Reasoning Methods for Image Sequence Interpretation

Bernd Neumann

Cognitive Systems Laboratory
Hamburg University
Germany

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

Contents

- Scope of tutorial
- Historical examples of high-level motion interpretation
- Relational matching
- Rule-based interpretation
- Description Logics
- Image interpretation as deduction
- Logics of image interpretation
- Image interpretation as configuration
- Signal-symbol interface
- Navigating in hallucination space
- Bayes Nets
- Conclusions

Scope of tutorial

Some application scenarios for high-level image sequence interpretation

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

Characteristics of high-level image 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

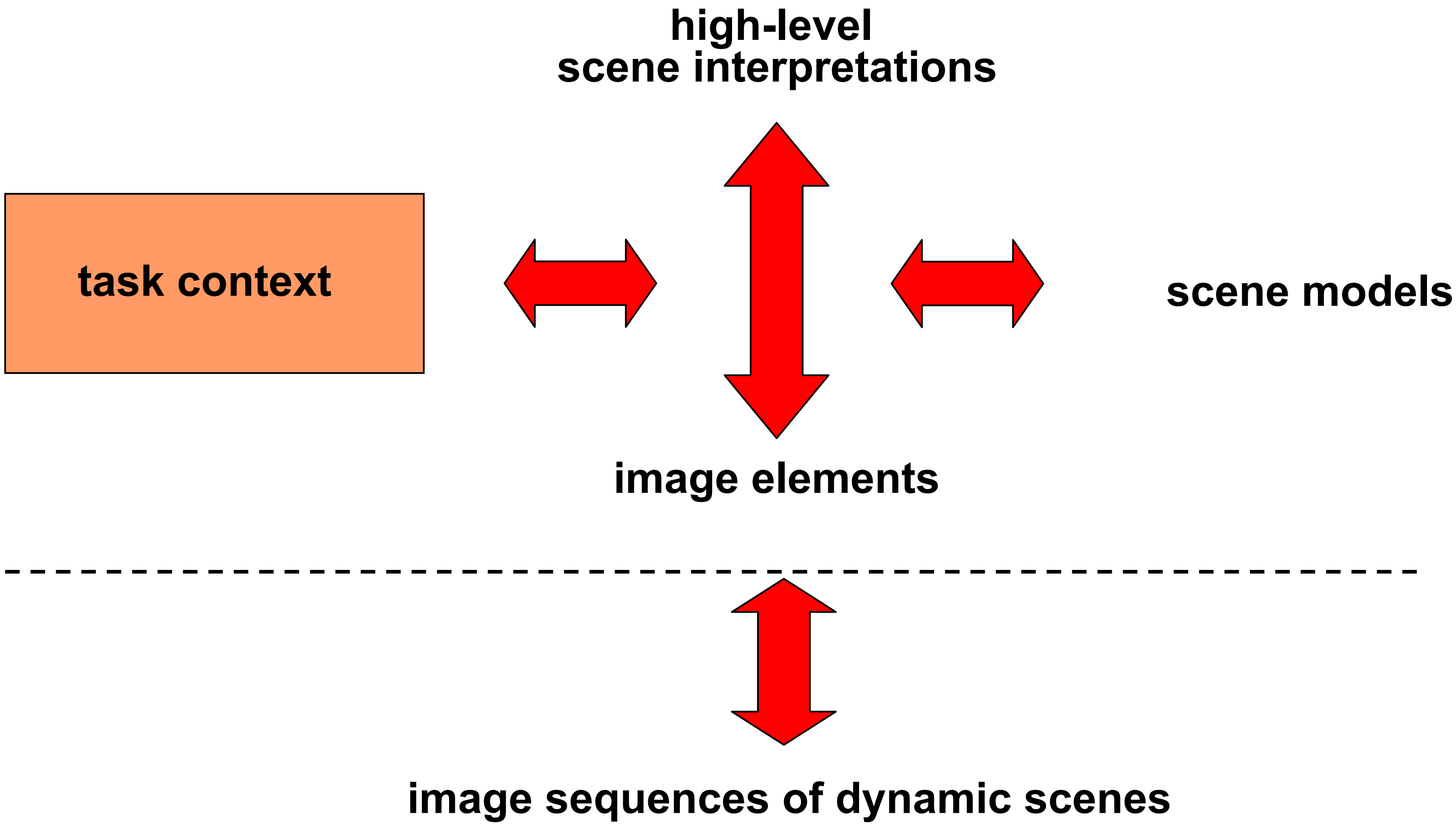
Context and task dependence

Interpretations may depend on

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

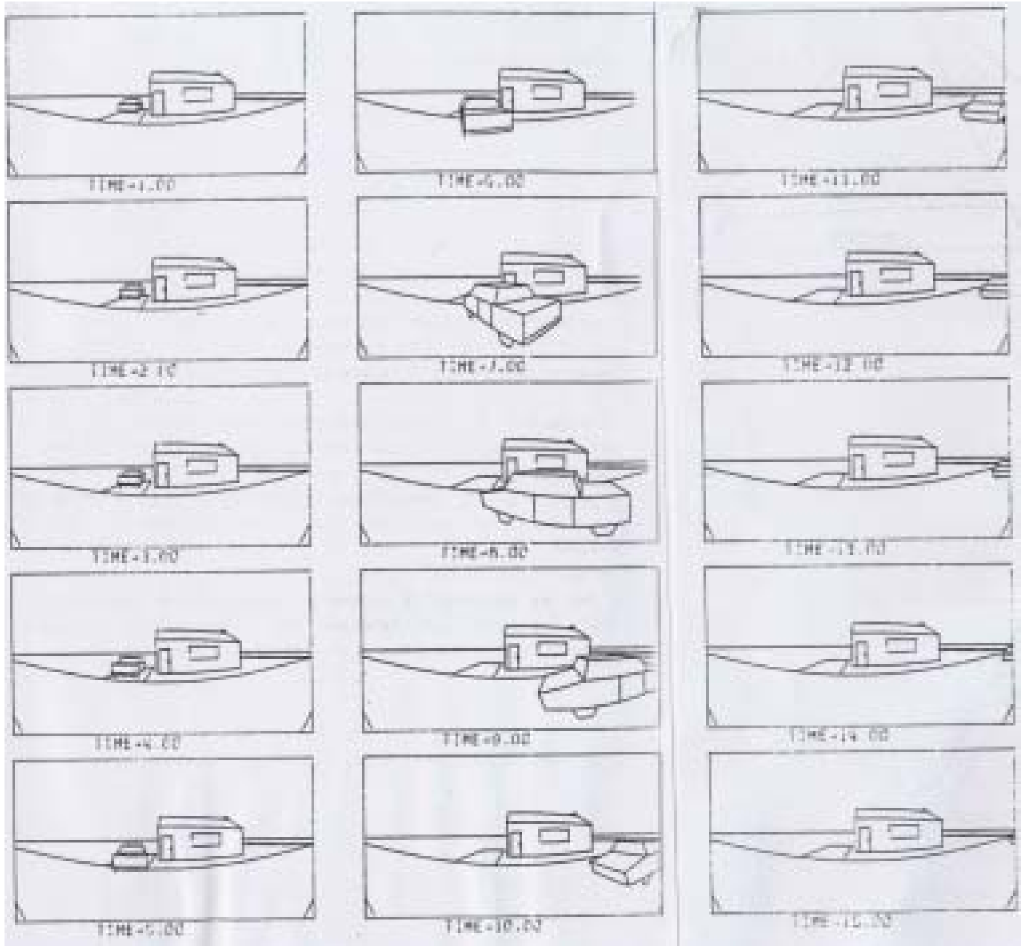
Constructing an interpretation is <u>not</u> a mapping from image data into interpretation space.

High-level scene interpretation



Historical examples

Early traffic scene analysis (Badler 75)



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

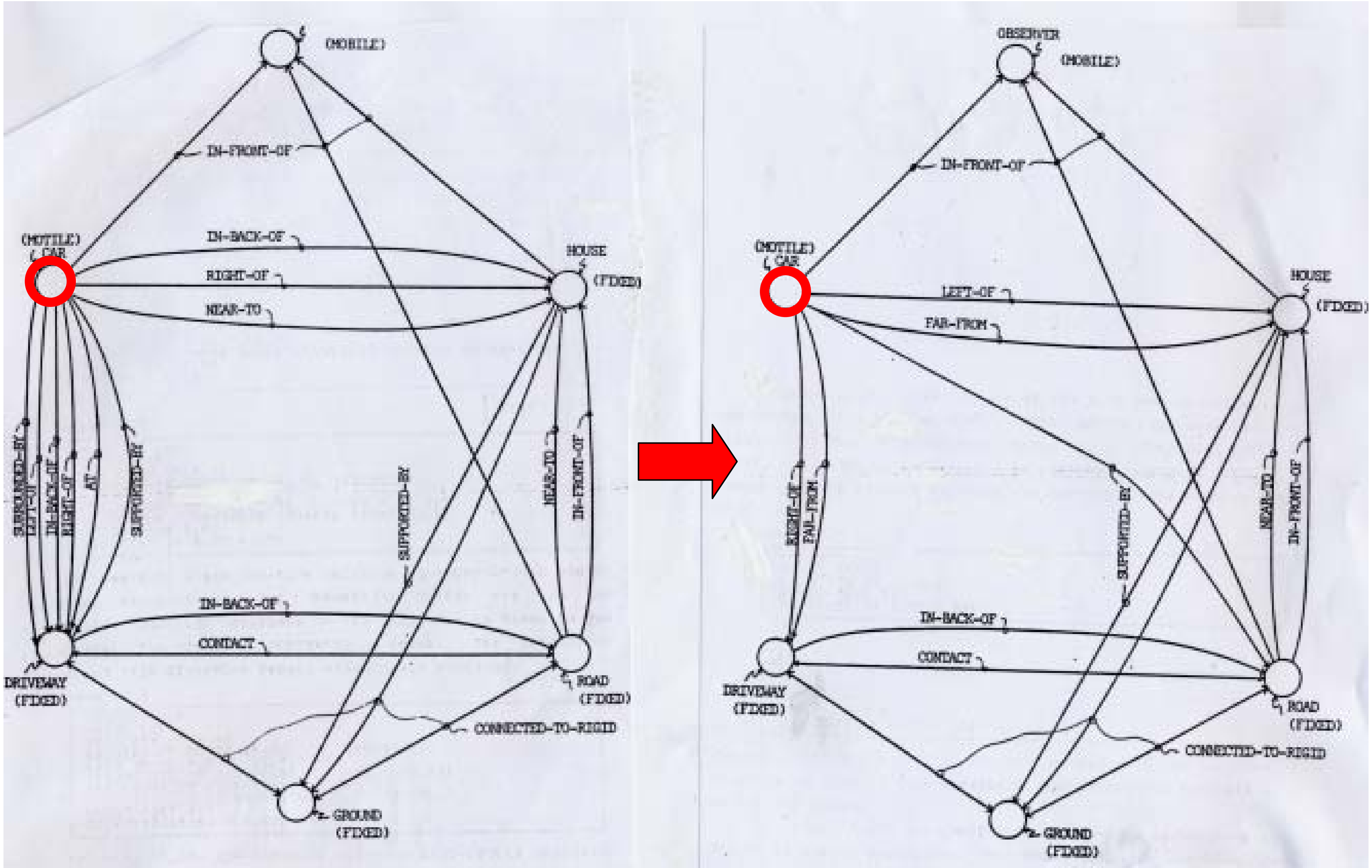
Directional adverbials for motion description (Badler 75)

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)

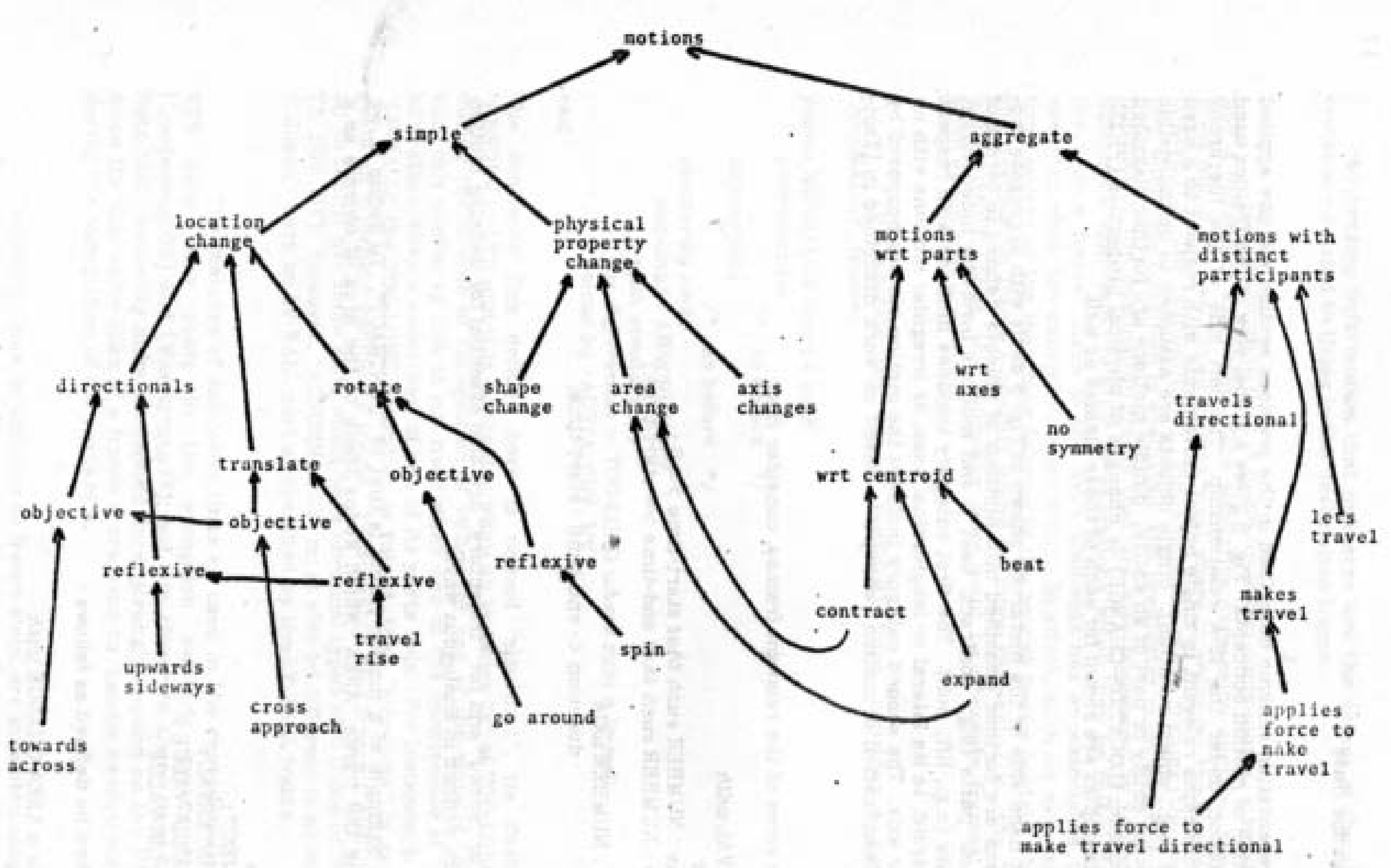
A NEAR-TO relation with one side of an object is broken and replaced by a similar relation with the other side. There is an implicit sense of passage ABOVE the object.

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.

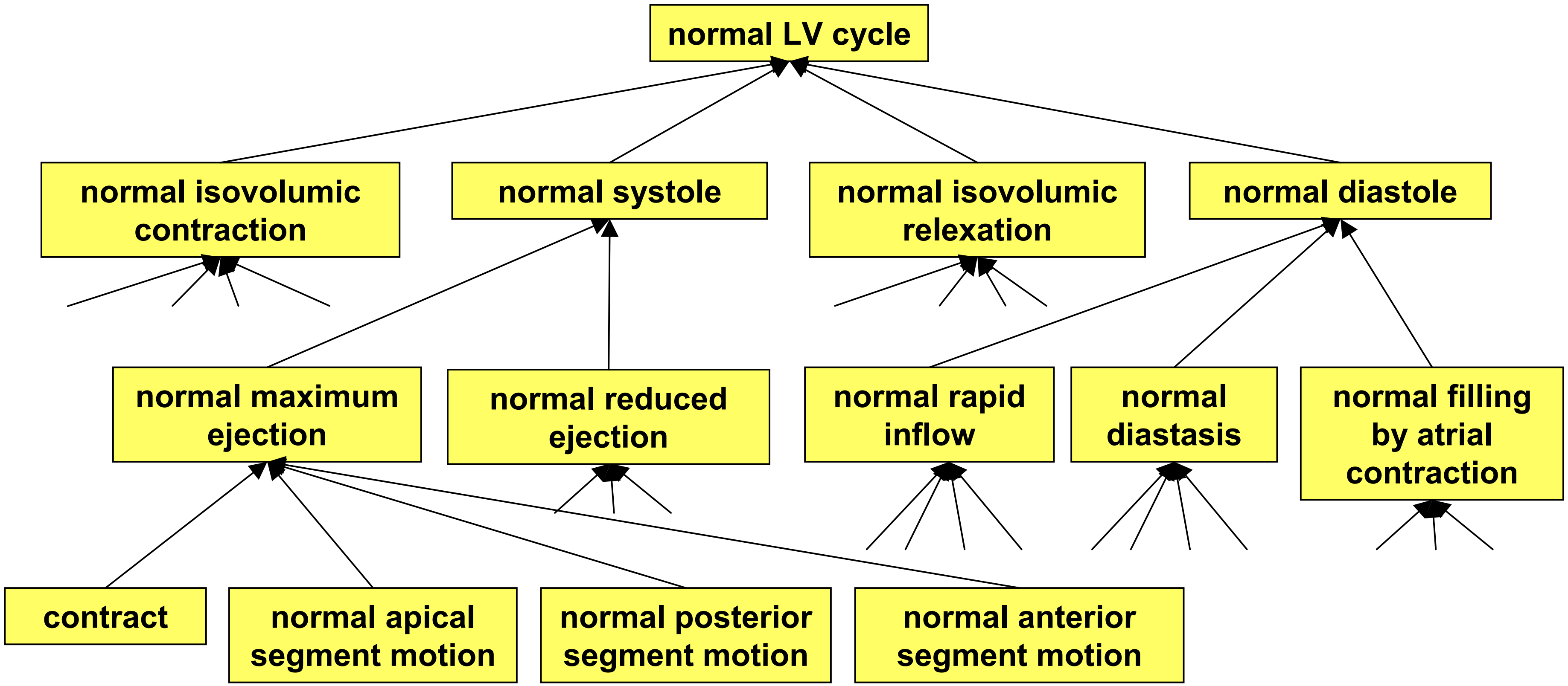
Postcondition
SUBJECT X
DIRECTION PCONS((ACROSS Y), DIRECTION)

Motion IS-A hierarchy (Tsotsos 79)



Left-ventricular motion PART-OF hierarchy (Tsotsos 79)

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

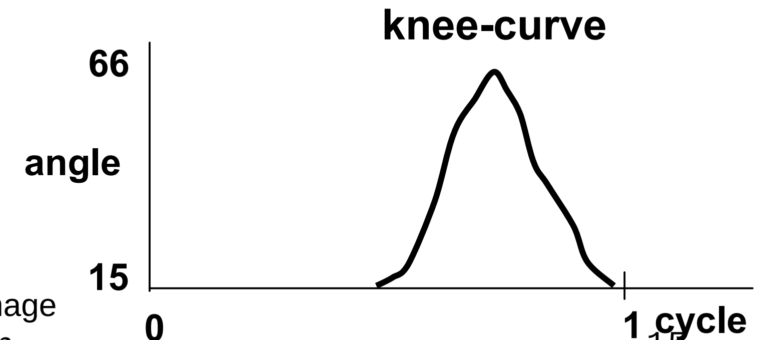


Model-based prediction for tracking a jointed moving object (Hogg 84)

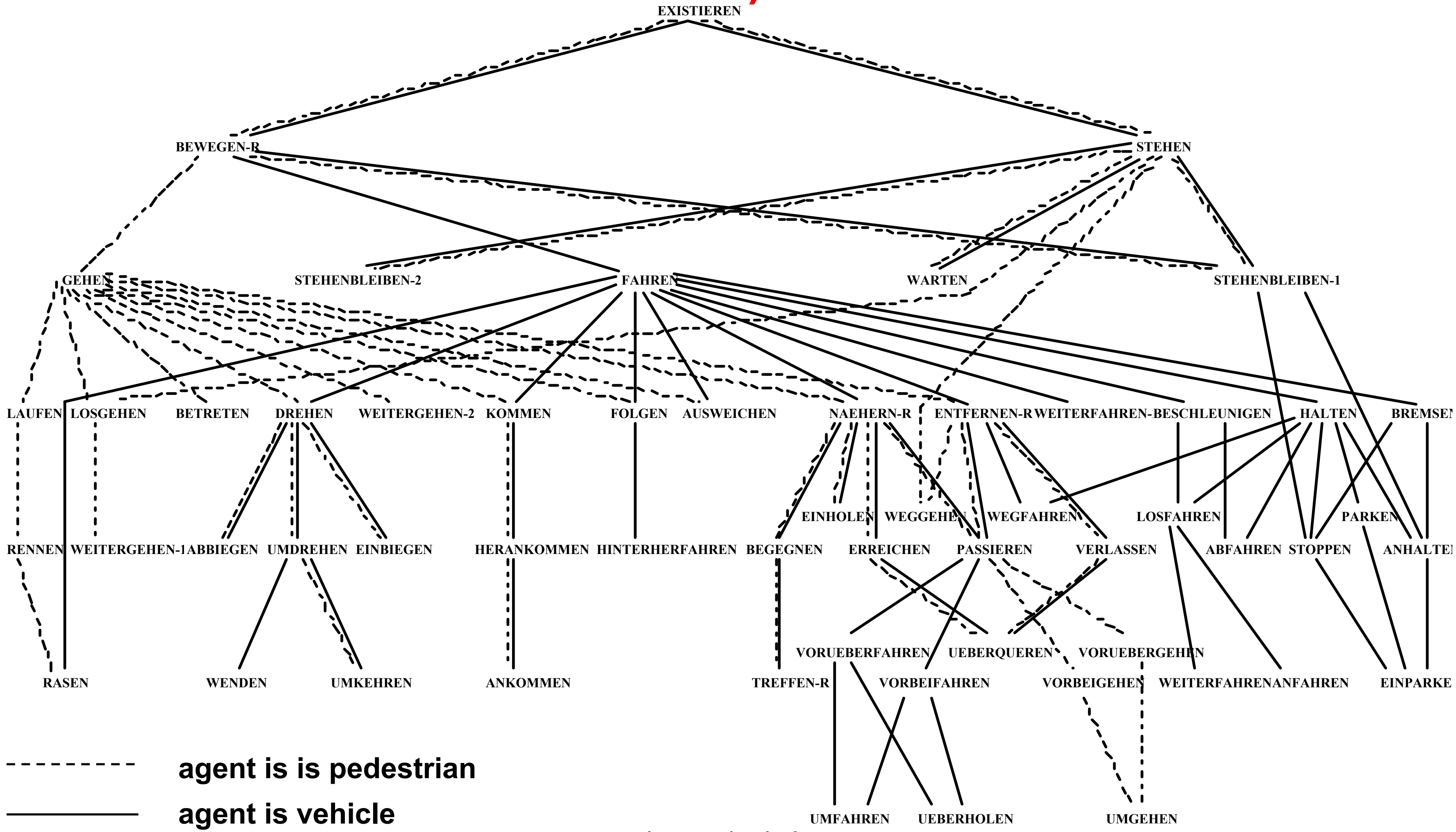
The case of highly coordinated motion of parts

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

Example:



Verb hierarchy for traffic scenes (Neumann & Novak 86)



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

- **observations provide begin and end time points of occurrences**
- **models express qualitative constraints on time points**

Choice of convex time point algebra [Vila 94]

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 **during** I_2 => $I_2.tb \leq I_1.tb$

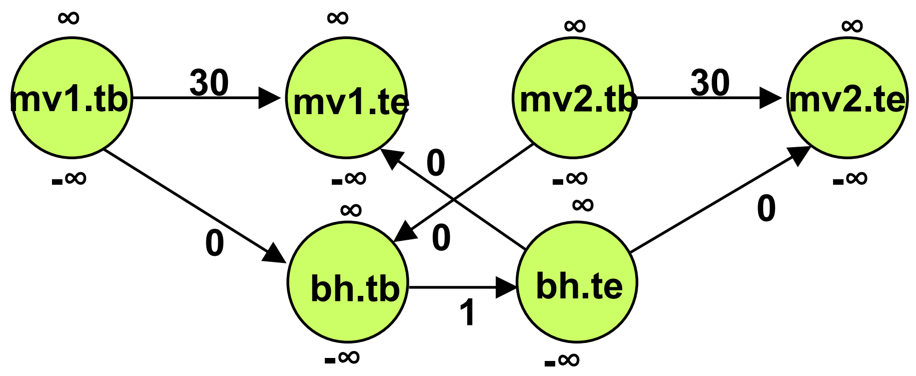
$I_1.te \leq I_2.te$

There exist efficient techniques for incremental evaluation of convex time point algebra constraints.

Constraint propagation for occurrence verification (1)

Example: "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)

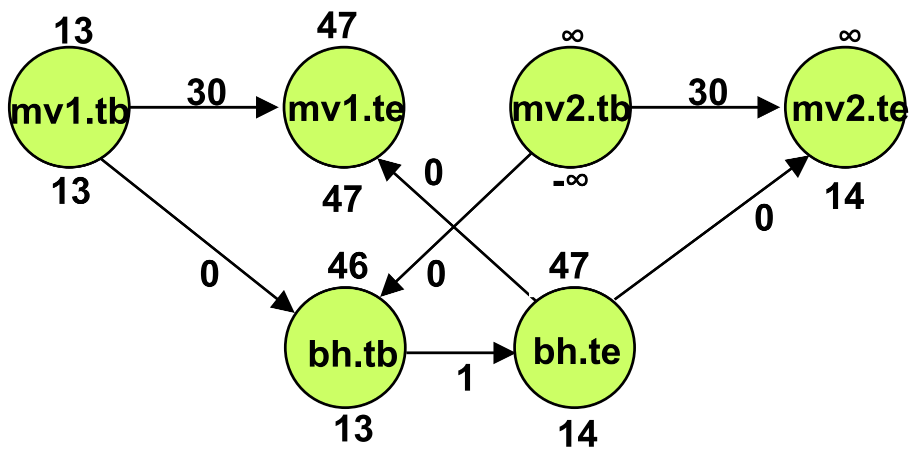
3. **Instantiate parts in occurrence model**

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 $t_{1max}' = \min \{t_{1max}, t_{2max} - c_{12}\}$

Example: move1 in scene instantiates mv1 of model

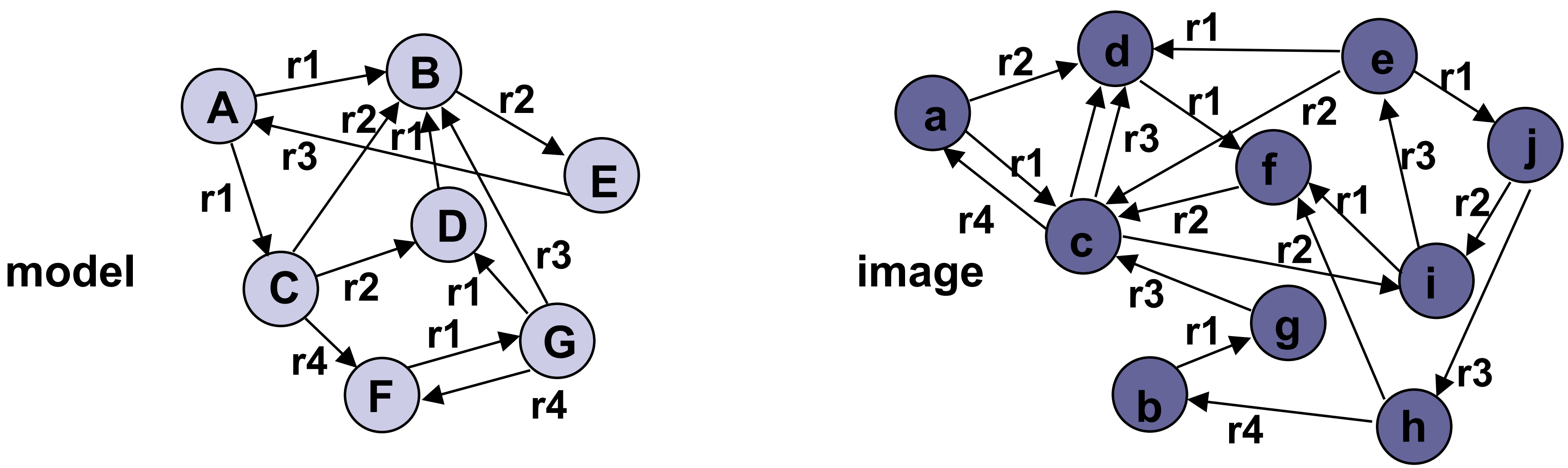
ID:	move1
instance:	move
parts:	mv-ob = obj1
	mv-tr = trj1
times:	mv-tb = 13
	mv-te = 47



Relational Matching

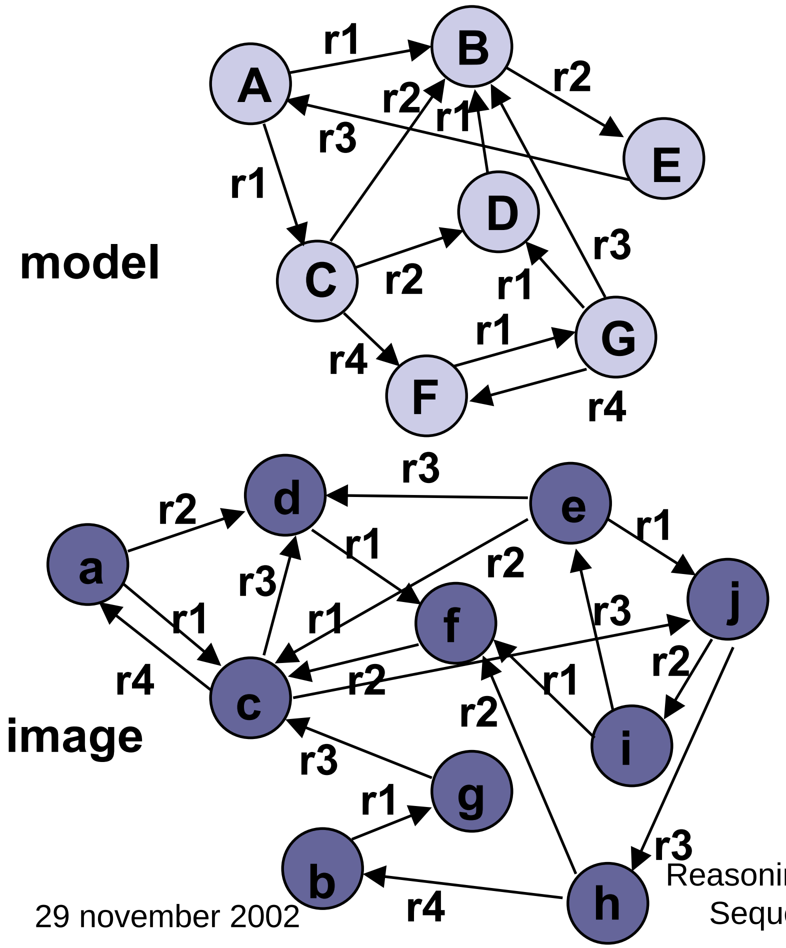
Relational models

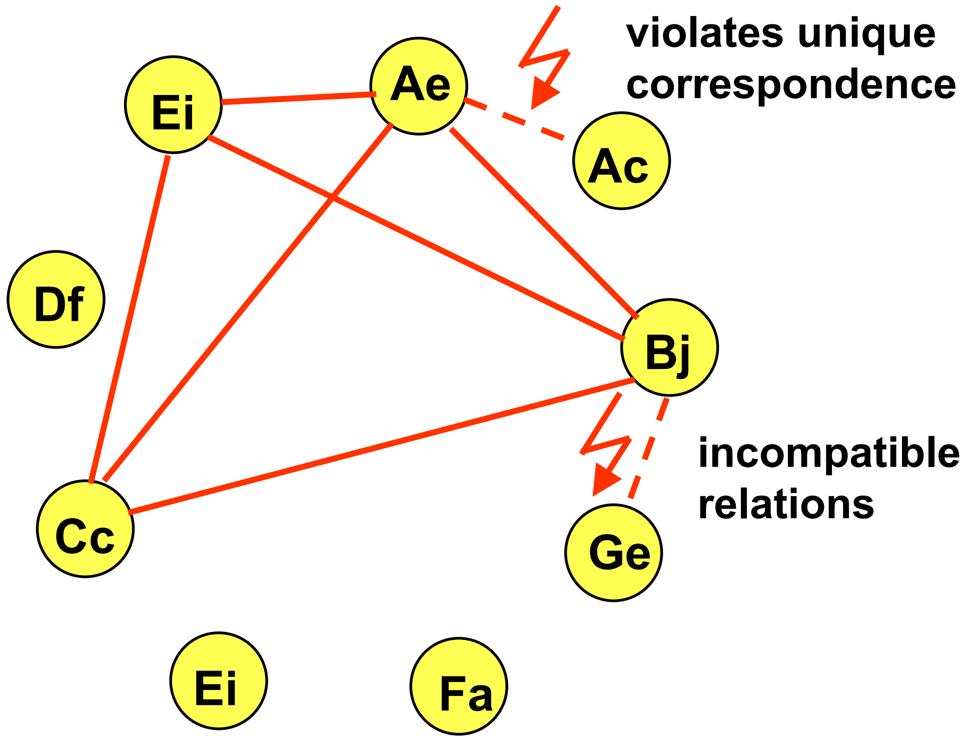
- relational models describe concepts (aggregates) in terms of parts (components) and relations between the parts
- interpretation is R-morphism (best partial match) between image and model(s)
- search for best partial match is based on "compatibility" of nodes and edges



Relational match 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 complete)

Finding maximal cliques

clique = complete subgraph

Find maximal cliques in a given compatibility graph

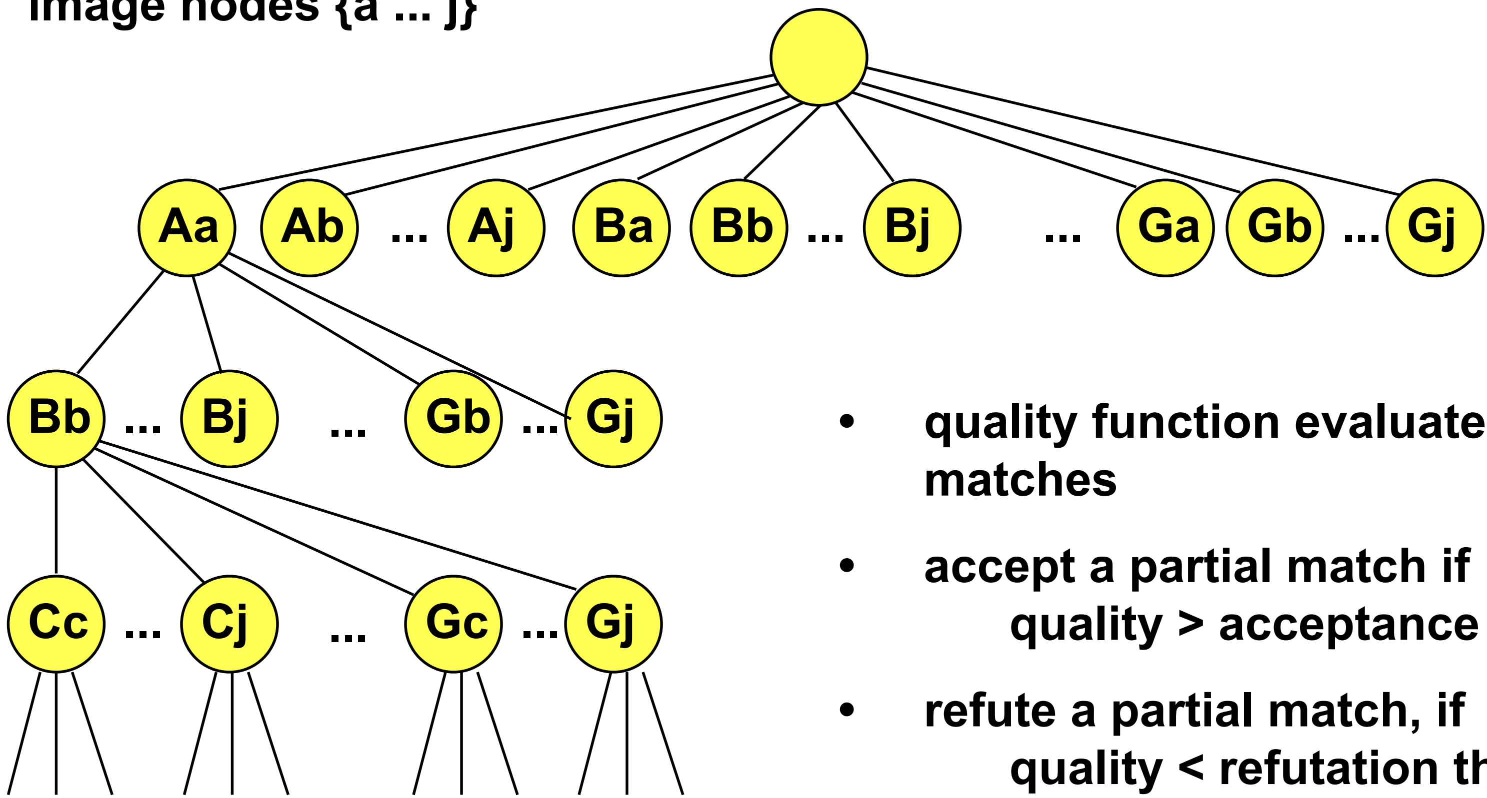
Algorithms are available in the literature, e.g.

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

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**

- **differentiated compatibility measure required**
  - fuzziness
  - compatibilty vs. consistency
  - probabilities
- **laws for temporal, spatial, physical relations**
- **uncertainty management**
- **no multi-level aggregate structure**

Rule-based interpretation

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 knowledge-based approach?

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

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)**

Important aspects of DL development

- **trade-off between expressiveness of terminology and complexity of reasoning services**
- **desirable features may easily lead to undecidability**
- **concrete domains must be incorporated to support spatial and temporal reasoning**
- **implementation must be highly optimized to be useful**
- **DL community must be pushed to deal with vision**

Family of Description Logics

AL
Attribute Language ∀ ∩

ALC
Complement

ALNFIh *(CLASSIC)*
Number restrictions (≥ n r) (≤ n r)
Features with same-as, Inverse
hierarchies with single inheritance

ALCNF *(KRIS)*
Number restrictions (≥ n r) (≤ n r)
Features with same-as

ALCQRIFO *(LOOM)*
Qualified number restrictions (≥ n r C)(≤ n r C)
Role conjunction, Inverse roles
Features with same-as, One-of, fills

ALC(D)
concrete Domains D, P

ALCRP(D)
Roles defined wrt Predicates

ALCHfR+ *(FaCT)*
role Hierarchies with multiple parents
features without same-as
transitive Roles

ALCNHR+ *(RACE)*
role Hierarchies with multiple parents
Number restrictions (≥ n r) (≤ n r)
transitive Roles

ALCQHIR+ *(RACER)*
role Hierarchies with multiple parents
Qualified number restrictions (≥ n r C) (≤ n r C)
Inverse roles, transitive Roles, integers and reals

Representing N-ary Relations

(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)

instances of a relation are reified

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 ... Ci)

roles

R -> RN
(inv RN)

concept-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)

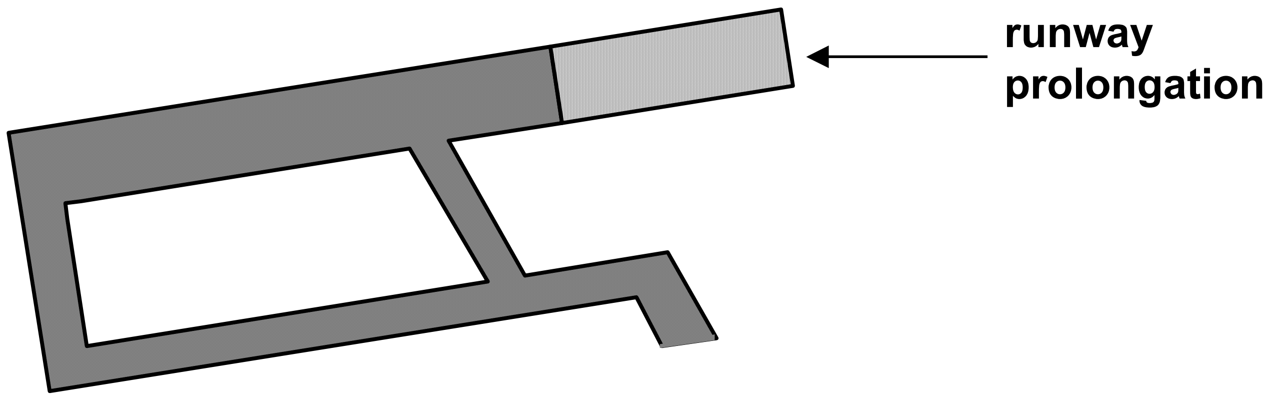
aexpr -> AN
real
(+ aexpr1 aexpr1*)
aexpr1

aexpr1 -> real
AN
(* real AN)

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

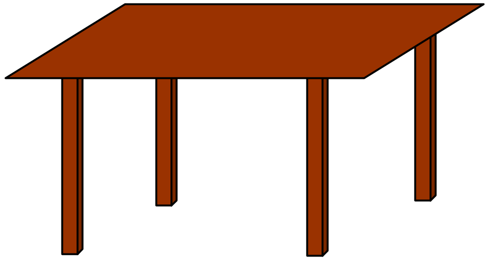
Problems:

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

Support of hypothesize-and-test cycle is required !

Logics of image interpretation

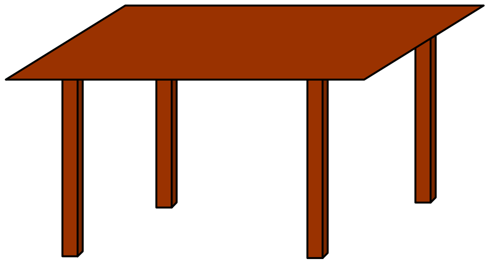
Describing image interpretation in logical terms



?

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 1990, 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).

Image interpretation as configuration

Image interpretation as a configuration problem

What is a configuration problem?

Construct an aggregate (a configuration) given

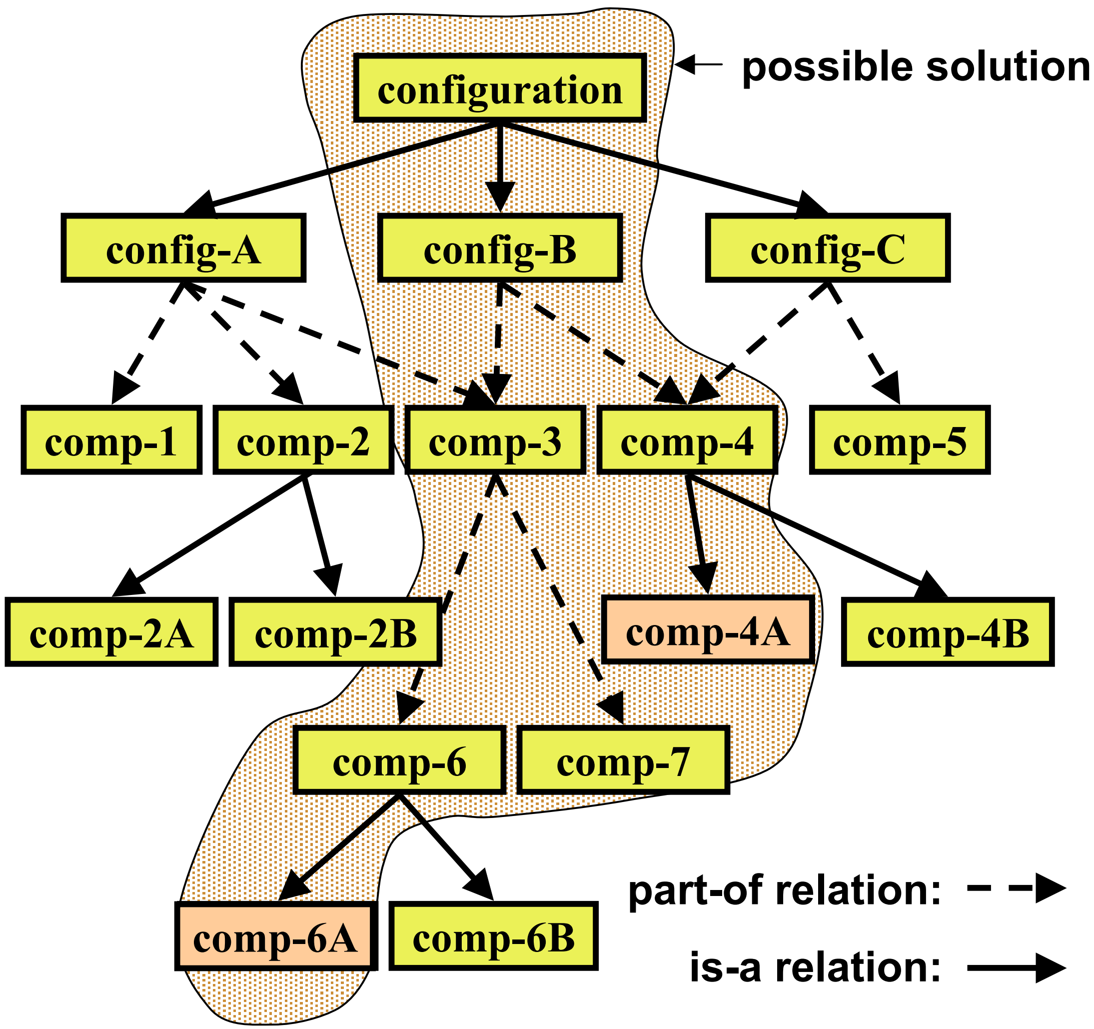
- **generic descriptions of parts**
- **compatibility constraints between parts**
- **a concrete task description**

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

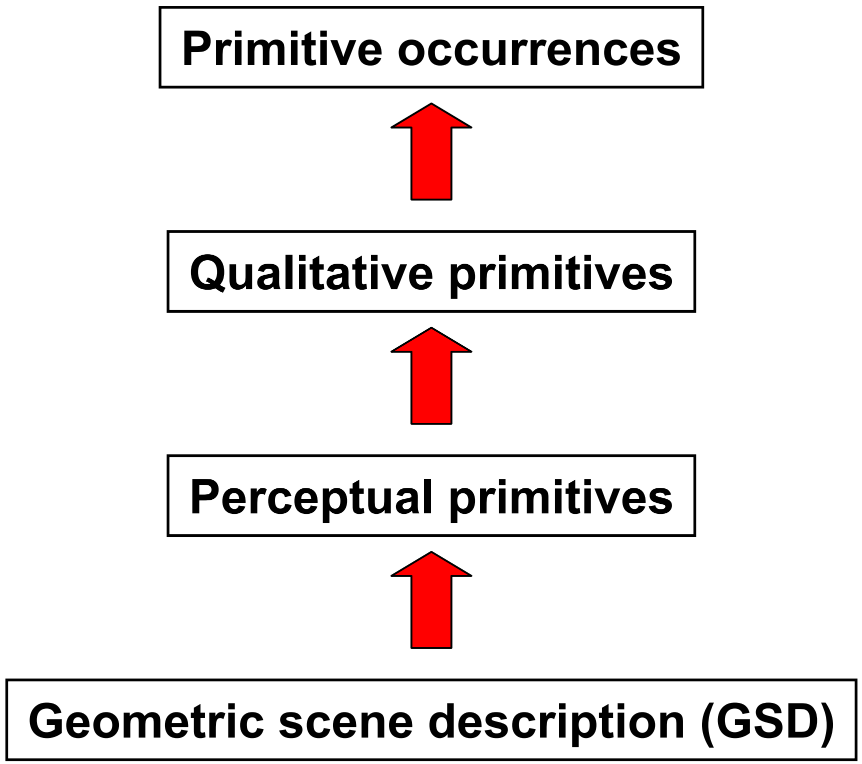
Illustration of configuration



- 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



Geometric scene description (GSD)

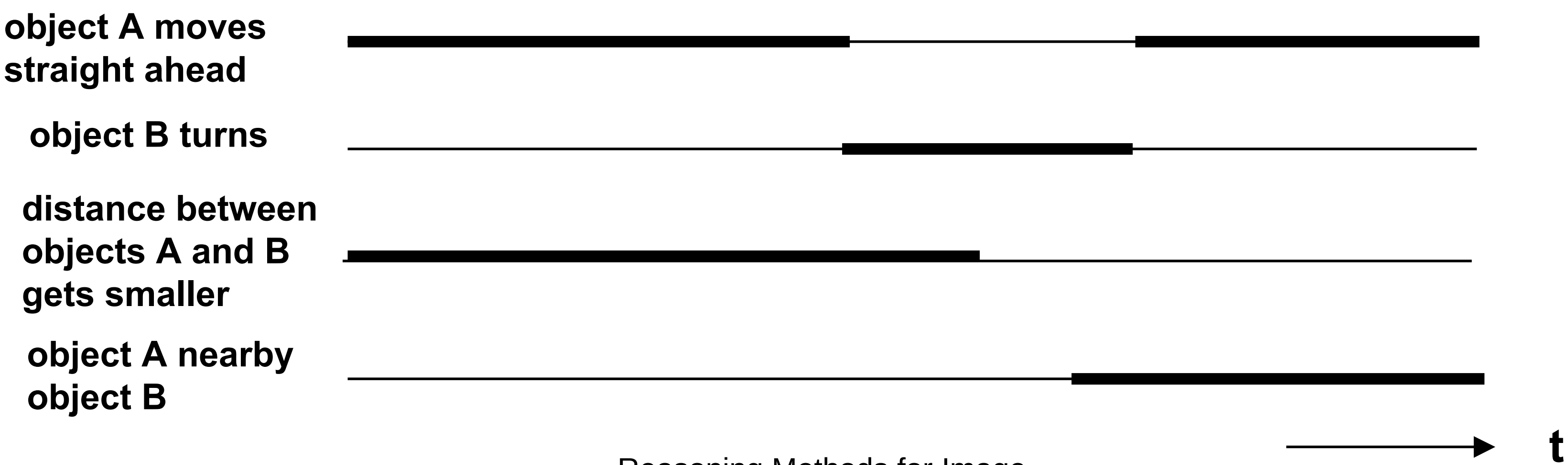
The GSD is a quantitative object-level scene interpretation in terms of
- **recognised objects and**
- **their (possibly time-varying) locations in the scene**

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

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.



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 gavity, 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

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**

  "Is there something on the table?"
  (after 30 min of processing)
  "Yes, a gold-rim plate, 112.4 mm diameter, position 324.3 mm off left table border, 24.8 mm off upper table border, orientation indeterminate, height above table-top 12.6 mm, ..."
- **interpretations must be "preferred"**
  - **aggregates vs. individual objects**
  - **most special concepts, basic categories, dissolved disjunctions**
  - **more likely vs. less likely interpretations**

Aggregates as basic representational units

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>

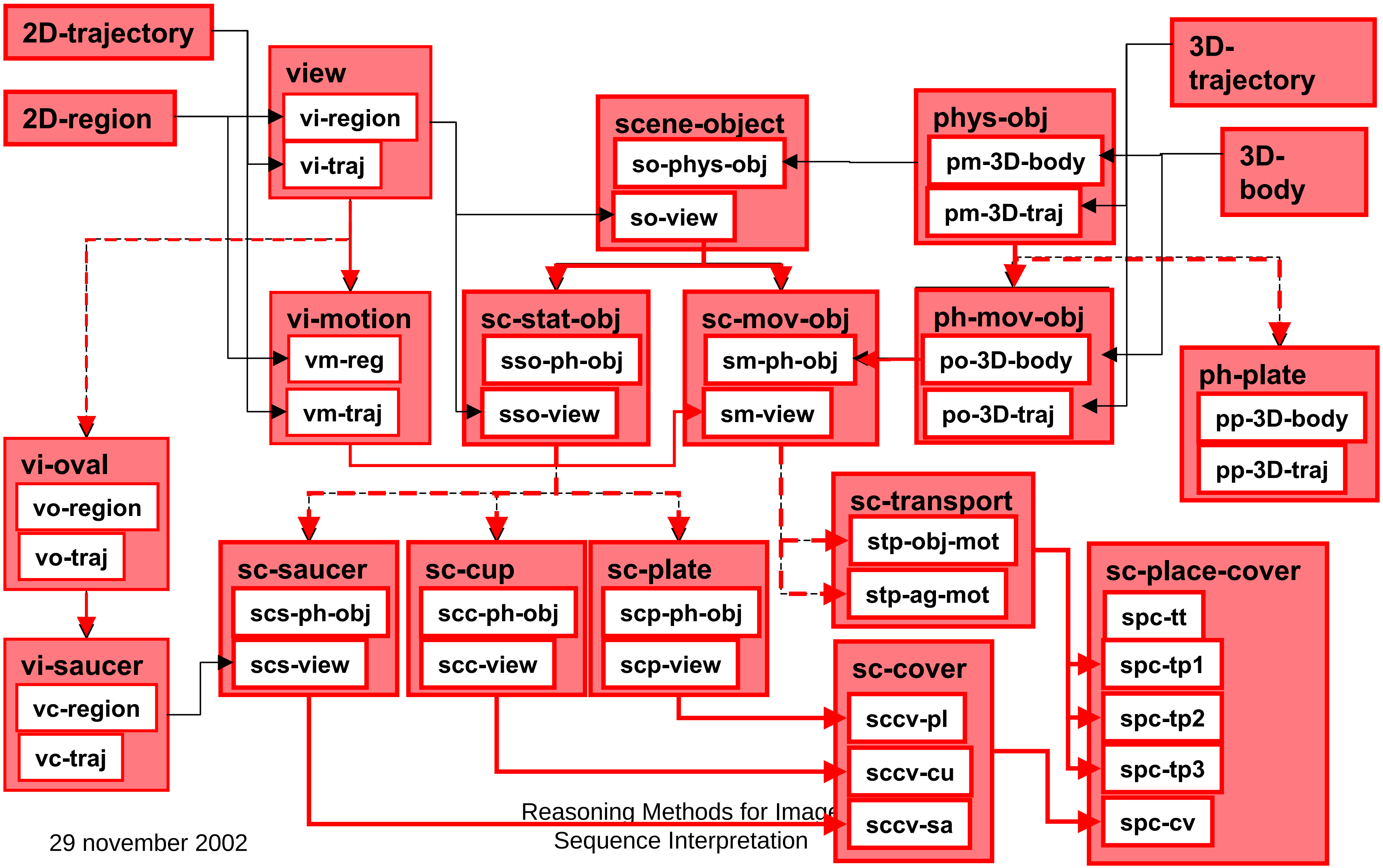
DL concept expressions

name:
(equivalent place-cover
(and agent-activity

roles:
(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)

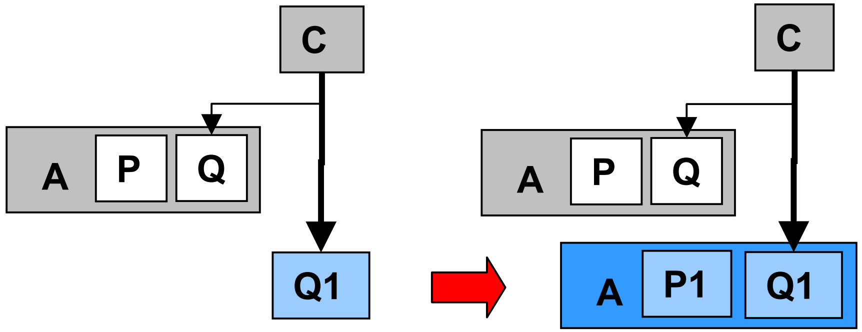
concrete domain predicates:
<identity constraints on parts>
<spatial constraints on parts>
<temporal constraints on parts>

Aggregates in taxonomical hierarchies

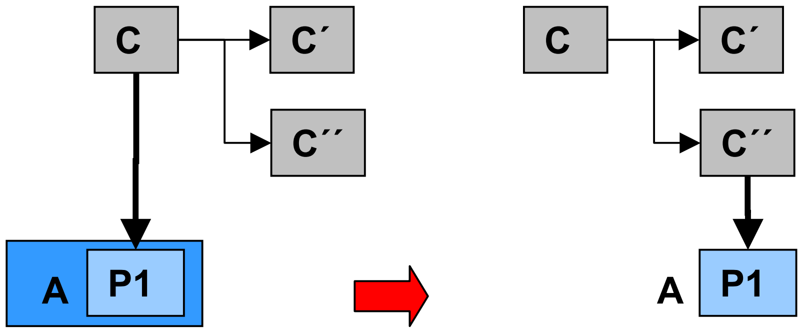


Interpretation steps

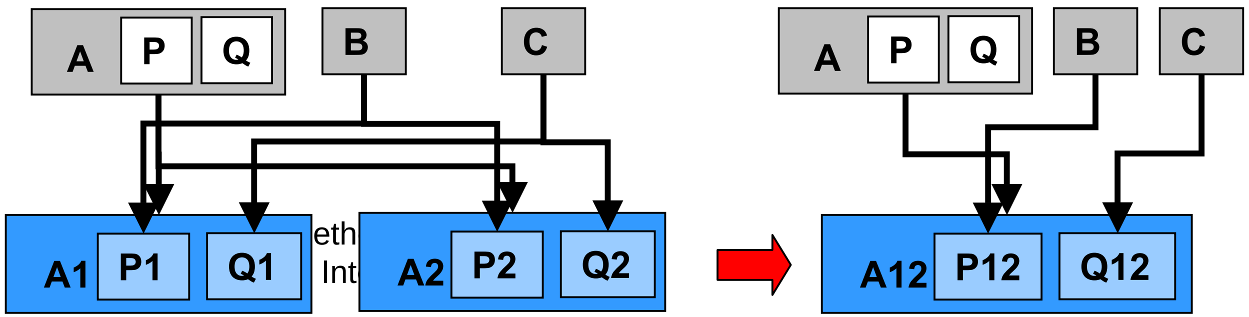
aggregate instantiation ("part-whole-reasoning")



instance refinement ("specialisation")



instance merging ("converging evidence")



Bayes Nets

Probabilistic models for occurrences

Modelling probabilistic dependencies (causalities) and independencies between discrete events

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

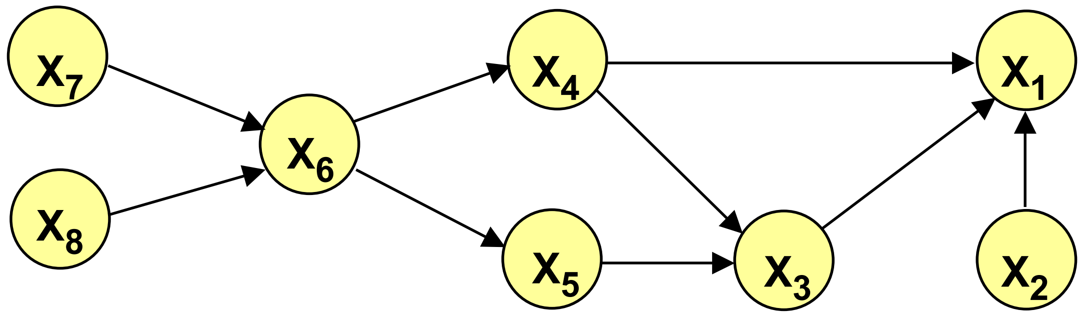
$$P(X \mid X_1, \dots X_{i-1}, X_i, X_{i+1}, \dots, X_n) = P(X \mid X_1, \dots X_{i-1}, X_{i+1}, \dots, X_n)$$

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:



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

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

Image interpretation with Bayes Nets

Constructing a Bayes Net:

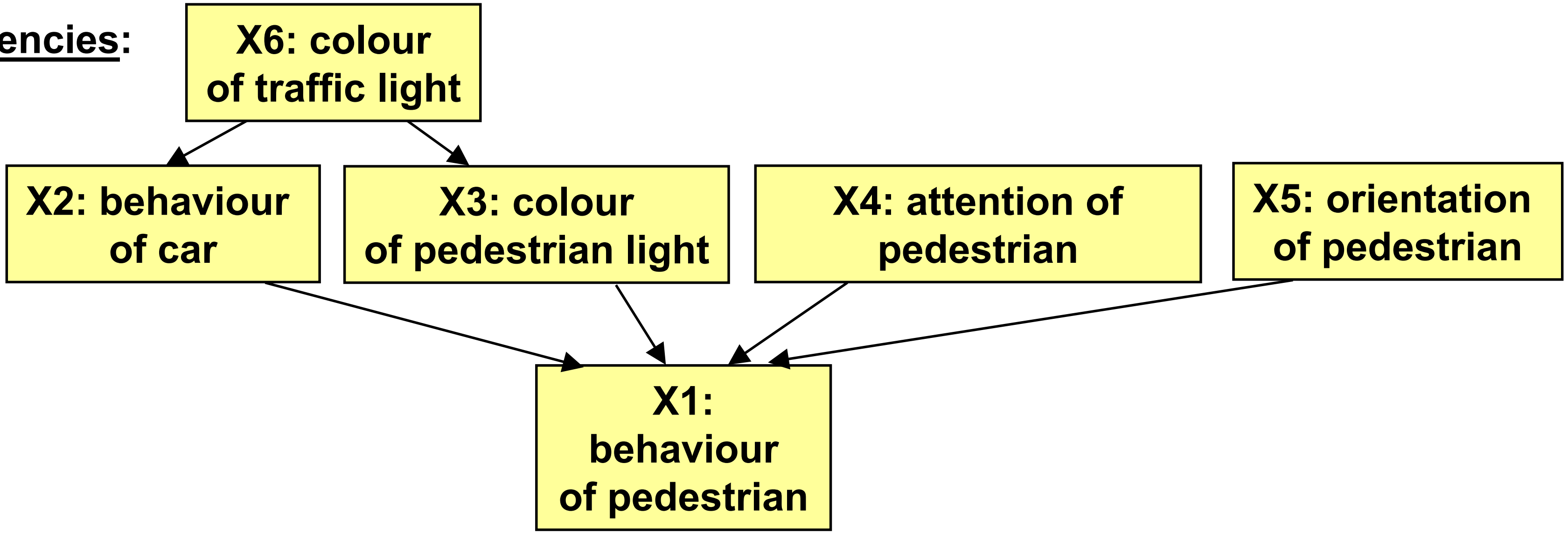
1. Select discrete variables X_i relevant for domain
2. Establish partial order of variables according to causality
3. In the order of decreasing causality:
   (i) Generate node X_i in net
   (ii) As predecessors of X_i choose the smalles subset of nodes which are already in the net and from which X_i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

Computing expected values:

1. Compute $P(X_i)$, $i = j, \ldots, k$ for the predecessors of a node X_m in the Bayes Net (recursively or from a priori knowledge or from image analysis uncertainty)
2. Compute $P(X_m) = P(X_m | X_j, \ldots, X_k) \bullet P(X_j) \ldots P(X_k)$

Example: Behaviour of pedestrians at traffic lights

Dependencies:



Conditional probabilities for concrete values of random variables must be known to compute expected values

Examples:

P(X1 = enters_street |
X2 = car_comes, X3 = red, X4 = inattentive, X5 = towards_street) = 0.8

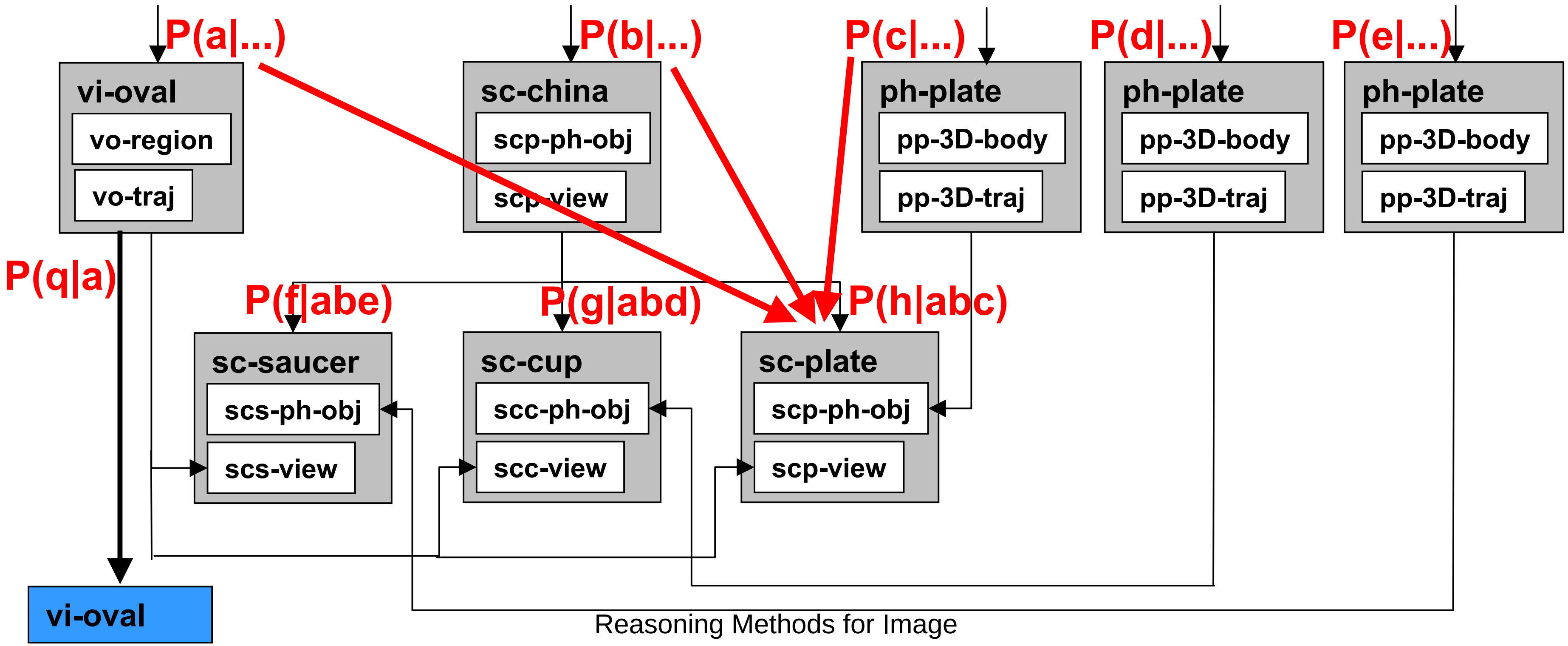
P(X4 = inattentive) = 0.6

P(X4 = attentive) = 0.4

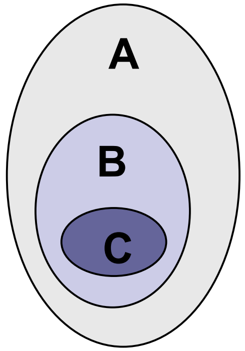
Bayes Net for hypotheses ranking

Probability of a successful interpretation step can be computed by

- **Bayes Net along is-a structure**
- **conditional probabilities for evidence classification**



Bayes Nets for taxonomical structures

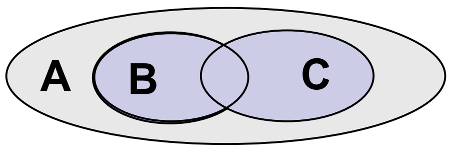


A, B, C, ... — **concepts**
a, b, c, ... — **respective instances**

<u>Basic idea</u>: **P(a b c) = P(c|b) P(b|a) P(a)**
if C á B á A

<u>In general</u>: **The is-a structure of a set of concepts equals the Bayes Net structure of the corresponding instances iff the specialisations of each concept are disjoint.**

If concepts are not is-a related but intersect, a Bayes Net along the is-a structure would not reflect the correlation between these concepts.



Some insights

- 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