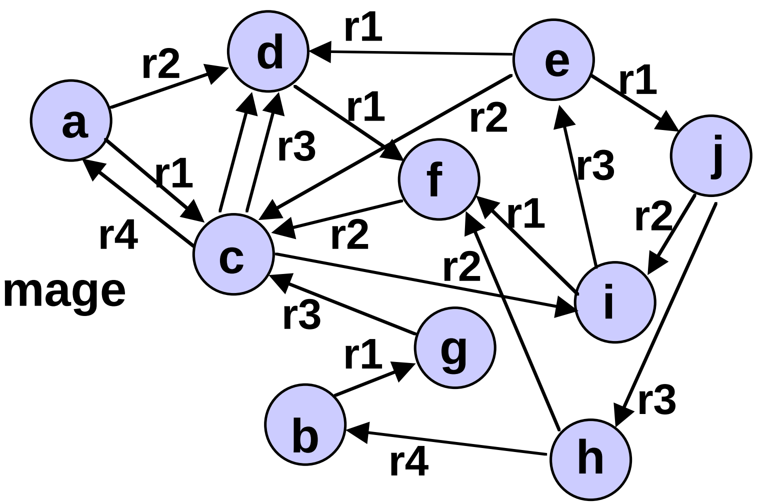
Principle:

- **construct relational model(s) for object class(es)**
- **construct relational image description**
- **compute morphism (best partial match) between image and model(s)**

model



image



Compatibility of Relational Structures

Different from graphs, nodes and edges of relational structures may represent entities with rich distinctive descriptions.

Example: **nodes = image regions with diverse properties**
edges = spatial relations

1. Compatibility of nodes

An image node is compatible with a model node, if the properties of the nodes match.

2. Compatibility of edges

An image edge is compatible with a model edge, if the edge types match.

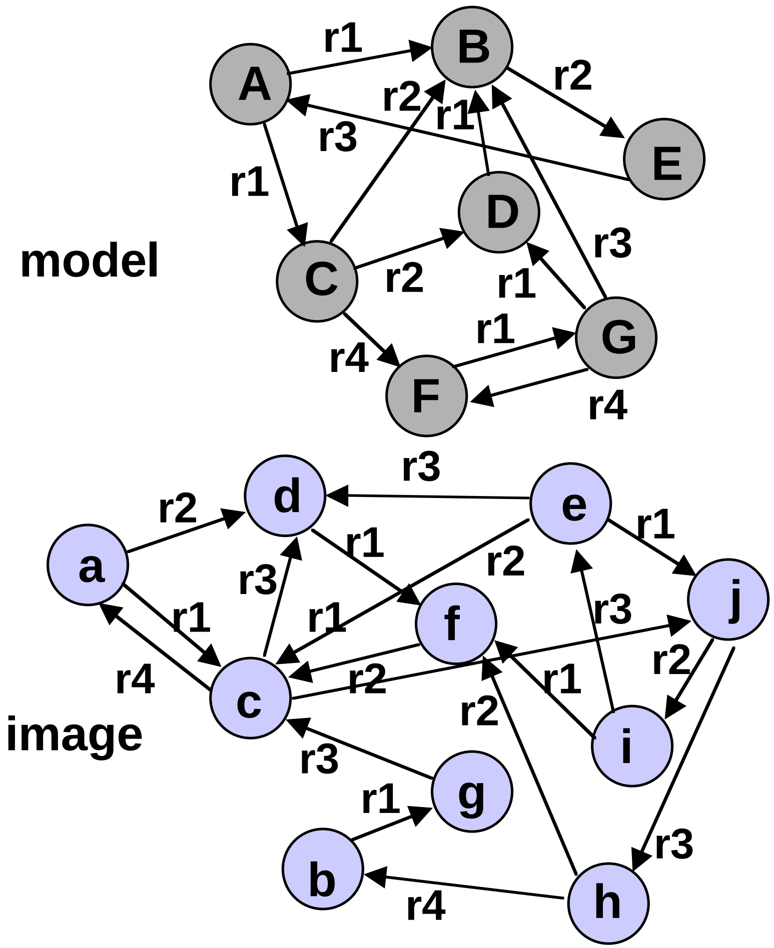
3. Compatibility of structures

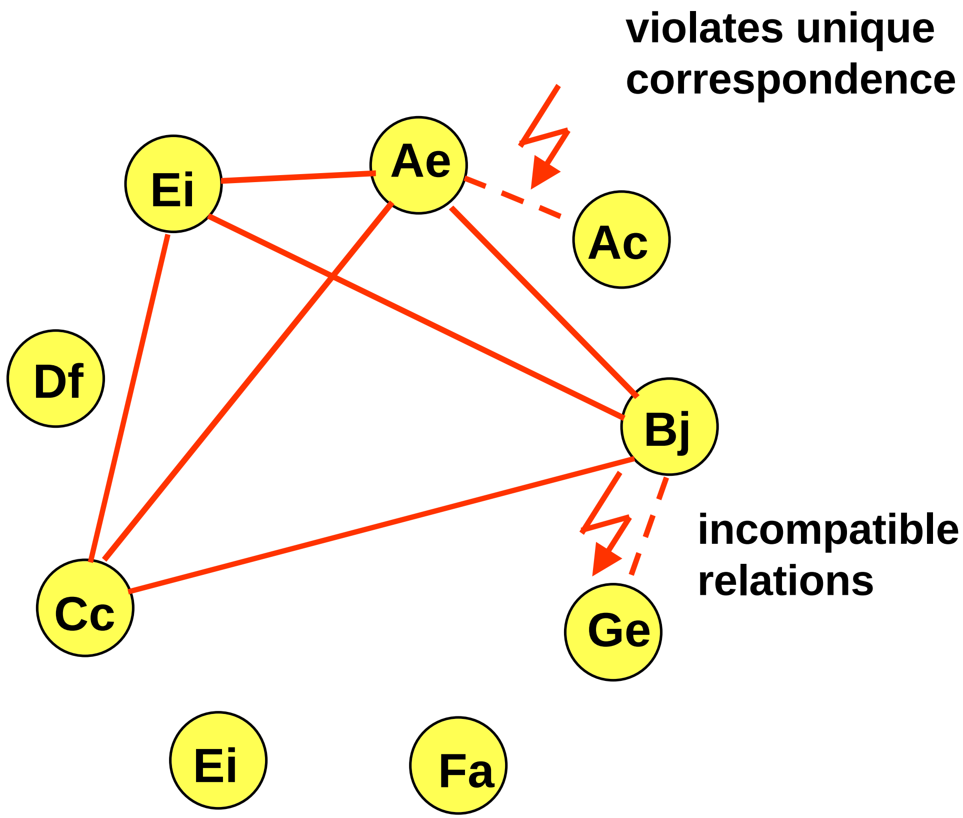
A relational image description B is compatible with a relational model M, if there exists a bijective mapping of nodes of a partial structure B´of B onto nodes of a partial structure M´of M such that

- corresponding nodes and edges are compatible
- M is described by M´ with sufficient completeness

Relational Matching Using a Compatibility Graph

nodes of compatibility graph = pairs with compatible properties
edges of compatibility graph = compatible pairs
cliques in compatibility graph = compatible partial structures





compatibility graph
(not shown completely)

Finding Maximal Cliques

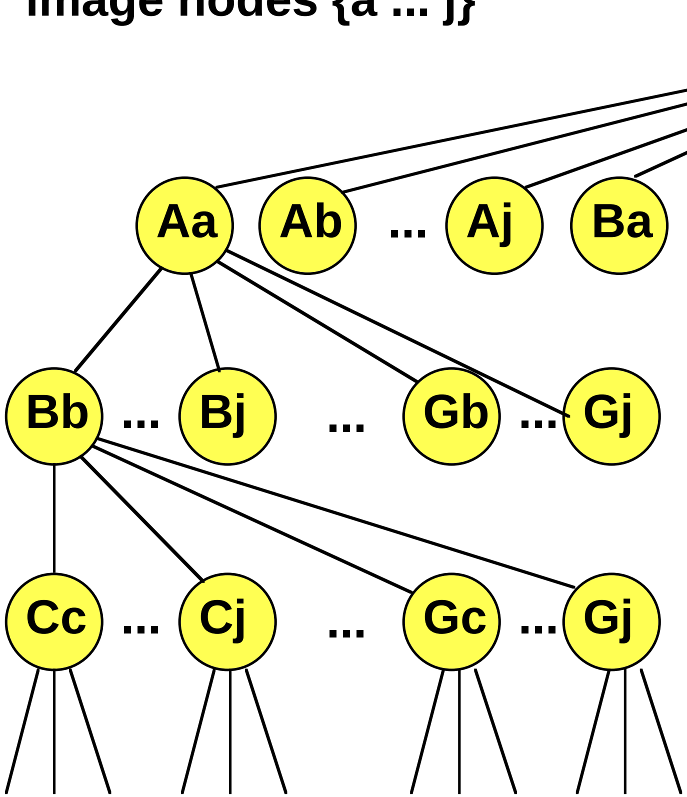
Algorithms are available in the literature, e.g.

Bron & Kerbusch, Finding all Cliques of an Undirected Graph, Communications of the ACM, Vol. 16, Nr. 9, S. 575 - 577, 1973.

- **Complexity is exponential relative to number of nodes of compatibility graph**
- **Efficient (suboptimal) solutions based on heuristic search**

Relational Matching with Heuristic Search

Stepwise correspondence search between model nodes {A ... G} and image nodes {a ... j}



- quality function evaluates partial matches
- accept a partial match if quality > acceptance threshold
- refute a partial match, if quality < refutation threshold

Shortcomings of Relational Matching for High-level Scene Interpretation (1)

Natural hierarchical structures and groupings are not well represented by flat relational structures

<u>Example</u>: Modelling dining room views

- **room**
  - **door**
  - **cupboard**
  - **table-group**
    - **table-top**
      - **dishes**
      - **candles**
      - **sets**
        - **plate**
        - **cutlery-group**
          - **knife**
          - **fork**
        - **cup-group**
          - **cup**
          - **saucer**
    - **chairs**
  - **lamp**

In a model, repeated identical structures should only be represented once

Shortcomings of Relational Matching for High-level Scene Interpretation (2)

Node compatibility is not clearly defined

model	compatible?	image
A is-a person size tall	⟷	a is-a man size 198 state running

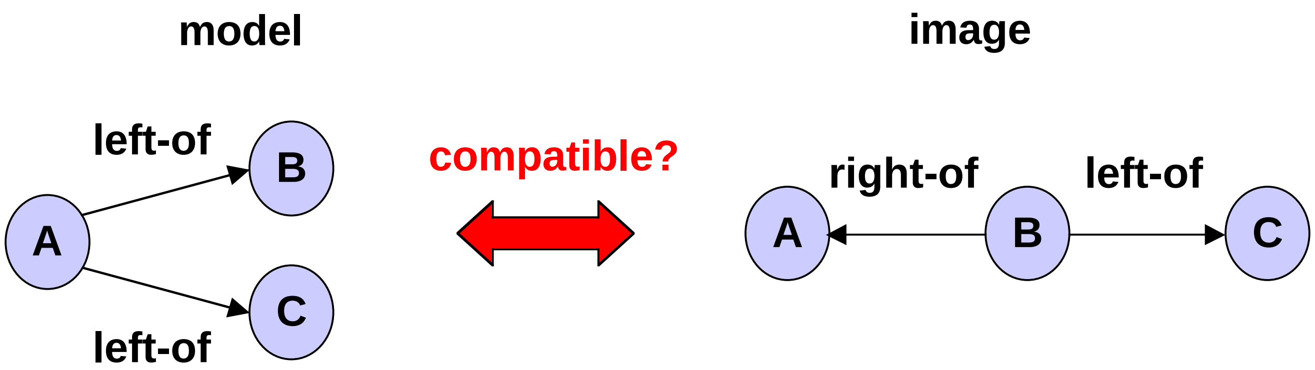
Edge compatibility is not clearly defined

$A \xrightarrow{\text{nearby}} B$ ⟷ compatible? ⟷ $a \xrightarrow{\text{touch}} b$

Logical relations between different node descriptions and different edge labels must be clarified

Shortcomings of Relational Matching for High-level Scene Interpretation (3)

Implicit information is not considered



Reasoning may be required to determine compatibility

How Useful is Relational Matching?

- **relational structure captures basic high-level notions**
- **graceful degradation w.r.t. completeness and degree of match**
- **well-understood computional procedures**
  - finding maximal cliques in compatibility graphs
  - heuristic search
  - constraint satisfaction
  - neural network implementations
- **improvement by hierarchical matching**

- **multi-level aggregate structure required**
- **differentiated compatibility measure required**
  - fuzziness
  - compatibilty vs. consistency
  - probabilities
- **reasoning about temporal, spatial, physical relations**
- **uncertainty management required**

Rule-based Interpretation

Rule Systems

Rule systems provide

- **user specified inference steps**
- **an inference engine which applies rules to a database**
- **inference strategies which determine the order of rule applications**

In principle, rule systems may provide inferencing capabilities needed for relational matching, e.g.

(and (right-of X Y) (right-of Y Z)) => (right-of X Z)

The usability of rule systems is limited, however, because of

- **liberal rule specifications at the users discretion**
- **lack of control over rule applications (data-driven paradigm)**
- **no guarantees for logical correctness or completeness**
- **lack of higher-level structures for data and rules**

Rule System OPS5

OPS5 ("Official Production System, Version 5")
- developed at CMU 1980 ...
- implementation language for successful XPS (XCON, XSEL a.o.)

CLIPS
- reimplementation of OPS5 in C for NASA
- freeware

JESS
- reimplementation of OPS5 in Java
- freeware

Rules in OPS5

Syntax of a rule in OPS5:

<rule>::= [P <rule-name> <antecedent> --> <consequent>]

<antecedent>::= {<condition>}

<condition> ::= <pattern> | - <pattern>

<pattern> ::= [<object> {^<attribute> <value>}]

<consequent> ::= {<action>}

<action> ::= [MAKE <object> {^<attribute> <value>}] |
[MODIFY <pattern-number> {^<attribute> <value>}]
[REMOVE <pattern-number>] |
[WRITE {<value>}]

<u>Example</u>: "If there are 2 disks close to each other and with equal size, make them a wheel pair"

```
[P   find-wheel-pair [disk    ^location    <x1>                    ^size    <y>]
                     [disk    ^location    |<x2> - <x1>| < 10      ^size    <y>]      --> ... ]
```

Variable

- depth-first search
- limited expressiveness for constraints

When is Rule-based Interpretation Feasible?

- **successful applications for restricted domains**
  - recognising airports (McKeown et al. 85)
  - classification of forestry in aerial images (Pinz 85)
  - 2D image analysis
- **problems with degraded images**
- **domain knowledge and control not separated**
  - free choice of interpretation strategy dependent on task and context
  - separation required for complexity management
- **does not scale beyond - say - 1000 rules**

Description Logics

Why a Logic-based Approach?

- exploring a logic-based approach for a task which requires guess-work
- representing conceptual models with well-defined semantics
- exploiting validated inference procedures
- interfacing to common-sense knowledge

Description Logics for Knowledge Representation

Family of knowledge-representation formalisms

- **object-centered, roles and features (binary relations)**
- **necessary vs. sufficient attributes**
- **inference services**
  - subsumption check
  - consistency check
  - classification
  - abstraction
  - default reasoning
  - spatial and temporal reasoning
- **guaranteed correctness, completeness, decidability and complexity properties**
- **highly optimized implementations (e.g. RACER)**

Development of Description Logics

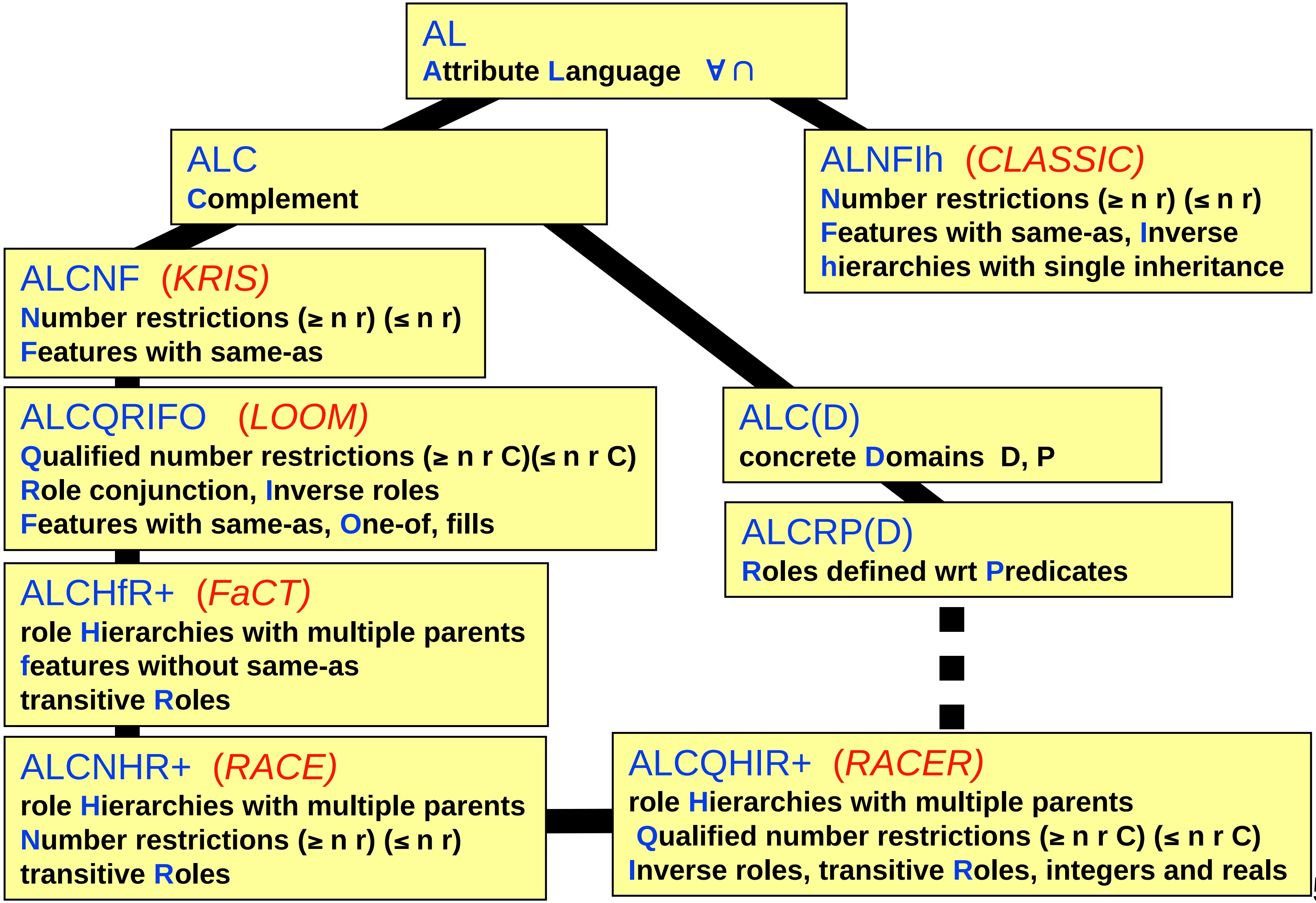
There exist several commercial and experimental developments of DLs, among them

- KL-ONE first conception of a DL (1985)
- CLASSIC commercial implementation by AT&T
- LOOM experimental system at USC
- FaCT experimental and commercial system (Horrocks, Manchester)
- RACER experimental system in Hamburg and Montreal (Haarslev & Moeller)

There is active research on DLs:

- **extending the expressivity of concept languages**
- **decidability and tractability of inference services**
- **integration of predicates over concrete domains (e.g. numbers)**
- **highly optimized implementations**
- **developing new inference services (e.g. for scene interpretation)**

Family of Description Logics



RACER Concept Language

C	concept term
CN	concept name
R	role term
RN	role name

C -> *CN*
top
bottom
(not *C*)
(and *C1* ... *Cn*)
(or *C1* ... *Cn*)
(some *R* *C*)
(all *R* *C*)
(at-least *n* *R*)
(at-most *n* *R*)
(exactly *n* *R*)
(at-least *n* *R* *C*)
(at-most *n* *R* *C*)
(exactly *n* *R* *C*)
CDC

concept definition

(equivalent *CN* *C*)

concept axioms

(implies *CN* *C*)
(implies *C1* *C2*)
(equivalent *C1* *C2*)
(disjoint *C1* ... *Cn*)

roles

R -> *RN*
(inv *RN*)

concrete-domain concepts

AN *attribute name*

CDC -> (a *AN*)
(an *AN*)
(no *AN*)
(min *AN* *integer*)
(max *AN* *integer*)
(> *aexpr* *aexpr*)
(>= *aexpr* *aexpr*)
(< *aexpr* *aexpr*)
(<= *aexpr* *aexpr*)
(= *aexpr* *aexpr*)

aexpr -> *AN*
real
(+ *aexpr1* *aexpr1**)
aexpr1

aexpr1 -> *real*
AN
(* *real* *AN*)

Primitive and Defined Concepts

Concept expressions of a DL describe sets of entities within terms of properties (unary relations) and the roles (binary relations).

The main building blocks are primitive oder defined concepts.

Primitive concepts: **concept => satisfied properties and relations**

satisfied properties and relations are <u>necessary</u> conditions for an object to belong to a class

Defined concepts: **concept <=> satisfied properties and relations**

satisfied properties and relations are <u>necessary and sufficient</u> conditions for an object to belong to a classt

Primitive concept "person":

(implies person (and human (some has-gender (or female male))))

Defined concept "parent":

(equivalent parent (and person (some has-child person)))

Example of a TBox

```
(signature   :atomic-concepts  (person human female male woman man parent
                                 mother father grandmother aunt uncle sister brother)
             :roles    ((has-child :parent has-descendant)
                        (has-descendant :transitive t)
                        (has-sibling)
                        (has-sister :parent has-sibling)
                        (has-brother :parent has-sibling)
                        (has-gender :feature t)))
```

Signature of T-Box

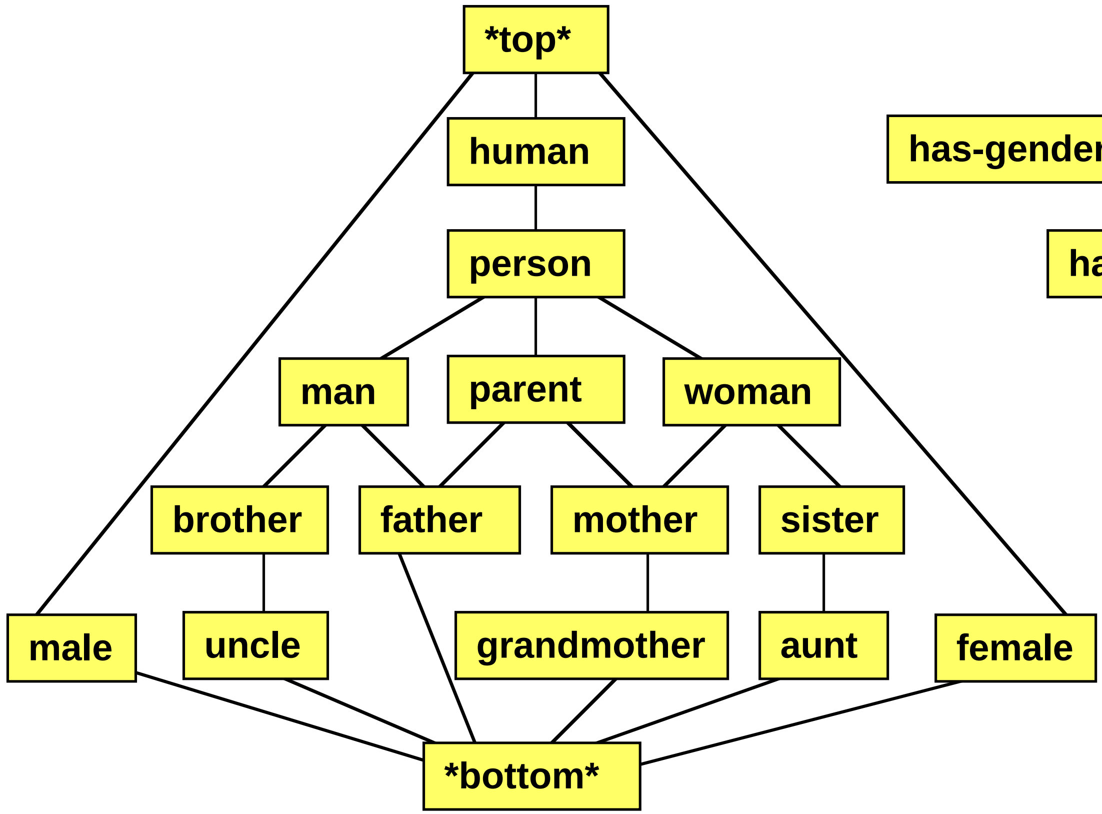
```
(implies *top* (all has-child person))
(implies (some has-child *top*) parent)
(implies (some has-sibling *top*) (or brother sister))
(implies *top* (all has-sibling (or sister brother)))
(implies *top* (all has-sister (some has-gender female)))
(implies *top* (all has-brother (some has-gender male)))
```

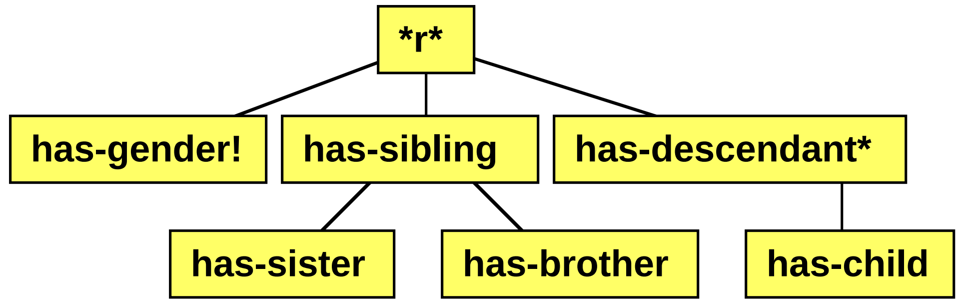
domain and range restrictions for roles

```
(implies person (and human (some has-gender (or female male))))
(disjoint female male)
(implies woman (and person (some has-gender female)))
(implies man (and person (some has-gender male)))
(equivalent parent (and person (some has-child person)))
(equivalent mother (and woman parent))
(equivalent father (and man parent))
(equivalent grandmother (and mother (some has-child (some has-child person))))
(equivalent aunt (and woman (some has-sibling parent)))
(equivalent uncle (and man (some has-sibling parent)))
(equivalent brother (and man (some has-sibling person)))
(equivalent sister (and woman (some has-sibling person)))
```

concepts

Concept and Role Hierachies Implied by TBox





r	universal role
!	attribute (feature)
*	transitive role

TBox Inferences

A DL system offers several inference services. At the core is a consistency test:

$$C \overset{?}{\models} \textbf{*bottom*} \quad \textbf{(the empty concept)}$$

Example: (and (at-least 1 has-child) (at-most 0 has-child)) $\models$ *bottom*

Consistency checking is the basis for several other inference services:

- **subsumption**

  (implies C1 C2) <=> (and C1 (not C2)) $\models$ *bottom*

- **classification of a concept expression**

  searches the existing concept hierarchy for the most special concept which subsumes the concept expression

ABox of a Description Logic System

TBox = terminological knowledge (concepts and roles)

ABox = assertional knowledge (facts)

An ABox contains:

- **concept assertions** **(instance IN C)**
  individual IN is instance of a concept expression C
- **role assertions** **(related IN_1 IN_2 RN)**
  individual IN_1 is related to IN_2 by role RN

- **An ABox always refers to a particular TBox.**
- **An ABox requires unique names**
- **ABox facts are assumed to be incomplete (OWA).**

  OWA = Open World Assumption
  (new facts may be added, hence inferences are restricted)

  CWA = Closed World Assumption
  (no facts may be added)

ABox Inferences

ABox inferences = inferring facts about ABox individuals

Typical queries:

- **consistency** ***is ABox consistent?***
- **retrieval** ***which individuals satisfy a concept expression?***
- **classification** ***what are the most special concept names which describe an individual?***

ABox consistency checking is in general more complicated than TBox consistency checking

ABox consistent <=> there exists a "model" for ABox and TBox

All ABox inferences are based on the ABox consistency check.

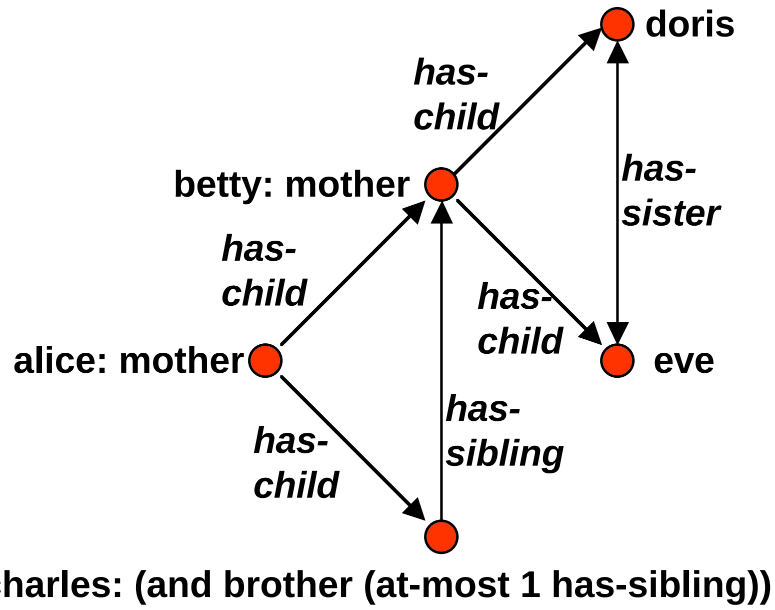
Example of ABox Queries

Contents of ABox

(instance alice mother)
(related alice betty has-child)
(related alice charles has-child)

(instance betty mother)
(related betty doris has-child)
(related betty eve has-child)

(instance charles brother)
(related charles betty has-sibling)
(instance charles (at-most 1 has-sibling))
(related doris eve has-sister)
(related eve doris has-sister)



Questions and answers

(individual-instance? doris woman)
T

Is doris *instance of the concept* woman*?*

(individual-types eve)
((sister) (woman) (person) (human) (*top*))

Of which concept names is eve *an instance?*

(individual-fillers alice has-descendant)
(doris eve charles betty)

What are the descendants of eve*?*

(concept-instances sister)
(doris betty eve)

Which instances has the concept sister*?*

Abstraction with Description Logics

abstraction = omission of properties or relations, extending a concept, generalization

<u>Examples</u>:

- **superordinate concept name of a concept expression (= concept classification)**

  (and person (some has-size tall)) → person

- **generalization of concept expressions**

  (and (some has-occupation professor) (at-least 3 has-child))

  ⇓

  (and (some has-occupation civil-servant) (at-least 1 has-child))

- **concept expression which subsumes several individuals**

  1. classify individuals

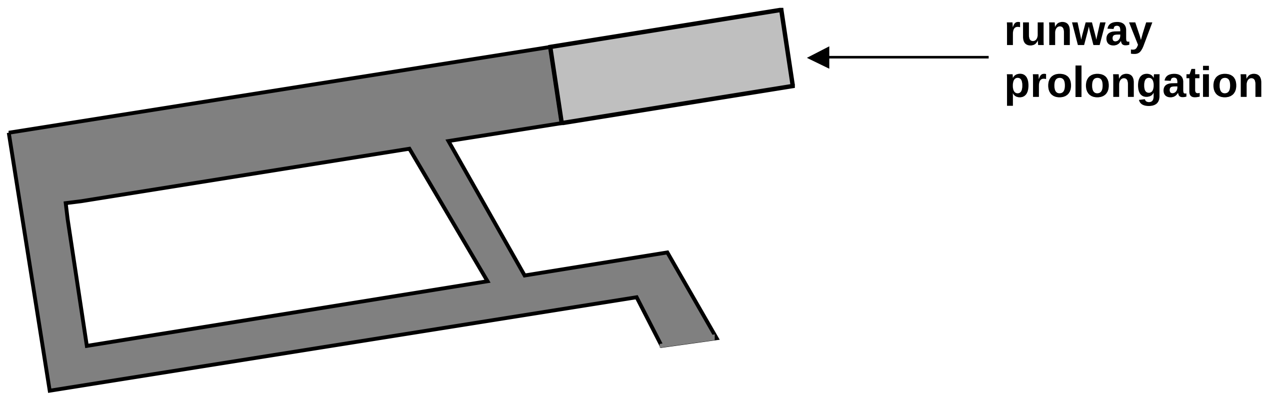
  2. determine least common subsumer (LCS)

  - **for RACER: trivial solution in terms of (OR C_1 ... C_n)**
  - **for DLs without OR: special abstraction operator LCS**

Image Interpretation as Deduction

Aerial Image Analysis as Classification

Classification of changes using a description logic (Lange and Schroeder 95)



- **Using the LOOM-classifier to determine the change concept which describes given evidence**
- **Bottom-up analysis of images, no hypothesis generation, no predictive control**

Concepts and Relations for Airfield Classification (1)

```
(defconcept road-object
     :is   (:and scene-object
                 (> has-length has-width)
                 (:the has-material (:one-of concrete asphalt)

(defconcept runway
     :is   (:and road-object
                 rectangle
                 (:the has-length (:through 2150 4000))
                 (>= has-width 45)
                 (:at-least 1 has-connecting-driveway)
                 (:all has-connecting-driveway (>= has-width 23))
                 (:satisfies
                   ((?x) ... driveway and taxiway constraints ...)))

(defrelation has-connecting-driveway
     :is   (:and has-neighbor
                 (:domain road-object)
                 (:range
                   (:and  road-object
                          (:at-least 2 has-neighbor road-object)))))

(defrelation has-neighbor
     :function   ((x) (compute-neighboring-objects x))
     :characteristics  (:symmetric :multiple-valued))
```

necessary and sufficient conditions for classifying
... a road-object

... a runway

procedural constraints

important geometrical relation has-neighbor *must be implemented procedurally*

Concepts and Relations for Airfield Classification (2)

```
(defconcept basic-change
     :implies    (:and (:exactly 1 has-before)
                       (:exactly 1 has-after)
                       (<   (:compose has-before has-time)
                            (:compose has-after has-time))))

(defconcept elongation
     :is    (:and basic-change
                  (:relates  has-contained-object
                             has-before
                             has-after)
                  (<      (:compose has-before has-length)
                          (:compose has-after has-length))
                  (=      (:compose has-before has-width)
                          (:compose has-after has-width))

(defconcept runway-elongation
     :is
            (:and elongation
                  (:all has-before runway)
                  (:all has-after runway)))
```

primitive concept **basic-change,** ***classification must be provided interactively***

defined concepts **elongation** ***and*** **runway-elongation*****, classification is provided by deduction***

Image Interpretation as Deduction?

The classifier of a description logic carries out classifications automatically:

evidence => class (concept) membership

Problems:

- **partial evidence must be sufficient**
- **deduction of all possible partial interpretations**
- **no goal-oriented analysis**
- **no comparative evaluation of conflicting interpretations**

Support of hypothesize-and-test cycle is required !

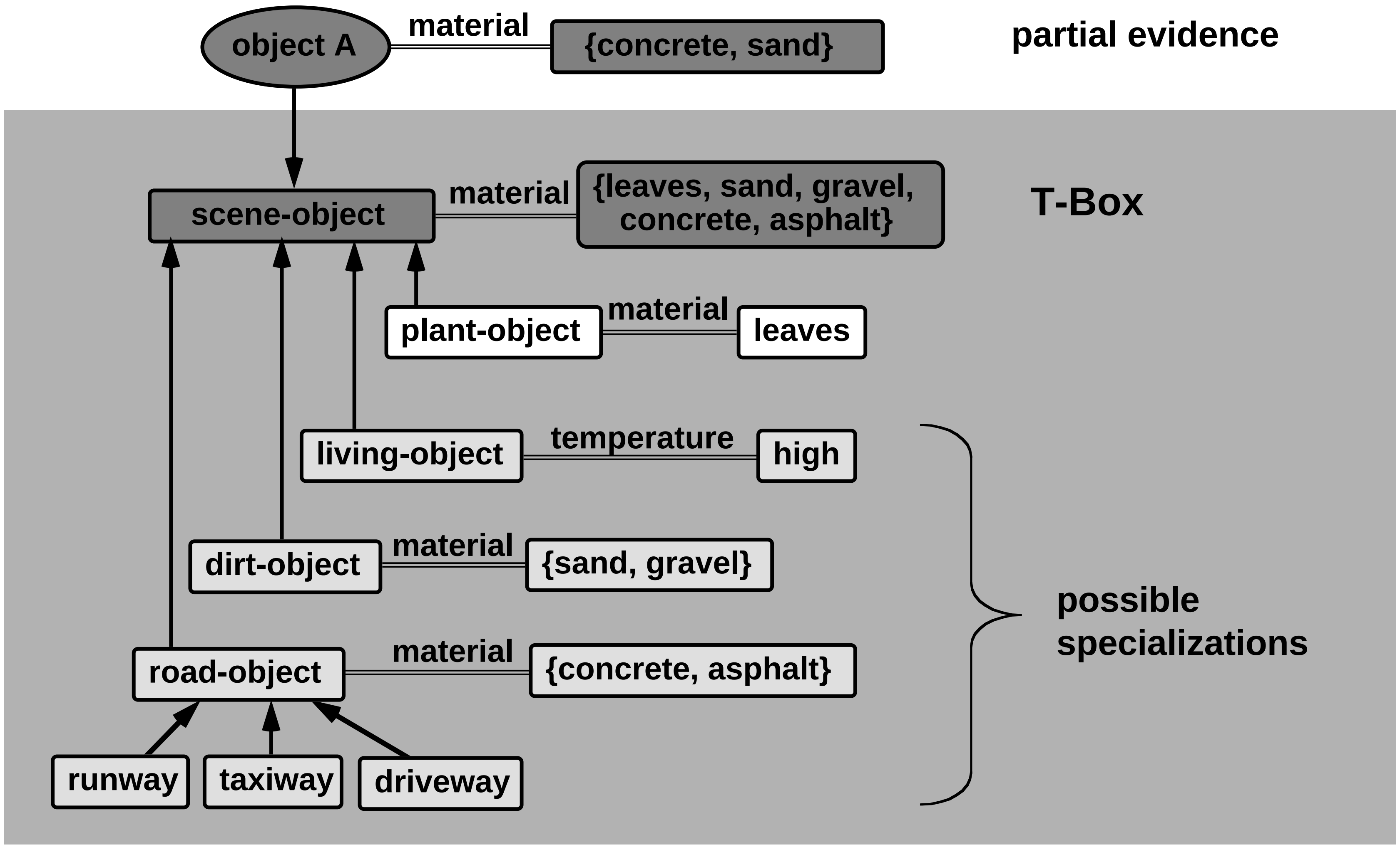
Hypothesizing Possible Concept Specializations

Extension of description logic reasoning service for hypothesis generation:

- **Which concept hypotheses can be specialized further consistent with existing evidence?**
- **Which additional evidence is required for specialization?**

1. partial evidence => consistent concepts
2. partial evidence + concepts => missing evidence

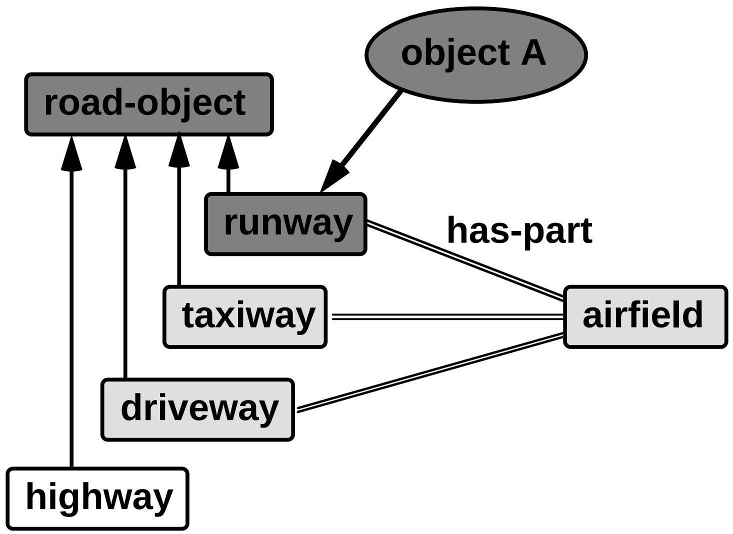
Example for Possible Concept Specializations



Hypothesizing Possible Aggregats (1)

For which concepts (aggregats) are roll fillers (parts) available?

- **Provide concepts which are consistent with existing role fillers**
- **Which roles provide decisive evidence?**
- **Criteria for ranking hypotheses**

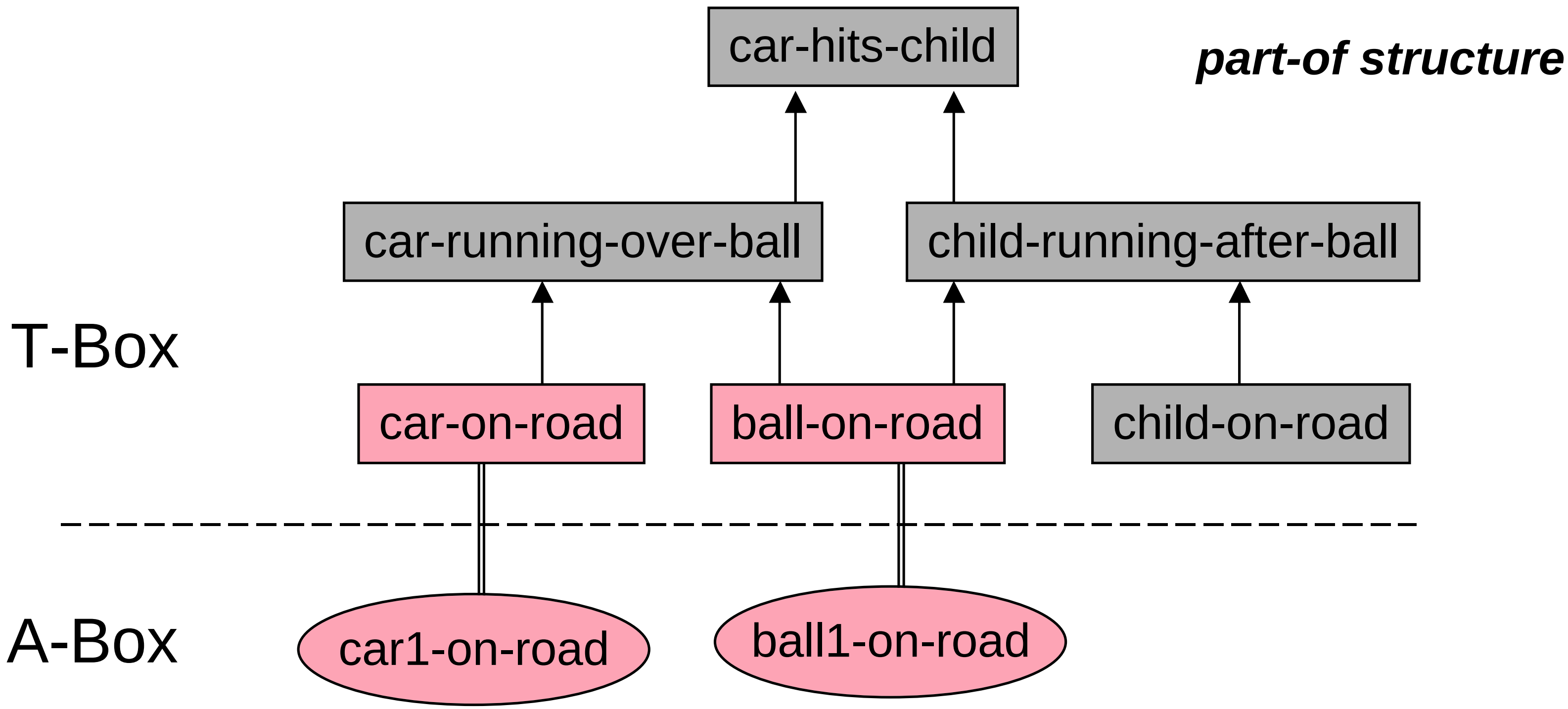


Existing instance runway ***is evidence for*** airfield ***and its further parts*** taxiway ***and*** driveway**.**

Hypothesizing Possible Aggregats (2)

For which concepts (aggregats) are roll fillers (parts) available?

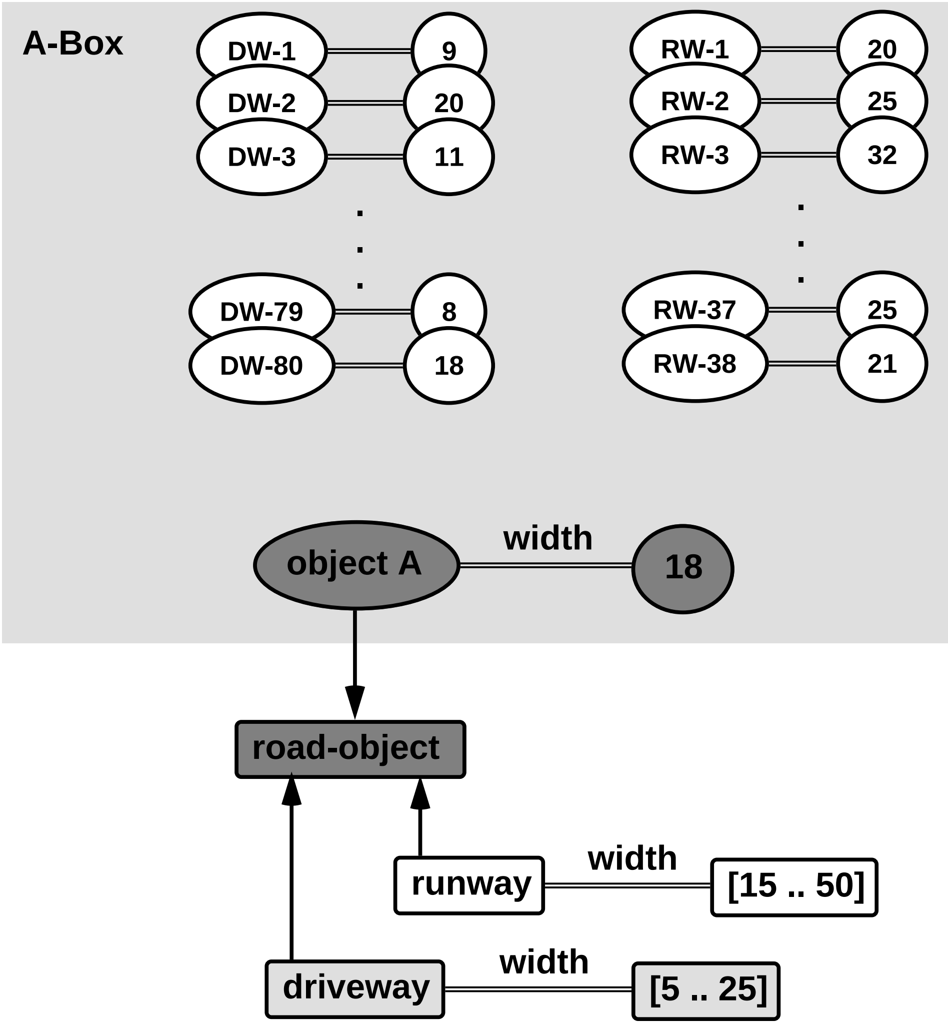
Generating temporal and spatial expectations:



Exploiting A-Box Statistics

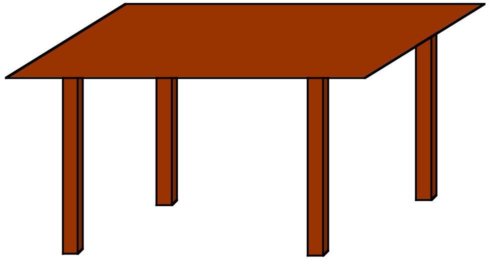
What are the most probable concepts (aggregates) for given parts (role fillers)?

- **using experiences for predictions**
- **ranking hypotheses**



Logics of Image Interpretation

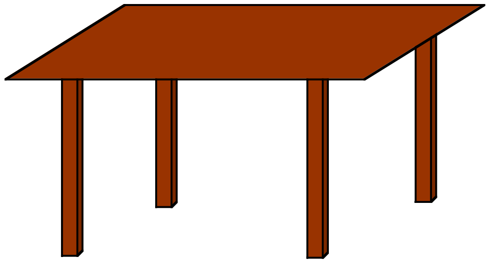
Describing Image Interpretation in Logical Terms



?

deduction

"from the evidence I <u>conclude</u> that this is a table"



!

model construction

"my conceptual model of a table <u>explains</u> the evidence"

Reiter & Mackworth 87, Matsuyama 90, Schröder 99

Image Interpretation as (Logical) Model Construction

An <u>interpretation</u> I = [D, φ, π] of a logical language maps
- constant symbols of the language into elements of a real-world domain D
- predicate symbols of the language into predicate functions over D

A <u>model</u> of some clauses is an interpretation where all predicates are true.

Image interpretation as model construction:

- establish mapping φ by assigning segmentation results to constant symbols
- establish mapping π by assigning computational procedures to predicate symbols
- find clauses for which predicates are true

Deciding whether a model exists is undecidable in FOPC!
There may be infinitely many models!

Finite Model Construction (Reiter & Mackworth 87)

- **an image consists of regions and chains (edges)**
- **the image elements constitute all constant symbols of an interpretation (domain closure assumption)**
- **different constant symbols denote different image elements and vice versa (unique name assumption)**

➡ **Problem can be expressed in Propositional Calculus and solved as a constraint satisfaction problem (CSP)**

For MAPSEE, scene interpretation amounts to finding a mapping π for predicates *road, river, shore, land, water.*

Logics of SIGMA (Matsuyama & Hwang 90)

Image interpretation is set of hypotheses which

- extend generic knowledge
- allow to deduce the observations

⇒ partial model construction

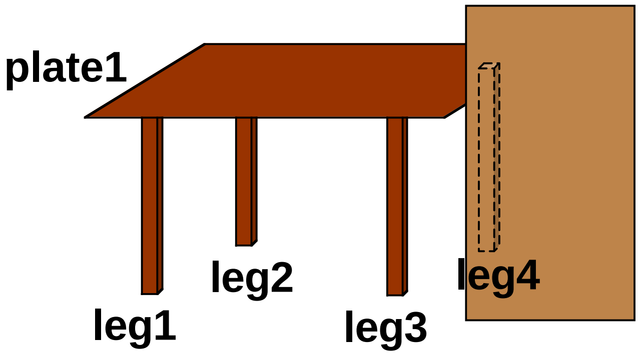
The number of existing objects must be limited for the interpretation procedure to terminate. (e.g. no interpretations involving invisible objects).

ABox Consistency Checking in Description Logics

Consistency checking of an ABox amounts to model construction:

Consistency checking generates an interpretation including all additional individuals which are required to satisfy a conceptual framework.

Example: part-whole completion



conceptual model requires 4 legs for a table

hallucinated 4th leg

However:

- **Model construction in existing reasoning systems is an open-world consistency check**
- **Additional individuals are hypothesised liberally to generate a model, without consideration of missing visual evidence**
- **Ranking is required so that "preferred interpretations" can be delivered**

Image Interpretation as Configuration

Image Interpretation as a Configuration Problem

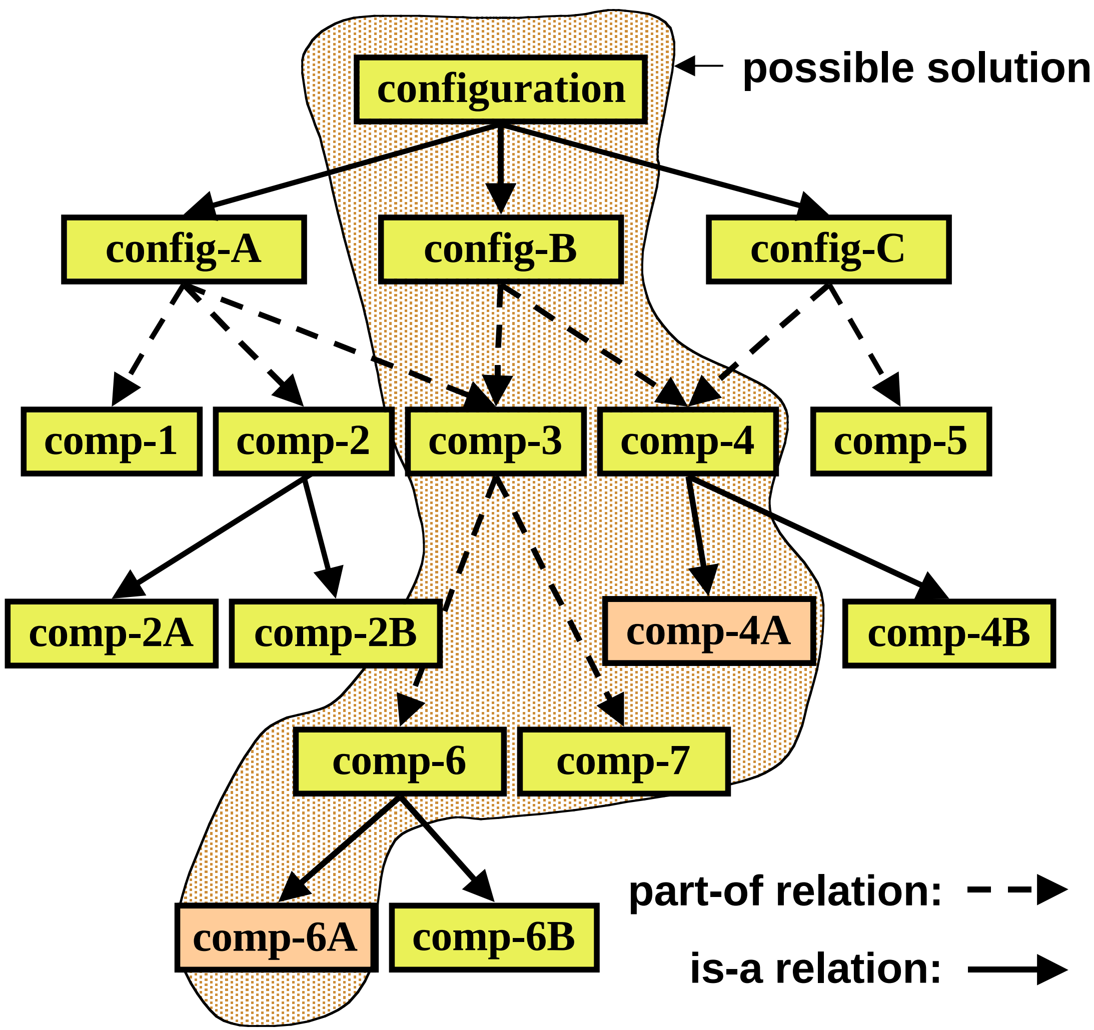
What is a configuration problem?

Construct an aggregate (a configuration) given
- **generic descriptions of parts**
- **compatibility constraints between parts**
- **a concrete task description, possibly in terms of given parts**

Image interpretation may be viewed as constructing a "scene aggregate" which
- **meets generic constraints and**
- **incorporates parts prescribed by the concrete task**

Methods and tools of configuration technology may be exploited

Illustration of Configuration



- boxes (frames) specify aggregate and component properties
- has-part relations bind components to aggregates
- is-a relations describe variants of entities
- constraints between entities (not shown) restrict choices and parameter combinations

Signal-symbol Interface

Computing Primitive Occurrences

Primitive occurrences

Qualitative primitives

Perceptual primitives

Geometric scene description (GSD)

Geometric Scene Description (GSD)

The GSD is a quantitative object-level scene interpretation in terms of
- recognised objects and
- their (possibly time-varying) locations in the scene

- useful definition of input for high-level scene interpretation
- objects may only be roughly classified (e.g. "moving-object")
- high-level processes must be able to correct mistakes and fill in missing evidence

Perceptual Primitives

Perceptual primitives are geometrical and photometrical attributes which can be immediately determined from a GSD.

For <u>object configurations</u>:

- **objects provide reference features in terms of**
  - **locations (center of gravity, corners, surface markings, etc.)**
  - **lines (edges, surface markings, axes of minimal inertia, etc.)**
  - **orientations (inate, motion, viewer)**
- **perceptual primitives are measurements between reference features:**
  - **distance**
  - **angle**
  - **temporal derivatives thereof**

Qualitative Primitives

Qualitative primitives are predicates over perceptual primitives constant over some time interval.

- **qualitatively constant values**
  e.g. constant orientation, constant distance
- **values within a certain range**
  e.g. topological relations, degrees of nearness, typical speeds
- **values smaller or larger than a threshold**
  e.g. increase of distance, slowing down

Qualitative Predicates for Occurrences in Traffic Scenes

Used in NAOS: "Natural-language description of object motions in traffic scenes"

exist
move
decelerate, accelerate
turn_left, turn_right
increasing_distance, reducing_distance
along, across
in_front_of, behind, beside
on, above, under, below
at, near_to
between
(and others)

Note that qualitative predicates are often (but must not be) part of natural language.

For technical applications one may use technical (artificial) qualitative predicates, e.g.

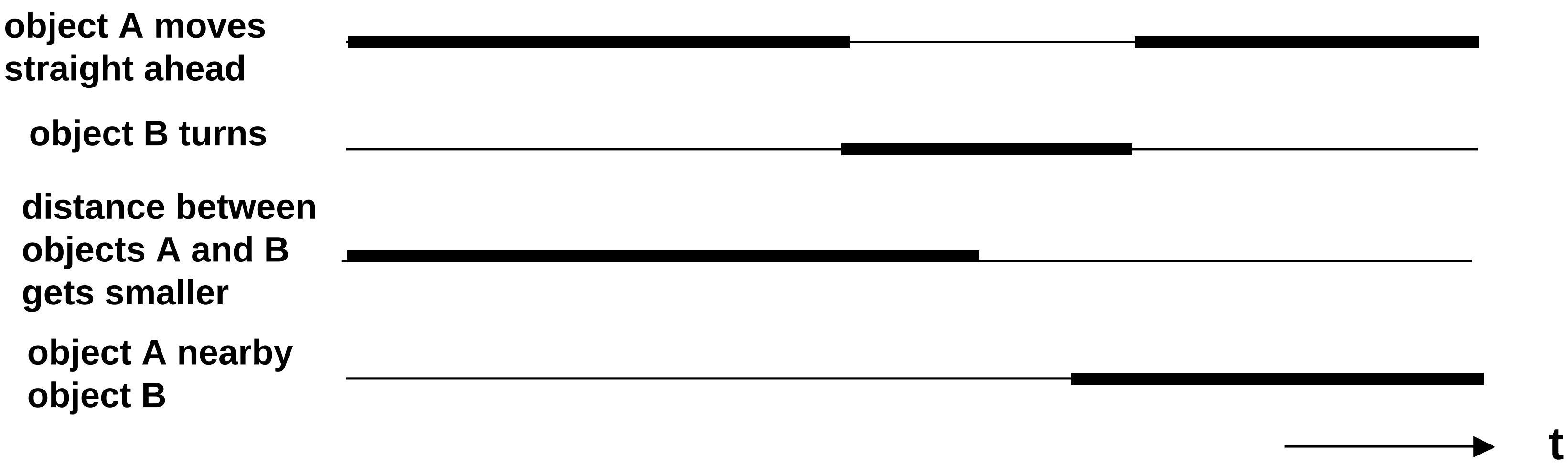
v50 (= 45 ≤ v ≤ 55 km/h)

shape_x (= shape_index ≤ 4.174)

Primitive Occurrences

A primitive occurrence is a conceptual entity with one or more objects for which a qualitative predicate is true over a time interval.

Primitive occurrences provide the raw material for high-level scene interpretations.



Temporal Decomposition of Scenes

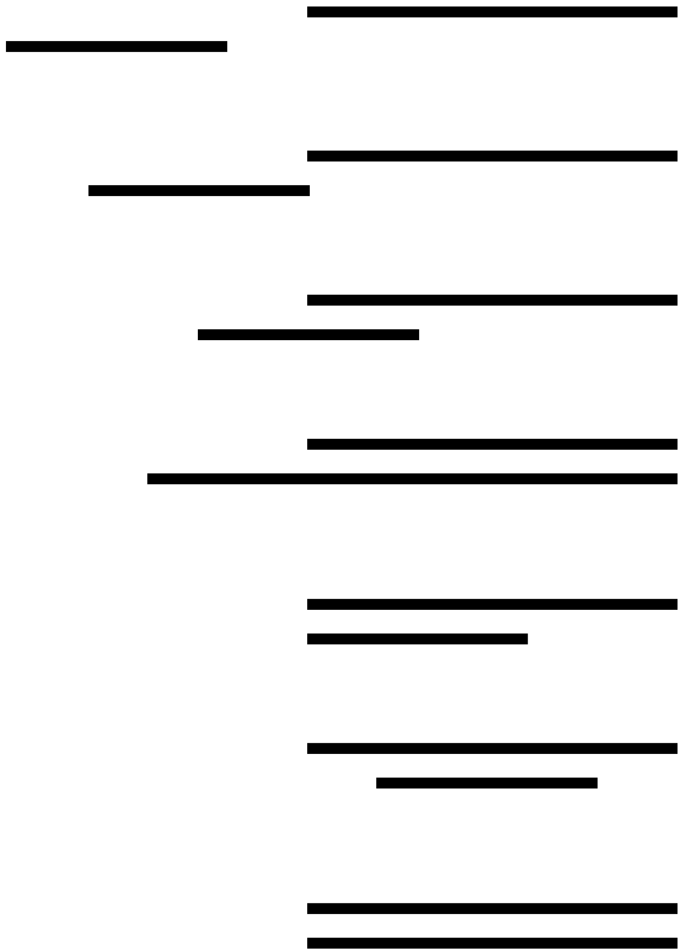
Temporal decomposition

- **by temporal segmentation:**
  constancies of time-dependent properties of an image sequence
- **by model matching:**
  occurrences which obey a model

Compare with spatial decomposition

- **by spatial segmentation:**
  image regions with spatially constant (uniform) properties
- **by model matching:**
  image regions which obey a model

Interval Relations in Allen´s Algebra



BEFORE (I1, I2)	**< >**
MEETS (I1, I2)	**m mi**
OVERLAPS (I1, I2)	**o oi**
FINISHES (I1, I2)	**f fi**
STARTS (I1, I2)	**s si**
DURING (I1, I2)	**d di**
EQUAL (I1, I2)	**=**

Convex Time-point Algebra

Qualitative relations between time points which can be described by the inequality

$$T1 + c12 \leq T2$$

(T1, T2: time points; c12: constant)

"Convex relation":

All intervals satisfying a convex relation can be generated by continuous displacements of the begin and end points of an interval

In Allen´s Algebra:

convex relation e.g. d v m 

non-convex relation e.g. b v bi 

Occurrence Models

Structure of Occurrence Models

Basic ingredients:	• relational structure • taxonomy • partonomy • spatial relational language • temporal relational language • object appearance models

- **An occurrence model describes a class of occurrences by**
  - **properties**
  - **sub-occurrences (= components of the occurrence)**
  - **relations between sub-occurrences**

- **A primitive occurrence model consists of**
  - **properties**
  - **a qualitative predicate**

- **Each occurrence has a begin and end time point**

Occurrence Model for Overtaking in Street Traffic

Predicate:	**overtake**
	:is-a occurrence-model
	:local-name ov
Arguments:	**(?veh1 :is-a vehicle)**
	(?veh2 :is-a vehicle)
Time marks:	**(ue.B ue.E)**
Component events:	**(mv1 :is-a (move ?veh1 mv1.B mv1.E))**
	(mv2 :is-a (move ?veh2 mv2.B mv2.E))
	(bh :is-a (behind ?veh1 ?veh2 bh.B bh.E))
	(bs :is-a (beside ?veh1 ?veh2 bs.B bs.E))
	(bf :is-a (before ?veh1 ?veh2 bf.B bf.E))
	(ap :is-a (approach ?veh1 ?veh2 ap.B ap.E))
	(rc :is-a (recede ?veh1 ?veh2 rc.B rc.E))
Temporal relations:	**(ov.B = bh.B)**
	(ov.E = bf.E)
	(ap :during mv1)
	(ap :during mv2)
	(rc :during mv1)
	(rc :during mv2)
	(bh :overlaps bs)
	(bs :overlaps bf)
	(bh :during ap)
	(bf :during rc)

Table-laying Scenario

Important high-level characteristics:

- correlated multiple object motion
- intended actions
- influence of context (temporal, spatial, task context)
- qualitative spatial and temporal relations
- uncertainty
- smart room learning context (supervised, unsupervised)
- interface with common sense

Table-laying scenario of project CogVis:

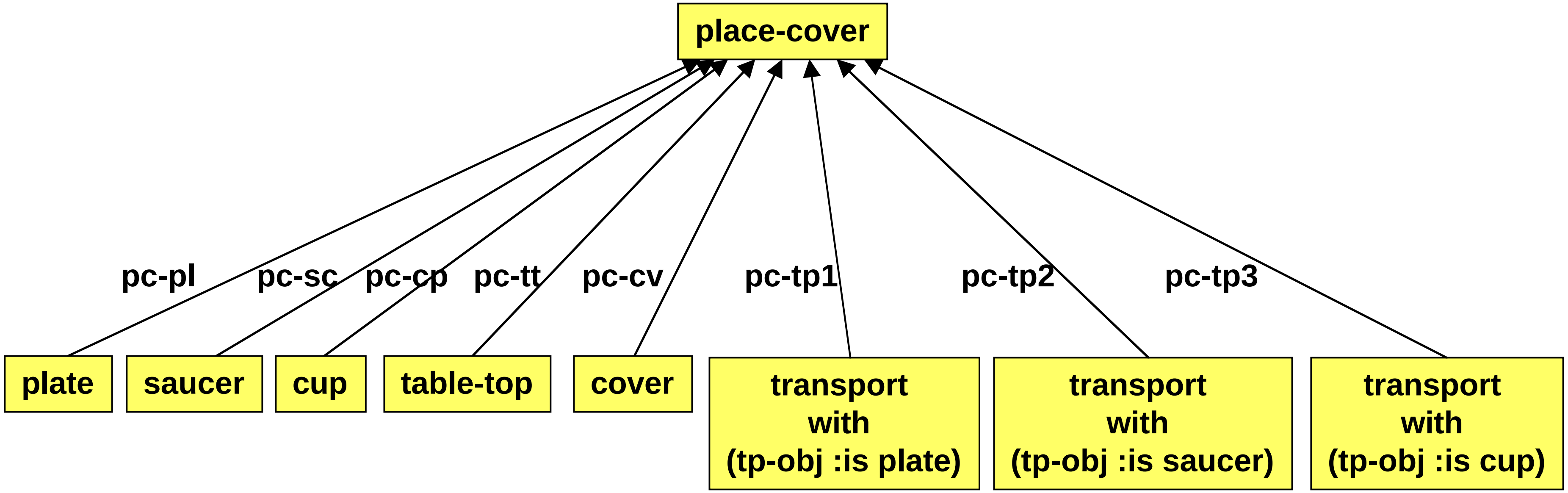
Stationary cameras observe living room scene and recognize meaningful occurrences, e.g. placing a cover onto the table.

Occurrence Model for Placing a Cover

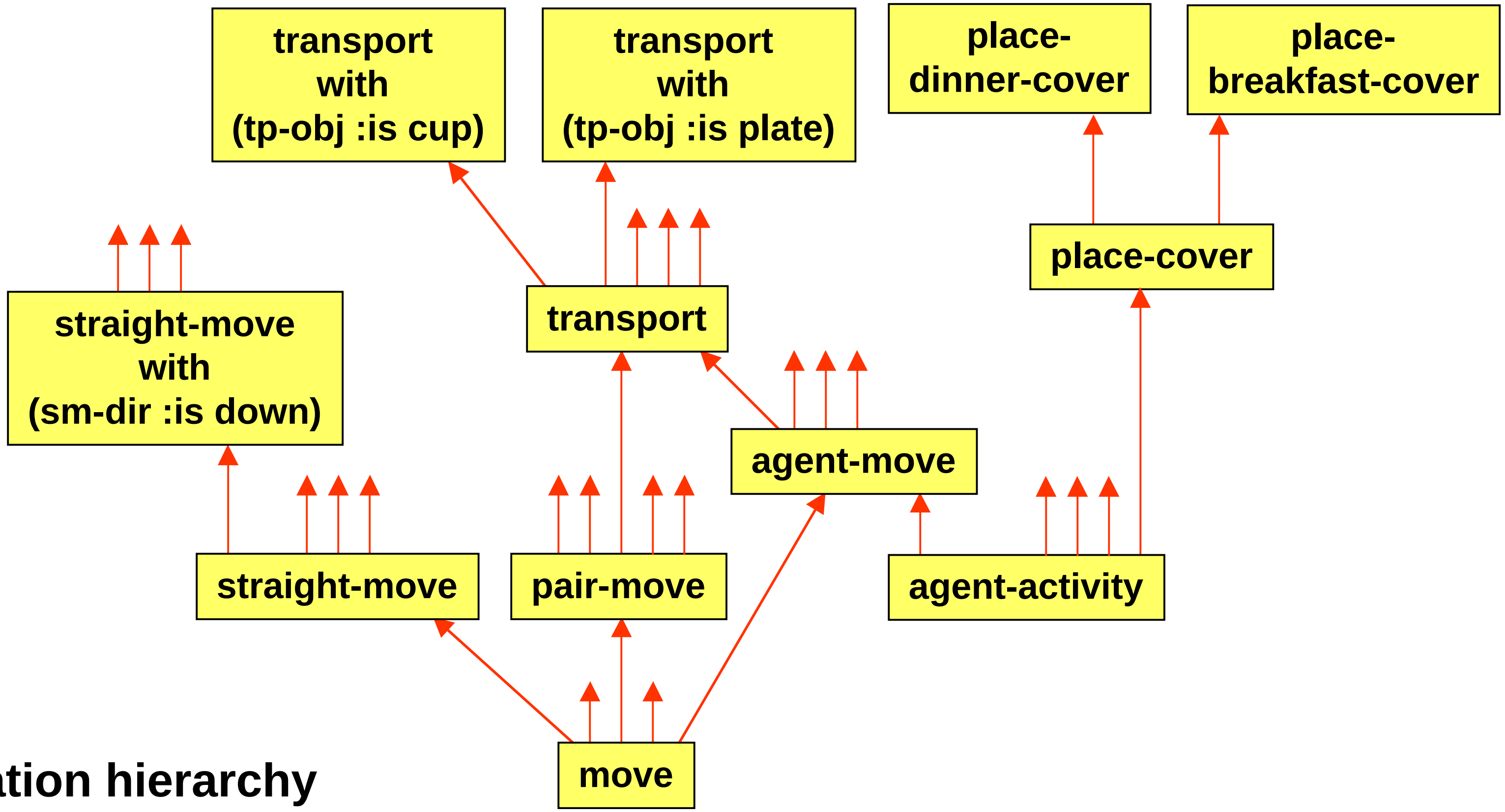
name:	place-cover
parents:	:is-a agent-activity
parts:	pc-pl :is plate
	pc-sc :is saucer
	pc-cp :is cup
	pc-tt :is table-top
	pc-tp1 :is (transport with (tp-obj :is plate))
	pc-tp2:is (transport with (tp-obj :is saucer))
	pc-tp3 :is (transport with (tp-obj :is cup))
	pc-cv :is cover
time marks:	pc-tb, pc-te :is timepoint
constraints:	pc-tp1.tp-ob = pc-cv.cv-pl = pc-pl
	pc-tp2.tp-ob = pc-cv.cv-sc = pc-sc
	pc-tp3.tp-ob = pc-cv.cv-cp = pc-cp
	pc-cv.cv-tb ≥ pc-tp1.tp-te
	pc-cv.cv-tb ≥ pc-tp2.tp-te
	pc-cv.cv-tb ≥ pc-tp3.tp-te
	pc-tp3.tp-te ≥ pc-tp2.tp-te
	pc-tb ≤ pc-tp1.tb
	pc-tb ≤ pc-tp2.tb
	pc-tb ≤ pc-tp3.tb
	pc-te ≥ pc-cv.cv-tb
	pc-te ≥ pc-tb + 80Δt

Parts Structure

- associational structure between aggregats and their parts
- probabilistic information may be added



Concept Hierarchy



- specialization hierarchy
- nodes are concept expressions
- multiple inheritance

Aggregates as DL Concepts

frame-like notation

```
NAME
place-cover is-a agent-activity

PARTS
pc-tt is-a table-top
pc-tp1 is-a transport
                  with (tp-obj is-a  plate)
pc-tp2 is-a transport
                  with (tp-obj is-a  saucer)
pc-tp3 is-a transport
                  with (tp-obj is-a  cup)
pc-cv is-a  cover

CONSTRAINTS
<identity constraints on parts>
<spatial constraints on parts>
<temporal constraints on parts>
```

name

roles

concrete domain predicates

DL concept expressions

```
(equivalent  place-cover
      (and  agent-activity
            (some  pc-tt  table-top)
            (some  pc-tp1
                  (and  transport
                        (some tp-obj plate) )
            (some  pc-tp2  transport)
                  (and  transport
                        (some tp-obj saucer) )
            (some  pc-tp3  transport)
                  (and  transport
                        (some tp-obj cup) )
            (some  pc-cv  cover)
            <identity constraints on parts>
            <spatial constraints on parts>
            <temporal constraints on parts>
```

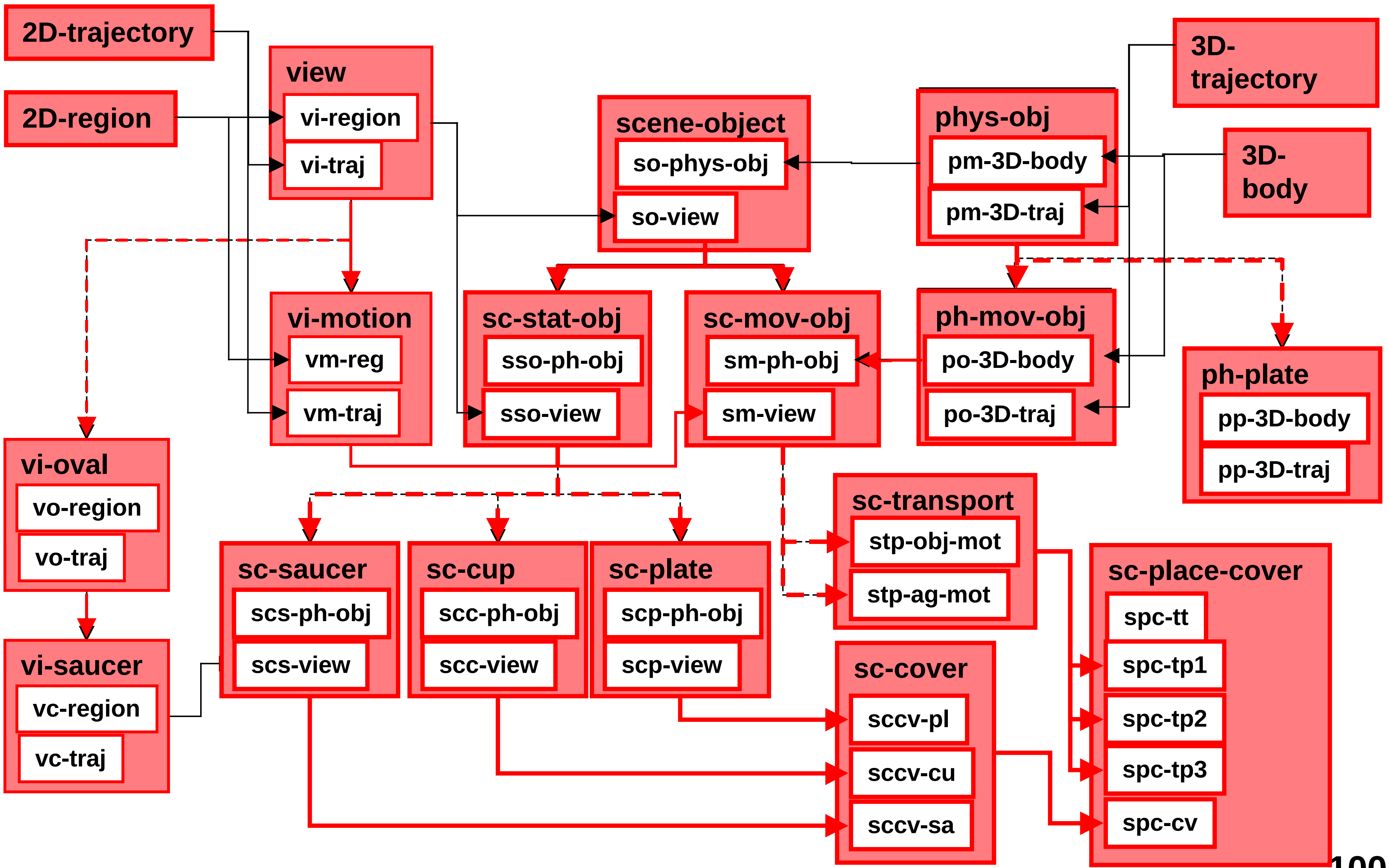
Navigating in Hallucination Space

What is the Space of Interpretations?

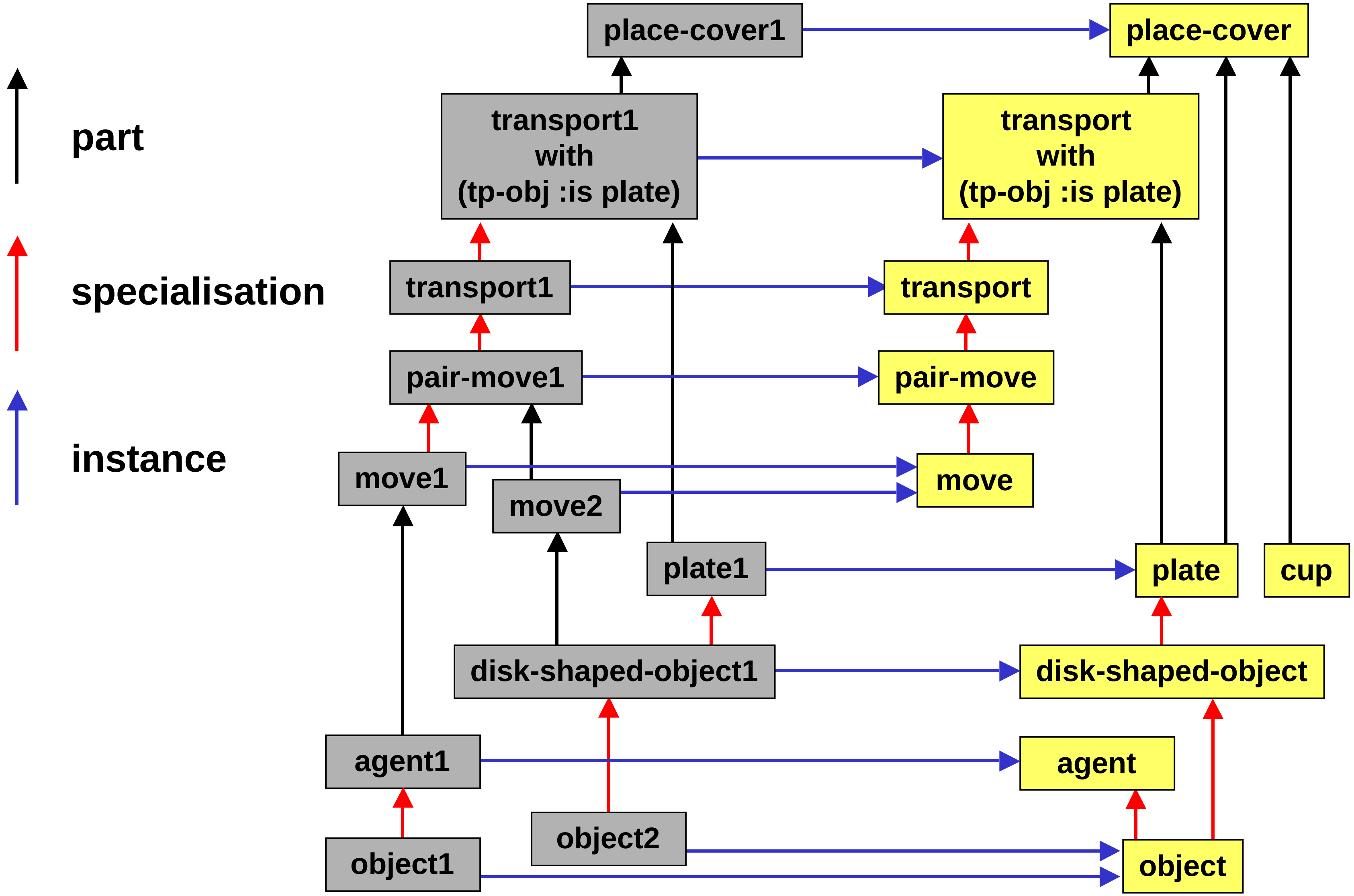
Vision is controlled hallucination
(Kender 1985?)

- **interpretations must be consistent**
  - **consistency is standard inference service of DLs**
  - **consistency tolerates interpretations without any evidence (complete hallucination)**
- **interpretations must be context and task dependent**
  - **do not expect beakfast covers at dinner time**
  - ***"Is the table laid?"*** **narrows down the interpretation task**
- **interpretations must be "preferred"**
  - **aggregates vs. individual objects**
  - **most special concepts, basic categories, dissolved disjunctions**
  - **more likely vs. less likely interpretations**

Aggregates in Taxonomical Hierarchies

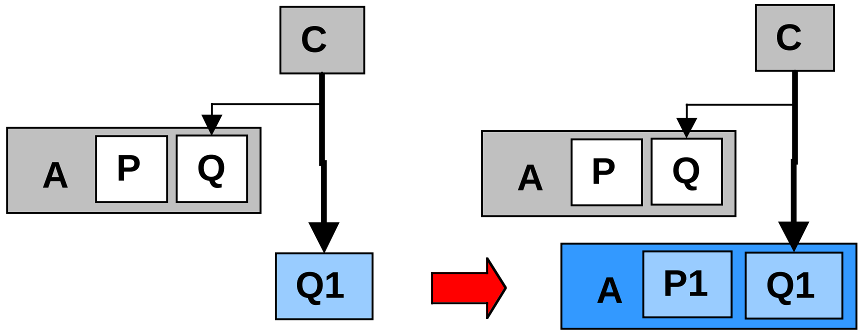


Typical Model-based Interpretation Steps

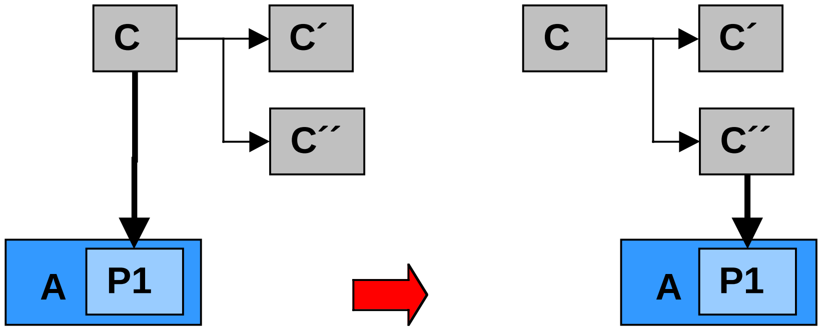


Three Kinds of Interpretation Steps

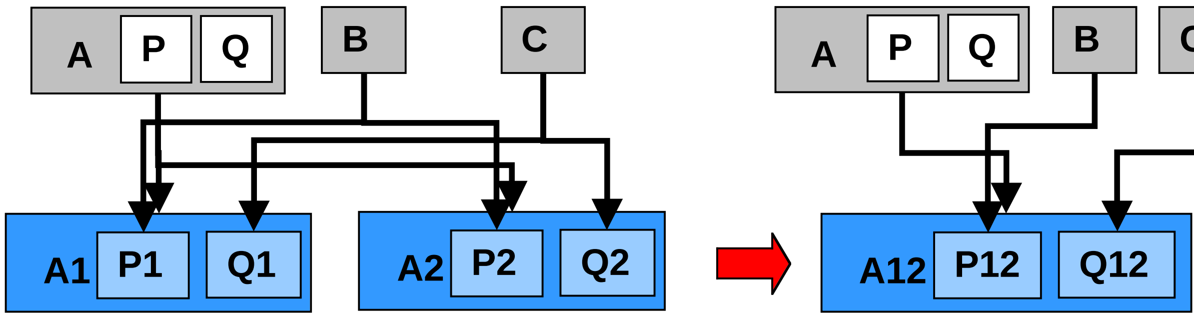
**aggregate instantiation
("part-whole-reasoning")**



**instance refinement
("specialisation")**

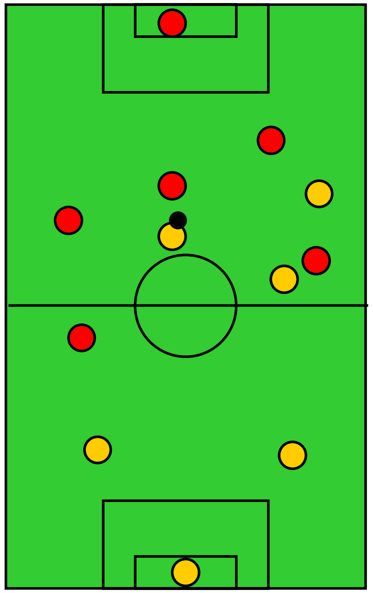


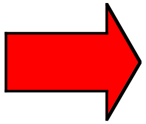
**instance merging
("converging evidence")**



Recognizing Intentions

Recognizing Intentions and Plans





Intention recognition in soccer games (Retz-Schmidt 91):

"Brandt dribbelt, um dem gegnerischen Tor nahe zu kommen"

("Brandt dribbles to get close to the opposing goal")

"Meier läuft zu Brandt, um ihn am Torschuß zu hindern"

("Meier runs to Brandt to prevent him from shooting a goal")

- **model-based representation of plans and counter plans**
- **partial instatiation allows predictions and explanations**

Intention recognition has been used in robot soccer (RoboCup)

Plan Recognition

Given:
- observed actions
- knowledge about likely goals of actor

⇒ predict further actions

⇒ plan own actions (cooperative or adversary)

Example ("smart room" or service robotics scenario):

Observations: tea-time: person gets up - person walks to door - ...

Predictions: ... - person goes to kitchen - person prepares tea

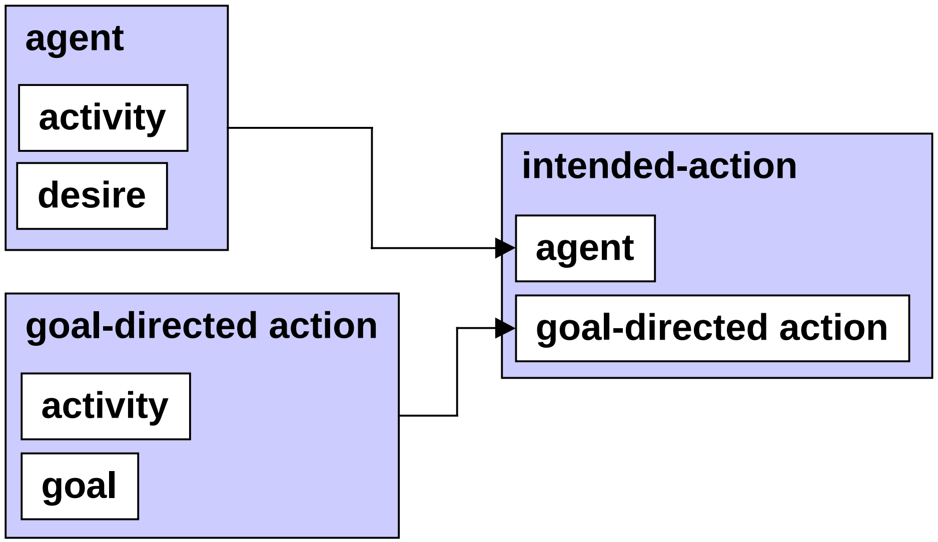
Plan recognition by

- matching partial action sequences to plan models
  (same principle as occurrence recognition)
- generating likely plans from the initial action sequence

Models for Intention Recognition

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

name:	**scene-intended-place-cover**
parents:	**:is-a scene-intended-action**
parts:	**sipc-pc :is-a scene-place-cover**
	sipc-ag :is-a scene-agent
	with (sipc-ag.desire = sipc-pc.goal)
	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.

Bayesian Nets

Probabilistic Models for Occurrences

Modelling probabilistic dependencies (causalities) and independencies between discrete events of a scene

X_i random variable *models uncertain propositions about a scene*

$X_i = a$ hypothesis

Decomposition of joint probabilities:

$$P(X_1, X_2, X_3, \dots, X_n) = P(X_1 \mid X_2, X_3, \dots, X_n) \bullet P(X_2 \mid X_3, X_4, \dots, X_n) \bullet \dots \bullet P(X_{n-1} \mid X_n) \bullet P(X_n)$$

Simplification in the case of statistical independence:

X independent of X_i

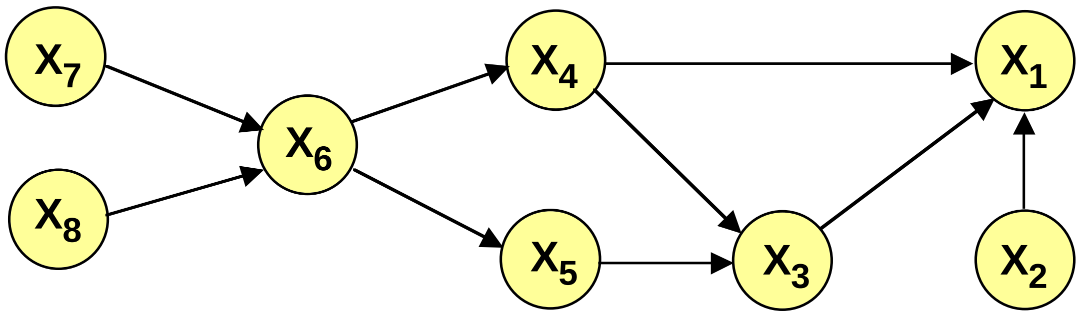
$$P(X \mid X_1, \dots X_{i-1}, X_i, X_{i+1}, \dots, X_n) = P(X \mid X_1, \dots X_{i-1}, X_{i+1}, \dots, X_n)$$

Joint probability of N variables may be simplified by ordering the variables according to their direct dependence (causality)

Causality Graph

Conditional dependencies (causality relations) of random variables define partial order.

Representation as a directed graph:



$P(X_1, X_2, X_3, \ldots, X_8) =$

$P(X_1 \mid X_2, X_3, X_4) \cdot P(X_2) \cdot P(X_3 \mid X_4, X_5) \cdot P(X_4 \mid X_6) \cdot P(X_5 \mid X_6) \cdot P(X_6 \mid X_7 X_8) \cdot P(X_7) \cdot P(X_8)$

Constructing a Bayesian Net

By domain analysis:

1. Select discrete variables X_i relevant for domain
2. Establish partial order of variables according to causality
3. In the order of decreasing causality:
   (i) Generate node X_i in net
   (ii) As predecessors of X_i choose the smallest subset of nodes which are already in the net and from which X_i is causally dependent
   (iii) determine a table of conditional probabilities for X_i

By data analysis:

Use a learning method to establish a Bayes Net approximating the empirical joint probablity distribution.

Computing Inferences

We want to use a Bayesian Net for probabilistic inferences of the following kind:

Given a joint probability $P(X_1, ... , X_N)$ represented by a Bayes Net, and evidence $X_{m_1}=a_{m_1}, ... , X_{m_K}=a_{m_K}$ for some of the variables, what is the probability $P(X_n= a_i \mid X_{m_1}=a_{m_1}, ... , X_{m_K}=a_{m_K})$ of an unobserved variable to take on a value a_i ?

In general this requires

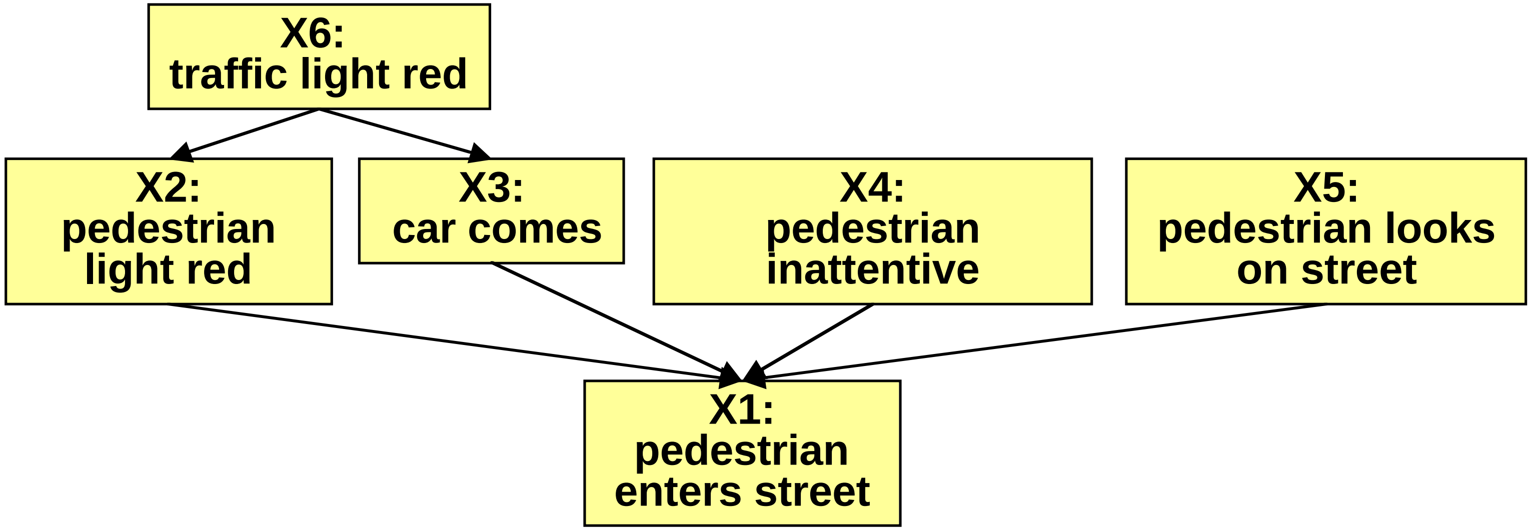
- expressing a conditional probability by a quotient of joint probabilities

$$P(X_n= a_i \mid X_{m_1}=a_{m_1}, ... , X_{m_K}=a_{m_K}) = \frac{P(X_n= a_i, X_{m_1}=a_{m_1}, ... , X_{m_K}=a_{m_K})}{P(X_{m_1}=a_{m_1}, ... , X_{m_K}=a_{m_K})}$$

- determining partial joint probabilities from the given total joint probability by summing out unwanted variables

$$P(X_{m_1}=a_{m_1}, ... , X_{m_K}=a_{m_K}) = \sum_{X_{n_1}, ... , X_{n_K}} P(X_{m_1}=a_{m_1}, ... , X_{m_K}=a_{m_K}, X_{n_1}, ... , X_{n_K})$$

Example: Traffic Behaviour of Pedestrians



Conditional probability table for each node must be known

Examples: P(X1 | X2, X3, X4, X5)

X1	X2	X3	X4	X5	P
T	T	T	T	T	0.3
F	T	T	T	T	0.7
T	F	T	T	T	0.9
F	F	T	T	T	0.1
⋮	⋮	⋮	⋮	⋮	⋮

P(X2 | X6)

X2	X6	P
T	T	0.2
F	T	0.8
T	F	1.0
F	F	0.0

Estimating Probabilities from a Database

Given a sufficiently large database with tupels $\underline{a}^{(1)}$ ... $\underline{a}^{(N)}$ of an unknown distribution $P(\underline{X})$, we can compute maximum likelihood estimates of all partial joint probabilities and hence of all conditional probabilities.

$X_{m_1}, \dots, X_{m_K}$ = subset of $X_1, \dots X_L$ with $K \le L$

$w_{\underline{a}}$ = number of tuples in database with $X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}$

N = total number of tuples

Maximum likelihood estimate of $P(X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K})$ is

$$P'(X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}) = w_{\underline{a}} / N$$

If a priori information is available, it may be introduced via a bias $m_{\underline{a}}$:

$$P'(X_{m_1}=a_{m_1}, \dots, X_{m_K}=a_{m_K}) = (w_{\underline{a}} + m_{\underline{a}}) / N$$

Often $m_{\underline{a}} = 1$ is chosen for all tupels $\underline{a}$ to express equal likelihoods in the case of an empty database.

Expectation Maximization (1)

Recommended reading: Borgelt & Kruse, Graphical Models, Wiley 2002

Often databases are neither complete (insufficient samples, missing attributes) nor precise (ambiguous or uncertain values). In this case Expectation Maximation (EM) provides an iterative procedure to estimate probabilities.

1. Imprecise data

Given a tuple with ambiguous attributes

$$\underline{a}^T = [\ \{a_{11}, a_{12}, \ldots\}, \{a_{21}, a_{22}, \ldots\}, \ldots, \{a_{K1}, a_{K2}, \ldots\}\ ]$$

and number of occurrence $w_{\underline{a}}$, redistribute $w_{\underline{a}}$ equally among all combinations of attribute values.

2. Incomplete database

Execute iterative 2-step procedure:

A Compute sample frequencies from estimated probabilities

B Estimate probabilities from samples, maximizing likelihood of data (see previous slide)

Expectation Maximization (2)

Expectation step of EM:

Use current (initial) probability estimates to compute probability P($\underline{a}$) for all attribute combinations $\underline{a}$.

For Bayes Nets, this requires computing P($\underline{a}$) from the conditional probabilities assigned to the nodes.

At the initial step, absolute frequencies of missing attribute tuples $\underline{a}$* are completed:

$\underline{a}_1 = [X_1{=}a_1\,,\ X_2{=}a_{m2},\ X_3{=}a_{m3},\ \ldots\ ] \quad w_{\underline{a}*} \bullet P(\underline{a}_1)$

$\underline{a}_2 = [X_1{=}a_2\,,\ X_2{=}a_{m2},\ X_3{=}a_{m3},\ \ldots\ ] \quad w_{\underline{a}*} \bullet P(\underline{a}_2)$

$\underline{a}^* = [\ *\ ,\ X_2{=}a_{m2},\ X_3{=}a_{m3},\ \ldots\ ] \ \ w_{\underline{a}*}$

$\vdots$

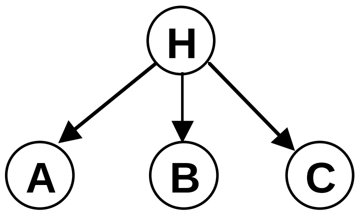
$\underline{a}_M = [X_1{=}a_M\,,\ X_2{=}a_{m2},\ X_3{=}a_{m3},\ \ldots\ ] \quad w_{\underline{a}*} \bullet P(\underline{a}_M)$

missing attribute **absolute frequency** **completed database**

Example for Expectation Maximization (1)

(adapted from Borgelt & Kruse, Graphical Models, Wiley 2002)

Given 4 binary probabilistic variables A, B, C, H with known dependency structure:



Given also a database with tuples [* A B C] where H is a missing attribute.

H	A	B	C	w
*	T	T	T	14
*	T	T	F	11
*	T	F	T	20
*	T	F	F	20
*	F	T	T	5
*	F	T	F	5
*	F	F	T	11
*	F	F	F	14

absolute frequencies of occurrence

Estimate of the conditional probabilities of the Bayes Net nodes !

Example for Expectation Maximization (2)

Initial (random) probability assignments:

H	P(H)
T	0.3
F	0.7

A	H	P(A|H)
T	T	0.4
T	F	0.6
F	T	0.6
F	F	0.4

B	H	P(B|H)
T	T	0.7
T	F	0.8
F	T	0.3
F	F	0.2

C	H	P(C|H)
T	T	0.8
T	F	0.5
F	T	0.2
F	F	0.5

With $$P(H \mid A,B,C) = \frac{P(A \mid H) \bullet P(B \mid H) \bullet P(C \mid H) \bullet P(H)}{\sum_H P(A \mid H) \bullet P(B \mid H) \bullet P(C \mid H) \bullet P(H)}$$

one can complete the database:

H	A	B	C	w
T	T	T	T	1.27
T	T	T	F	3.14
T	T	F	T	2.93
T	T	F	F	8.14
T	F	T	T	0.92
T	F	T	F	2.37
T	F	F	T	3.06
T	F	F	F	8.49

H	A	B	C	w
F	T	T	T	12.73
F	T	T	F	7.86
F	T	F	T	17.07
F	T	F	F	11.86
F	F	T	T	4.08
F	F	T	F	2.63
F	F	F	T	7.94
F	F	F	F	5.51

Example for Expectation Maximization (3)

Based on the modified complete database, one computes the maximum likelihood estimates of the conditional probabilities of the Bayes Net.

Example: $$P(A = T \mid H = T) \approx \frac{1.27 \cdot 3.14 \cdot 2.93 \cdot 8.14}{1.27 \cdot 3.14 \cdot 2.93 \cdot 8.14 \cdot 0,92 \cdot 2.73 \cdot 3.06 \cdot 8.49} \approx 0.51$$

This way one gets new probability assignments:

H	P(H)
T	0.3
F	0.7

A	H	P(A|H)
T	T	0.51
T	F	0.71
F	T	0.49
F	F	0.29

B	H	P(B|H)
T	T	0.25
T	F	0.39
F	T	0.75
F	F	0.61

C	H	P(C|H)
T	T	0.27
T	F	0.60
F	T	0.73
F	F	0.40

This completes the first iteration. After ca. 700 iterations the modifications of the probabilities are less than 10^{-4}. The resulting values are

H	P(H)
T	0.5
F	0.5

A	H	P(A|H)
T	T	0.5
T	F	0.8
F	T	0.5
F	F	0.2

B	H	P(B|H)
T	T	0.2
T	F	0.5
F	T	0.8
F	F	0.2

C	H	P(C|H)
T	T	0.4
T	F	0.6
F	T	0.6
F	F	0.4

Summary

Computer Vision Across Abstraction Levels

symbolic descriptions
abstract concepts,
concepts for occurrences,
plans, predictions

qualitative descriptions
object categories,
qualitative spatial and
temporal relations

quantitative descriptions
metric conceptual spaces,
perceptual primitives

physical signals
sensor input

recognizing high-level concepts in lower-level descriptions

exploiting high-level context for lower-level analysis

Representation and interpretation formalisms must support integration across abstraction levels

Aggregates as Basic Representational Units

Probabilistic view	Relational view	Logical view
Object configurations in space and time governed by joint probabilities	Frames with part-of structure embedded in taxonomical and part-of hierarchies	Expressions of a logic-based concept language
• Bayes Net representations • part-whole reasoning by probabilistic inference • uncertainty management for hypothesis formation • concept learning	• intuitive object-centered notation • highly expressive • work-horse for knowledge representation • established tools for configuration methodology	• well-founded inference procedures • image interpretation as abduction • inductive learning • logic-based reasoning about actions • temporal and spatial logics

Integrating Aggregates and Bayes Nets

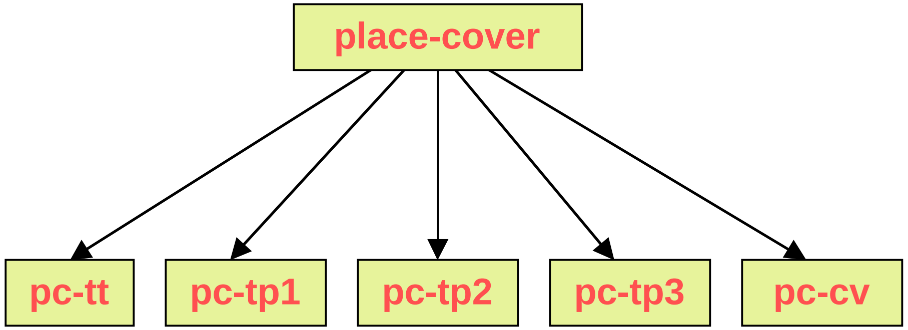
frame

NAME
place-cover is-a agent-activity

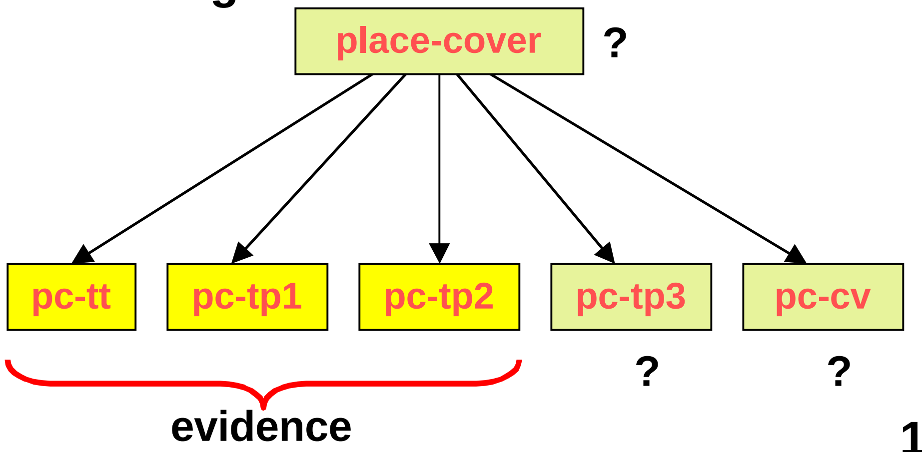
PARTS
pc-tt is-a table-top
pc-tp1 is-a transport
with (tp-obj is-a plate)
pc-tp2 is-a transport
with (tp-obj is-a saucer)
pc-tp3 is-a transport
with (tp-obj is-a cup)
pc-cv is-a cover

CONSTRAINTS
<identity constraints on parts>
<spatial constraints on parts>
<temporal constraints on parts>

Bayes Net



Probabilistic inference for part-whole reasoning:



Integrating Aggregates and Description Logics

Frame-based notation

```
NAME
place-cover is-a agent-activity

PARTS
pc-tt is-a table-top
pc-tp1 is-a transport
                with (tp-obj is-a  plate)
pc-tp2 is-a transport
                with (tp-obj is-a  saucer)
pc-tp3 is-a transport
                with (tp-obj is-a  cup)
pc-cv is-a  cover

CONSTRAINTS
<identity constraints on parts>
<spatial constraints on parts>
<temporal constraints on parts>
```

Concept expressions of the Description Logic ALCF(D)

name

roles

concrete domain predicates

```
(equivalent  place-cover
    (and  agent-activity
        (some  pc-tt  table-top)
        (some  pc-tp1
            (and  transport
                (some tp-obj plate) )
        (some  pc-tp2  transport)
            (and  transport
                (some tp-obj saucer) )
        (some  pc-tp3  transport)
            (and  transport
                (some tp-obj cup) )
        (some  pc-cv  cover)

        <feature agreement constraints>
        <admissible numerical constraints>
```

Extending Symbolic Space into Metric Space

ALCF(D)

symbolic domain

metric domain

- space
- time
- colour
- shape
- ...

signal domain

(Gärdenfors 2000)

reasoning may incorporate measures of distance and similarity

Examples:

- interval durations in occurrence models
- distances in spatial relations

Reasoning services available in Description Logics

Standard reasoning services

- consistency check
- subsumption check
- classification
- abstraction
- default reasoning

available in optimized reasoning systems, e.g. in the system RACER

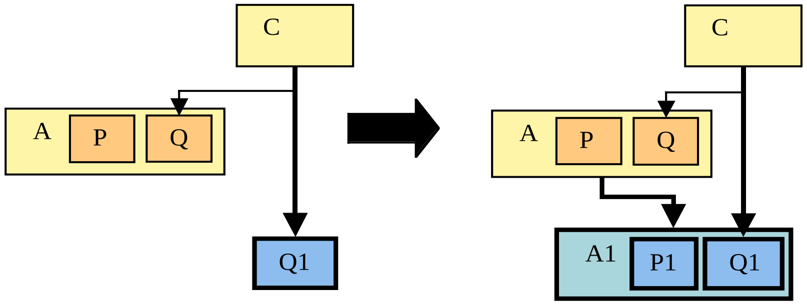
RACER User´s Guide and Reference Manual Version 1.7, http://www.fh-wedel.de/˘mo/3214/racer-manual-1-7.pdf

Extensions of reasoning services necessary for high-level vision:

- **temporal reasoning**
- **spatial reasoning**
- **generating specialization hypotheses**
- **generating aggregate hypotheses**

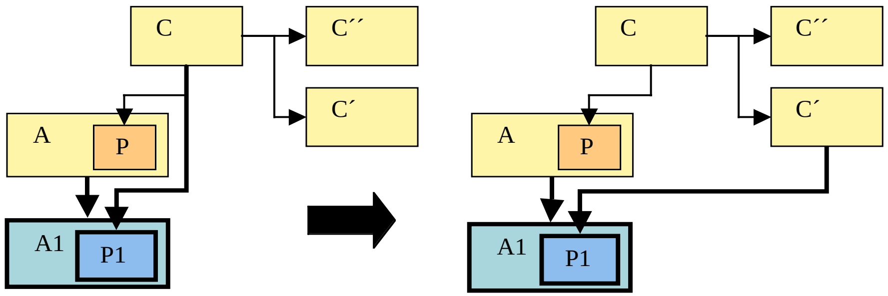
Designed, partially implemented

Bayes-net guided Interpretation Steps



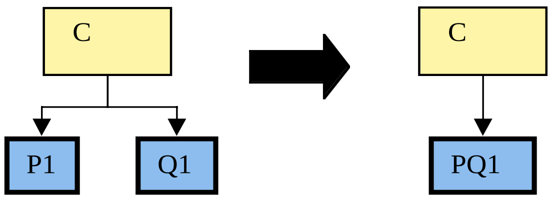
Aggregate instantiation

Aggregate A1 is instantiated based on instance Q1



Instance refinement

Instance P1 in aggregate A1 is specialised from concept C to C´



Instance merging

Instances P1 and Q1 are merged to PQ1

Conclusions

- Generic high-level image sequence interpretation requires model-based approach
- Specialisation and aggregation hierarchies support efficient navigation in interpretation space
- Spatial, temporal and task context is modelled by instantiated high-level aggregates
- Temporal and spatial constraints require dedicated constraint satisfaction mechanisms
- Statistics of vision memory may feed Bayes Net for hypotheses ranking