Constraints

1

Basic Constraint Consistency Algorithm

Given:

- Variables V_1, V_2, ... , V_N, each with an associated domain dom(V_i)
- Constraint relations on various subsets of variables determine acceptable combinations of these variables.

Consistency Algorithm:

A Of each domain, prune values which are ruled out by any of the constraints. => domain consistency

B Of each domain, prune values for which there are no corresponding values in each of the constraint relations. Repeat until no more values can be pruned. => arc consistency

C If one domain is empty there is no solution. If each domain has a single value, the values are a unique solution.

D If some domains have more than one value, the values may or may not be a solution. By repeatedly splitting a domain and solving the reduced constraint problem, all solutions can be obtained. => global consistency

2

Constraint Evaluation for Stepwise Scene Interpretation

Incremental scene interpretation requires incremental constraint evaluation.

Case 1:
As a scene develops in time, which occurrences can be expected based on past occurrences and constraints relating to the future?

Case 2:
As objects of a scene are composed to tentative aggregates, what constraints are relevant for further parts?

Incremental constraint evaluation serves to reduce search space and remaining interpretation possibilities.

Example 1:
In a traffic scene, a ball running across the street raises the expectation of a child following the ball.

Example 2:
Given constraints for the distance of table-leg positions, the space of possible positions is reduced as table-legs are recognised incrementally.

3

Checking Temporal Constraints for Scene Interpretation

Variables:	Time variables of an aggregate model
Domains:	Time points covering the period of interest
Constraints:	1. Constraints imposed by aggregate model 2. Constraints arising from evidence

Example:

Aggregate model:

name:	traffic_ light_violation
parts:	red_traffic_light pass_traffic_light
constraints:	pass_traffic_light during red_traffic_light

Scene:

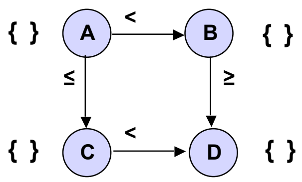
4

Constraint Net for Traffic-Light Violation

Nodes:
A = red_traffic_light.beg
B = red_traffic_light.end
C = pass_traffic_light.beg
D = pass_traffic_light.end

Arcs:
$A \leq C$, $B \geq D$ } pass_traffic_light during red_traffic_light
$A < B$, $C < D$ } begin of occurrence before end

Domains: dom(A) = dom(B) = dom(C) = dom(D) = { 0:0:0 ... 23:59:59 }



Step 1: Obtain consistency for initial constraint net
Step 2: Observe A=10:05:08, prune dom(A), obtain consistency
Step 3: Observe C=10:05:30, prune dom(C), obtain consistency
Step 4: Observe B=10:05:33, prune dom(B), obtain consistency
Step 5: Observe D=10:05:36, prune dom(D), obtain consistency, no solution is possible

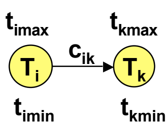
Animated slide!

5

Constraint Propagation in Convex Time-point Algebra

Variables:	time variables	T_i
Domain of a variable:	range of integers	$[t_{imin} .. t_{imax}]$
Constraints:	inequalities with offset	$T_i + c_{ik} \leq T_k$

Graphical representation:



- Domains may always be represented by min- and max-values ("convexity property").
- An increase of a min-value affects only time variables connected in edge direction.
- A decrease of a max-value affects only time variables connected against edge direction.
- In a cycle-free constraint net with N variables, any change of a domain can be propagated in at most N(N-1) steps.

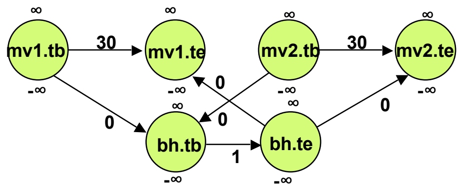
6

Constraint Propagation for Occurrence Recognition (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 Recognition (2)

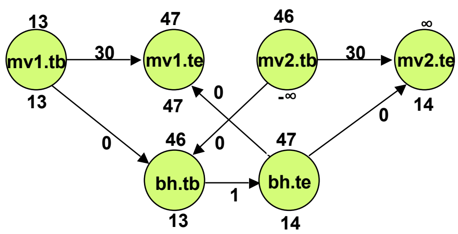
3. **Instantiate parts in occurrence model**

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



Animated slide!

Constraint Propagation for Occurrence Recognition (3)

4. **Consistency and completeness test**

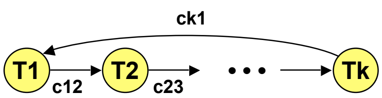
   A (partially) instantiated model is inconsistent, if for any node T one has: $t_{min} > t_{max}$

   => 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.

9

Convex Time-point Algebra Constraint Nets with Cycles



$t1 + c12 \le t2$ $t2 + c23 \le t3$ ... $tk + ck1 \le t1$

$\Rightarrow c12 + c23 + ... + ck1 \le 0$

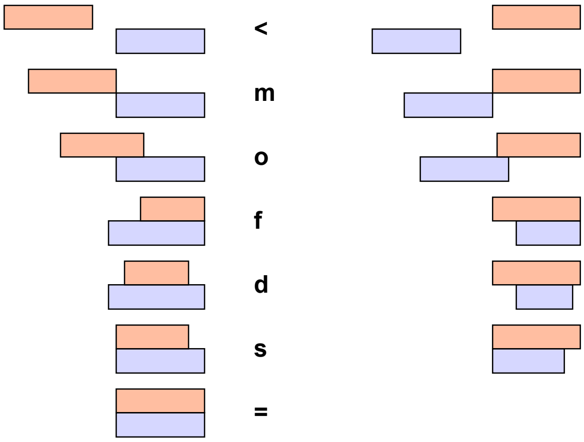
$\Sigma\, cik \le 0$:
The edge Tk - T1 can be omitted without affecting the propagation results.

$\Sigma\, cik > 0$:
Propagation will always lead to inconsistency, can be avoided alltogether.

➡ **Complexity of constraint propagation is not affected by cycles**

10

Basic Relations in Allen´s Interval Algebra



A before B	<	>
A meets B	m	mi
A overlaps B	o	oi
A finishes B	f	fi
A during B	d	di
A starts B	s	si
A equals B	=	

Composition Table for Interval Algebra (1)

For I_1 R_{12} I_2 and I_2 R_{23} I_3, the table specifies possible relations I_1 R_{13} I_3.
=> enables spatial reasoning

	<	m	o	fi	di	si	=
<	<	<	<	<	<	<	<
m	<	<	<	<	<	m	m
o	<	<	< m o	< m o	< m o fi di	o fi di	o
fi	<	m	o	fi	di	oi mi >	fi
di	< m o fi di	o fi di	o fi di	di	di	di	di
si	< m o fi di	o fi di	o fi di	di	di	si	si
=	<	m	o	fi	di	si	=
s	<	<	< m o	< m o	< m o fi di	s = si	s
d	<	<	< m o s d	< m o s d	full	d f oi mi >	d
f	<	m	o s d	f = fi	di si oi mi >	oi mi >	f
oi	< m o fi di	o fi di	o fi di si = s d f oi	di si oi	di si oi mi >	oi mi >	oi
mi	< m o fi di	s = si	d f oi	mi	>	>	mi
>	full	d f oi mi >	d f oi mi >	>	>	>	>

Composition Table for Interval Algebra (2)

	=	s	d	f	oi	mi	>
<	<	<	< m o s d	< m o s d	< m o s d	< m o s d	full
m	m	m	o s d	o s d	o s d	fi = f	di si oi mi >
o	o	o	o s d	o s d	o f d s = si di fi oi	di si oi	di si oi mi >
fi	fi	o	o s d	fi	di si oi	di si oi	di si oi mi pi
di	di	o fi di	o fi di si = s d f oi	di	di si oi	di si oi	di si oi mi pi
si	si	s = si	d f oi	di	oi	mi	>
=	=	s	d	f	oi	mi	>
s	s	s	d	p m o	d f oi	mi	>
d	d	d	d	< m o s d	d f oi mi >	>	>
f	f	d	d	f = fi	oi mi >	>	>
oi	oi	d f oi	d f oi	di si oi	oi mi >	>	>
mi	mi	d f oi	d f oi	mi	>	>	>
>	>	d f oi mi >	d f oi mi >	<	>	>	>

Note that only 27 disjunctive combinations out of 8192 possible combinations occur.

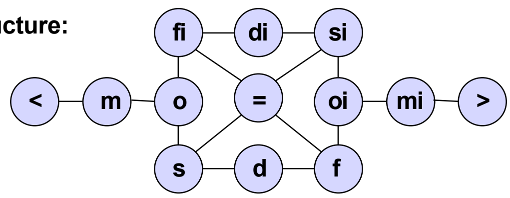
Conceptual Neighborhoods

C. Freksa: Conceptual Neighborhood and its role in temporal and spatial reasoning. In: M. Singh, L. Trave-Massuyes (eds.), Proc. IMACS Workshop on Decision Support Systems and Qualitative Reasoning, North-Holland, 1991, 181-187

In order to permit coarse reasoning, it is useful to identify "neighboring" interval relations.

Two relations between pairs of events are <u>conceptual neighbors</u> if they can be directly transformed into one another by continuous deformation (i.e. shortening or lengthening) of the events.

Conceptual neighborhood structure:



Note that entries of the composition table contain only conceptual neighbors.

Coarse Reasoning

Generate new "primitive" relations for coarse reasoning by combining conceptual neighbors out of the 13 original primitive relations.

{<} {m} {o fi di} {s = si} {d f oi} {o s d} {fi = f} {di si oi} {mi} {>}

The 10 coarse primitives generate a combination table for coarse inferences by disjunctive merging of rows and columns of the original table.

Example:
{o fi di} X {di si oi} => {m o fi di si = s d f oi mi}

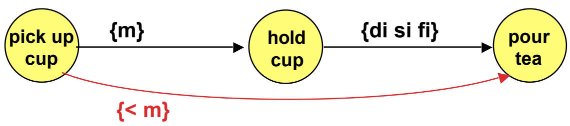
Reasoning within conceptual neighborhoods is monotonic:
If more information is added (i.e. disjunctive uncertainty removed), the refined result is contained in the coarse result.

Constraint Satisfaction with Intervals (1)

Constraint graph for reasoning with time intervals
Nodes: time intervals
Arcs: disjunctions of interval relations

Example: Pouring-tea-into-cup

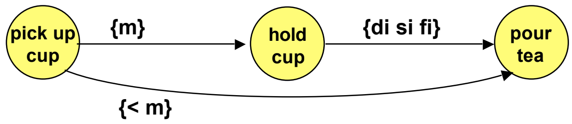
Assume that picking up a cup immediately precedes holding a cup, and pouring tea occurs during holding a cup.



Possible relations between pick-up-cup and pour-tea can be inferred using the composition table.

Animated slide!

Constraint Satisfaction with Intervals (2)

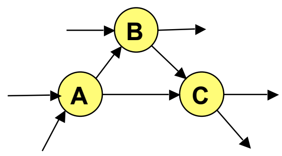


As observations of a specific scene become available, arc labels are pruned and remaining constraints can be checked for arc consistency.

Example 1: hold-cup overlaps pour-tea
=> inconsistent with model

Example 2: pick-up-cup meets pour-tea
=> di and fi relations between hold-cup and pour-tea are pruned

In general, interval constraint nets can be pruned by checking all triangles against the combination table until no more changes occur.



But: Arc consistency does not guarantee global consistency!

17

Spatial Constraints

In scene interpretation, spatial constraints restrict the relative position and orientation of parts of aggregates.

<u>Example</u>:

Relative positions of plate, saucer and table boundary as parts of a cover

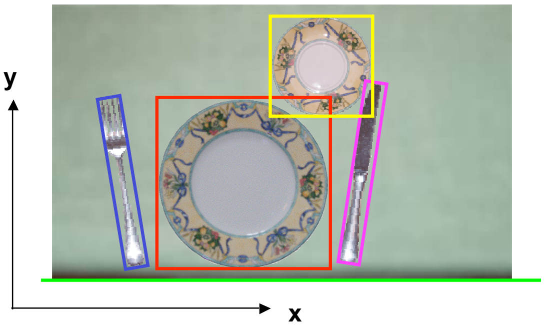
Several ways to represent 2D spatial constraints:

- Bounding box constraints
- Topological relations
- Various other qualitative spatial representations
- Grid region constraints
- Probability distributions

18

Bounding Box Constraints

A bounding box is an approximate 2D shape description



A bounding box is specified by
xmin, xmax, ymin, ymax
relativ to a reference coordinate system

- object-centric vs. global reference coordinate system
- position constraints in terms of relative distances between bounding-box boundaries
- orientation constraints in terms of angles between object axes

19

Extending Discrete Time-point Algebra to 2D-Space

Use linear inequalities independently in two spatial dimensions.
(Bounding boxes must be parallel to reference system.)

Example:

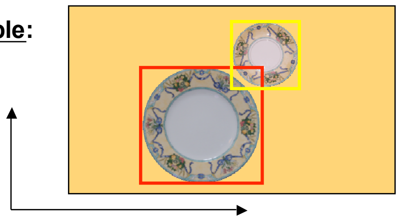
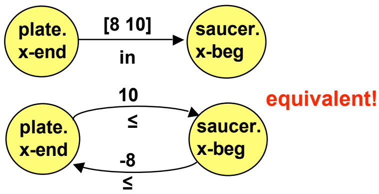


plate.x-end ≤ saucer.x-beg + 10
plate.x-end ≥ saucer.x-beg + 8
plate.y-end ≤ saucer.y-beg + 5
plate.y-end ≥ saucer.y-beg + 3
plate.x-beg ≥ table.x-beg
plate.x-end ≤ table.x-end
plate.y-beg ≤ table.y-beg + 5
plate.y-beg ≥ table.y-beg

Pairwise constraints can be combined to (quantitative) interval constraints:

plate.x-end in saucer.x-beg + [8 10]
plate.y-end in saucer.y-beg + [3 5]
plate.x-beg in table.x-beg + [0 inf]
plate.x-end in table.x-end + [-inf 0]
plate.y-beg in table.y-beg + [0 5]

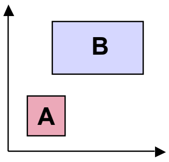


20

Extending Allen´s Interval Algebra to 2D-Space

Use Allen´s interval relations independently for two spatial dimensions.

Example:



horizontal relation: A o B
vertical relation: A < B

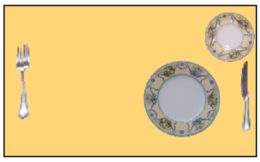
combination: A o|< B

Interval relations are often not restrictive enough to describe the variability of realistic spatial configurations.

Example: Cover configuration

plate o|m saucer
plate d|d table
plate >|s fork
plate <|s knife
saucer d|d table
fork d|d table
knife d|d table

Also covered by this description:



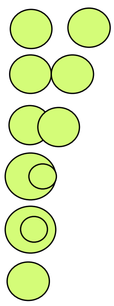
Animated slide!

21

Topological Relations in RCC8

RCC8: Region Connection Calculus with 8 topological binary relations

Elementary relations (disjunct):



- disconnected — dc
- externally connected — ec
- partial overlap — po
- tangential proper part — tpp, tppi
- non-tangential proper part — ntpp, ntppi
- equal — eq

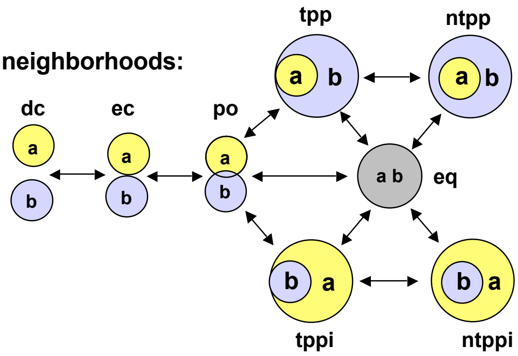
Composed relations:

- spatially_related
- connected
- overlapping
- inside

22

RCC8 Conceptual Neighborhoods

Conceptual neighborhoods:



Observations of two regions at two time points must be connected by transitions along a conceptual-neighborhood path.

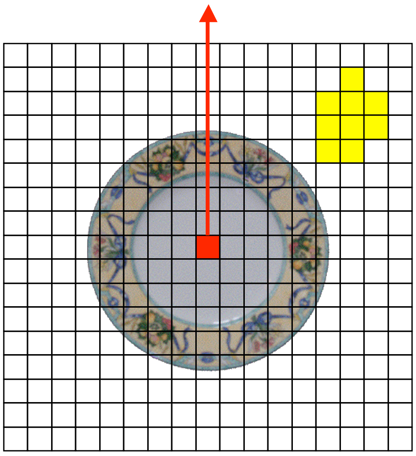
RCC8 Composition Table

Table entries denote possible relations R_{AC}, given R_{AB} and R_{BC}

○	DC	EC	PO	TPP	NTPP	TPPi	NTPPi	EQ
DC	DC,EC,PO TPP,NTPP TPPi,= NTPPi	DC,EC PO TPP NTPP	DC,EC PO TPP NTPP	DC,EC PO TPP NTPP	DC,EC PO TPP NTPP	DC	DC	DC
EC	DC,EC,PO TPPi NTPPi	DC,EC,PO =,TPP TPPi	DC,EC,PO TPP NTPP	EC,PO TPP NTPP	PO TPP NTPP	DC EC	DC	EC
PO	DC,EC,PO TPPi NTPPi	DC,EC,PO TPPi NTPPi	DC,EC,PO TPP,TPPi,= NTPP,NTPPi	PO TPP NTPP	PO TPP NTPP	DC,EC,PO TPPi NTPPi	DC,EC,PO TPPi NTPPi	PO
TPP	DC	DC EC	DC,EC PO,TPP NTPP	TPP NTPP	NTPP	DC,EC,PO =,TPP TPPi	DC,EC,PO TPPi NTPPi	TPP
NTPP	DC	DC	DC,EC PO TPP NTPP	NTPP	NTPP	DC,EC PO TPP NTPP	DC,EC,PO TPP,TPPi NTPP,= NTPPi	NTPP
TPPi	DC,EC,PO TPPi NTPPi	EC,PO TPPi NTPPi	PO TPPi NTPPi	PO,= TPP TPPi	PO TPP NTPP	TPPi NTPPi	NTPPi	TPPi
NTPPi	DC,EC,PO TPPi NTPPi	PO TPPi NTPPi	PO TPPi NTPPi	PO TPPi NTPPi	PO,TPP,= NTPP,TPPi NTPPi	NTPPi	NTPPi	NTPPi
EQ	DC	EC	PO	TPP	NTPP	TPPi	NTPPi	EQ

Spatial Relations as Grid-Point Sets

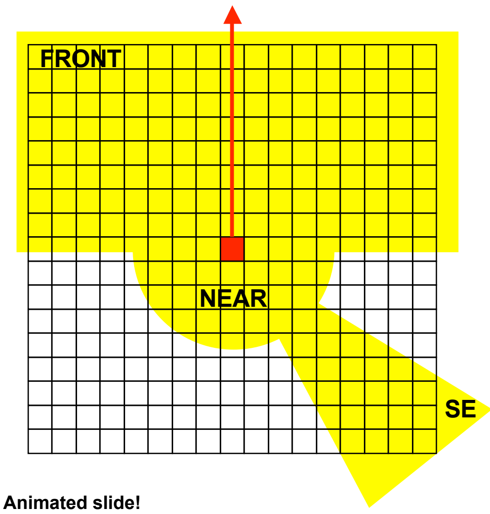
A grid region describes the possible locations (implicit OR) of a point r relativ to a reference point and a reference orientation of an object o.



Relative location is a relation
O x R
between an object o and some point r.

Example:
O = plate
r = center-of-gravity of saucer

Qualitative Spatial Relations as Grid-Point Sets



Grid-point sets constitute qualitative location concepts

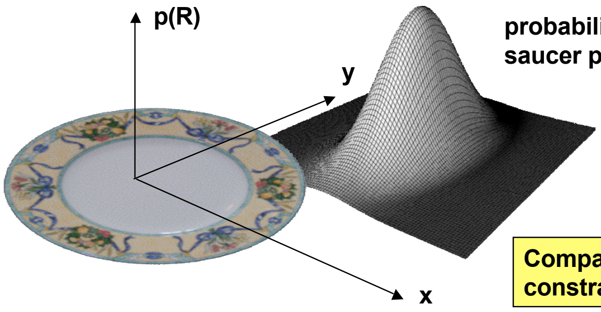
Constraint propagation is possible via set relationships

Example:
(SE plate saucer) ^
(FRONT plate saucer)
=> inconsistent

Animated slide!

Probability Distributions

Constraints on the coordinates (x, y) of a point relative to a reference coordinate system can be expressed in terms of a probability distribution (density).



probability density for saucer position

Compare with inequality constraints!

Probabilistic reasoning will be treated later.

27