# Advanced consistency methods
# Chapter 8

ICS-275

Winter 2016

# Relational consistency (Chapter 8)

- Relational arc-consistency
- Relational path-consistency
- Relational m-consistency
- Relational consistency for Boolean and linear constraints:
  - Unit-resolution is relational-arc-consistency
  - Pair-wise resolution is relational path-consistency

# Example

- Consider a constraint network over five integer domains, where the constraints take the form of linear equations and the domains are integers bounded by
  - $D_x$ in [-2,3]
  - $D_y$ in [-5,7]
  - $R_{xyz}$ : x + y = z
  - $R_{zlt}$ : z + t = l
  - From $D_x$ and $R_{xyz}$ infer z-y in [-2,3] from this and $D_y$ we can infer z in [-7,10]

# Relational arc-consistency

- Let *R* be a constraint network , *X*= $\{x_1,\ldots, x_n\}$, $D_1,\ldots, D_n$, $R_S$ a relation.
- $R_S$ in *R* is *relational-arc-consistent* relative to x in S, iff any *consistent* instantiation of the variables in *S*- *{x}* has an extension to a value in $D_x$ that satisfies $R_S$. Namely,

$\rho$(A) is the set
Of all solutions to constraints
Over A

$$\rho(S - x) \subseteq \pi_{S-x} R_S \bowtie D_x$$

# Enforcing relational arc-consistency

- If arc-consistency is not satisfied add:

$$R_{S-x} \leftarrow R_{S-x} \cap \pi_{S-x} R_S \otimes D_S$$

# Example

- $R_{xyz} = \{(a,a,a),(a,b,c),(b,b,c)\}$.
- This relation is not relational arc-consistent, but if we add the projection:

  $R_{xy} = \{(a,a),(a,b),(b,b)\}$, then $R_{xyz}$ will be relational arc-consistent relative to $\{z\}$.
- To make this network relational-arc-consistent, we would have to add all the projections of $R_{xyz}$ with respect to all subsets of its variables.

# Relational path-cosistency

- Let R_S and R_T be two constraints in a network.
- $R_S$ and $R_T$ are relational-path-consistent relative to a variable x in S U T iff any consistent instantiation of variables in S T - {x} has an extension to in the domain $D_x$, s.t. $R_S$ and $R_T$ simultaneously;
- $$\rho(A) \subseteq \pi_A R_S \otimes R_T \otimes D_x$$ $$A = S \cup T - x$$
- A pair of relations $R_S$ and $R_T$ is relational-path-consistent iff it is relational-path-consistent relative to every variable in S ∩ T. A network is relational-path-consistent iff every pair of its relations is relational-path-consistent.

$D_x$ in [-2,3]

D_y in [-5,7]

Example:

$R\_\{xyz\}$:= x + y = z

R_{ztl}:= z + t = l

- We can assign to *x, y, l* and *t* values that are consistent relative to the relational-arc-consistent network generated in earlier. For example, the assignment

- *(x=2, y= -5, t=3, l=15)* is consistent, since only domain restrictions are applicable, but no value of z that satisfies *x+y = z* and *z+t = l*.
- To make the two constraints relational path-consistent relative to *z* add : *x+y+t = l*.

# Enforcing relational arc, path and m-consistency

- If arc-consistency is not satisfied add:

r.a.c

$$R_{S-x} \leftarrow R_{S-x} \cap \pi_{S-x} R_S \otimes D_S$$

r.p.c

$$\rho(A) \subseteq \pi_A R_S \otimes R_T \otimes D_x$$

$$A = S \cup T - x$$

r.m.c

[illegible] $R_{S_i} \otimes D_x$

[illegible] $m$ [illegible]

# Extended composition

- The extended composition of relation $R_{S_1} \ldots R_{S_m}$ relative to A is defined by

$$EC_A(R_{1,} \ldots, R_m) = \pi_A(R_1 \otimes R_2 \otimes, \ldots, \otimes R_m)$$

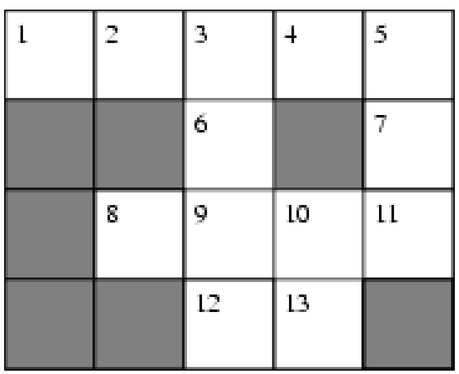
- If the projection operation is restricted to subsets of size i, it is called extended (*i,m*)-composition.
- Special cases: domain propagation and relational arc-consistency

$$D_x \leftarrow D_x \cap \pi_x(R_S \otimes D_S)$$

$$R_{S-x} \leftarrow R_{S-x} \cap \pi_{S-x}(R_S \otimes D_S)$$

# Example: crossword puzzle, DRC_2

$R_{1,2,3,4,5} = \{(H,O,S,E,S),(L,A,S,E,R),(S,H,E,E,T),$
$(S,N,A,I,L),(S,T,E,E,R)\}$
$R_{3,6,9,12} = \{(H,I,K,E),(A,R,O,N),(K,E,E,T),(E,A,R,N),$
$(S,A,M,E)\}$
$R_{5,7,11} = \{(R,U,N),(S,U,N),(L,E,T),(Y,E,S),(E,A,T),(T,E,N)\}$
$R_{8,9,10,11} = R_{3,6,9,12}$
$R_{10,13} = \{(N,O),(B,E),(U,S),(I,T)\}$
$R_{12,13} = R_{10,13}$




$bucket(x_1)$

$bucket(x_2)$

$bucket(x_3)$

$bucket(x_4)$

$bucket(x_5)$

$bucket(x_6)$

$bucket(x_7)$

$bucket(x_8)$

$bucket(x_9)$

$bucket(x_{10})$

$bucket(x_{11})$

$bucket(x_{12})$

$bucket(x_{13})$

# Example: crossword puzzle, Directional-relational-2

$$R_{1,2,3,4,5} = \{(H,O,S,E,S),(L,A,S,E,R),(S,H,E,E,T),(S,N,A,I,L),(S,T,E,E,R)\}$$

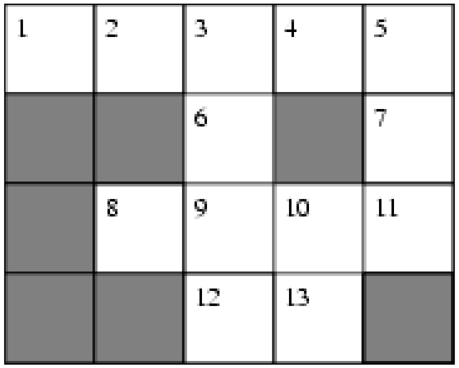
$$R_{3,6,9,12} = \{(H,I,K,E),(A,R,O,N),(K,E,E,T),(E,A,R,N),(S,A,M,E)\}$$

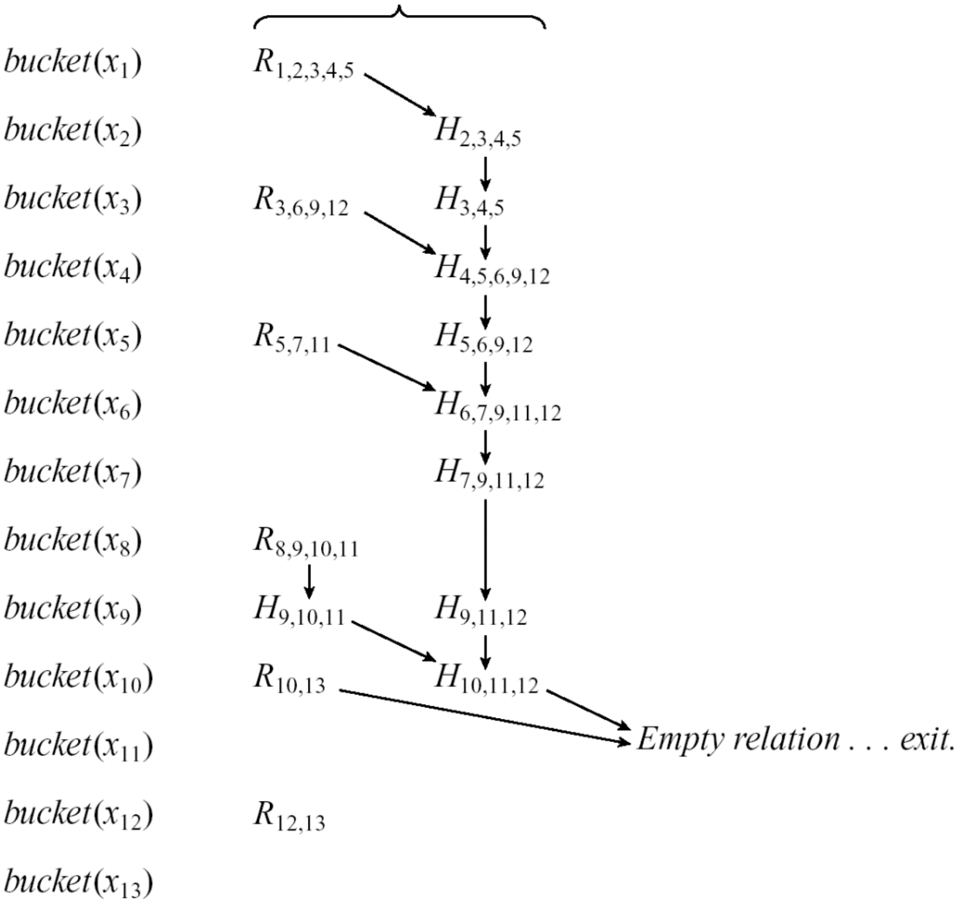
$$R_{5,7,11} = \{(R,U,N),(S,U,N),(L,E,T),(Y,E,S),(E,A,T),(T,E,N)\}$$

$$R_{8,9,10,11} = R_{3,6,9,12}$$

$$R_{10,13} = \{(N,O),(B,E),(U,S),(I,T)\}$$

$$R_{12,13} = R_{10,13}$$



# Complexity

- Theorem: DRC_2 is exponential in the induced-width.
- (because sizes of the recorded relations are exp in w).

- Crossword puzzles can be made directional backtrack-free by DRC_2

# Domain tightness

- **Theorem:** a strong relational 2-consistent constraint network over bi-valued domains  is globally consistent.

- **Theorem**: A strong relational k-consistent constraint network with at most k values is globally consistent.

# Inference for Boolean theories

- Resolution is identical to *extended 2 decomposition*
- check: $\{(f \vee x \vee y \vee \sim z), (x \vee y \vee f)\}$
- Boolean theories have domain size 2
- Therefore $DRC_2$ makes a cnf globally consistent.
- $DRC_2$ expressed on cnfs is directional resolution

# Directional resolution

DIRECTIONAL-RESOLUTION

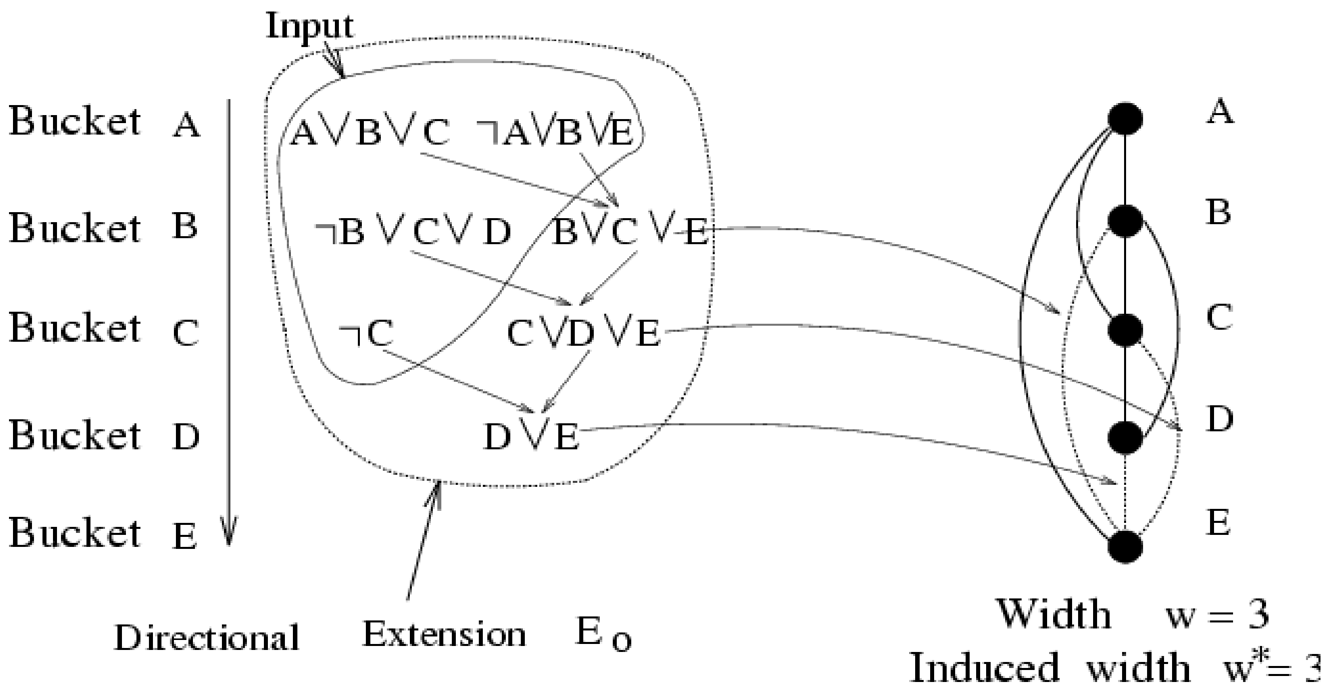
**Input:** A *CNF* theory $\varphi$, an ordering $d = Q_1, \ldots, Q_n$ of its variables.

**Output:** A decision of whether $\varphi$ is satisfiable. If it is, a theory $E_d(\varphi)$, equivalent to $\varphi$, else an empty directional extension.

1. **Initialize:** generate an ordered partition of clauses into buckets. $bucket_1, \ldots, bucket_n$, where $bucket_i$ contains all clauses whose highest literal is $Q_i$.
2. **for** $i \leftarrow n$ **downto** 1 process $bucket_i$:
3. **if** there is a unit clause **then** (the instantiation step)
   apply unit-resolution in $bucket_i$ and place the resolvents in their right buckets.
   **if** the empty clause was generated, theory is not satisfiable.
4. **else** resolve each pair $\{(\alpha \vee Q_i), (\beta \vee \neg Q_i)\} \subseteq bucket_i$.
   **if** $\gamma = \alpha \vee \beta$ is empty, return $E_d(\varphi) = \{\}$, theory is not satisfiable
   **else** determine the index of $\gamma$ and add it to the appropriate bucket.
5. **return** $E_d(\varphi) \leftarrow \bigcup_i bucket_i$

Figure 4.20: Directional-resolution

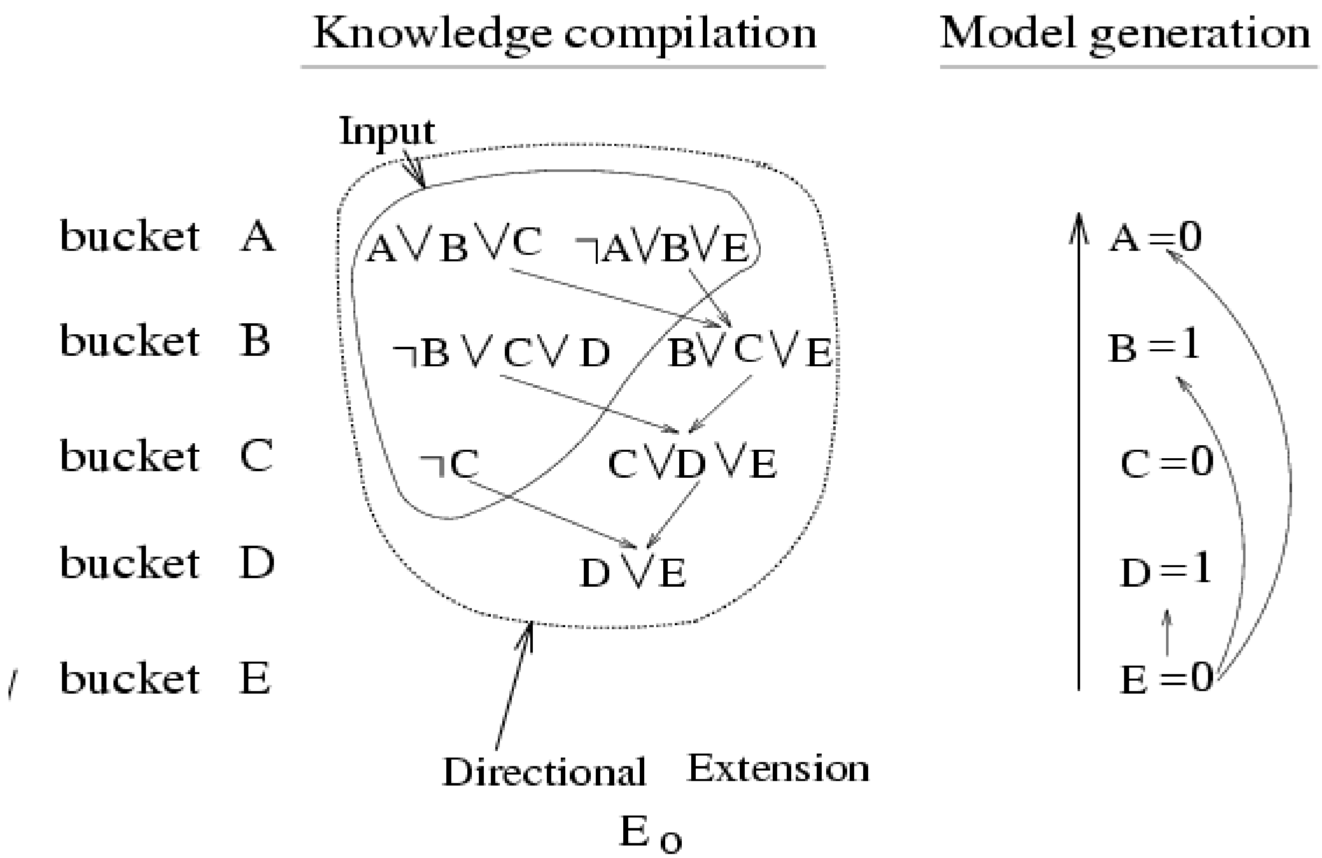
# DR resolution = adaptive-consistency=directional relational path-consistency




$|bucket_i| = O(\exp(w^*))$

DR time and space : $O(n \exp(w^*))$

# Directional Resolution ⇔ Adaptive Consistency



# Resolution – An Example

$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_3) \wedge (x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \quad \vdash$$

$$(\neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_3) \wedge (x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \quad \vdash$$

# Row convexity

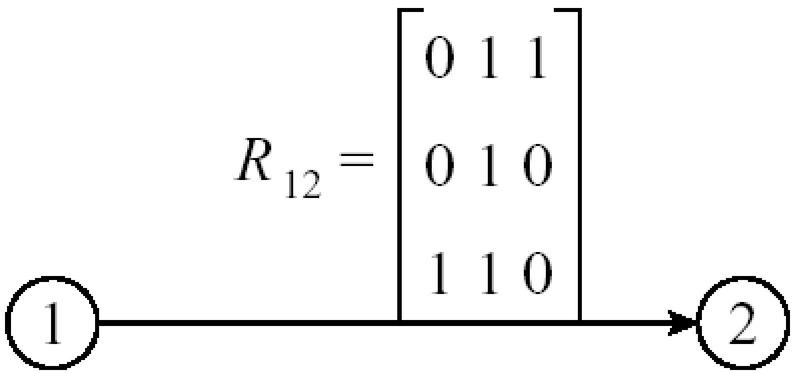
- **Functional constraints**: A binary relation R_{ij} expressed as a (0,1)-matrix is functional iff there is at most a single "1" in each row and in each column.

- **Monotone constraints**: Given ordered domain, a binary relation R_{ij} is monotone if (a,b) in R_{ij} and if c >= a, then (c,b) in R_{ij}, and if (a,b) in R_{ij} and c <= b, then (a,c) in R_{ij}.

- **Row convex constraints**: A binary relation R_{ij} represented as a (0,1)-matrix is row convex if in each row (column) all of the ones are consecutive}

# Example of row convexity

$$R_{12} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$



# Theorem:

- Let *R* be a path-consistent binary constraint network. If there is an ordering of the domains $D_1, \ldots, D_n$ of *R* such that the relations of all constraints are row convex, the network is globally consistent and is therefore minimal.

# Linear inequalities

- Consider r-ary constraints over a subset of variables $,x_1 .. \, x_r$ of the form
- $a_r x_r$ + … + $a_r x_r$ <= c, $a_i$ are rational constants. The r-ary inequalities define corresponding r-ary relations that are *row convex*.
- Since r-ary linear inequalities that are closed under relational path-consistency are row-convex, relative to any set of integer domains (using the natural ordering).

- **Proposition:** A set of linear inequalities that is closed under $RC_2$ is globally consistent.

# Linear inequalities

- Gausian elimination with domain constraint is relational-arc-consistency
- Gausian elimination of 2 inequalities  is relational path-consistency
- **Theorem**: directional relational path-consistency is complete for CNFs and for linear inequalities

# Linear inequalities: Fourier elimination

DIRECTIONAL-LINEAR-ELIMINATION $(\varphi, d)$

**Input:** A *set of linear inequalities* $\varphi$, an ordering $d = x_1, \ldots, x_n$.

**Output**A decision of whether $\varphi$ is satisfiable. If it is, a backtrack-free theory $E_d(\varphi)$.

1. **Initialize:** Partition inequalities into ordered buckets.
2. **for** $i \leftarrow n$ **downto** 1 **do**
3. **if** $x_i$ has one value in its domain **then**

. substitute the value into each inequality in the bucket and put the resulting inequality in the right bucket.

4. **else,for each** pair $\{\alpha, \beta\} \subseteq bucket_i$, compute $\gamma = elim_i(\alpha, \beta)$ **if** $\gamma$ has no solutions, return $E_d(\varphi) = \{\}$, "inconsistency" **else** add $\gamma$ to the appropriate lower bucket.
5. **return** $E_d(\varphi) \leftarrow \bigcup_i bucket_i$

Figure 4.22: Fourier Elimination; DLE

## Directional linear elimination, DLE : generates a backtrack-free representation

**Theorem 4.8.3** *Given a set of linear inequalities $\varphi$, algorithm DLE (Fourier elimination) decides the consistency of $\varphi$ over the Rationals and the Reals, and it generates an equivalent backtrack-free representation.* □

# Example

$$
\begin{aligned}
bucket_4 &: \quad 5x_4 + 3x_2 - x_1 \leq 5,\ x_4 + x_1 \leq 2,\ \ -x_4 \leq 0,\\
bucket_3 &: \quad x_3 \leq 5,\ x_1 + x_2 - x_3 \leq -10\\
bucket_2 &: \quad x_1 + 2x_2 \leq 0.\\
bucket_1 &:
\end{aligned}
$$

Figure 4.23: initial buckets

$$
\begin{aligned}
bucket_4 &: \quad 5x_4 + 3x_2 - x_1 \leq 5,\ x_4 + x_1 \leq 2,\ \ -x_4 \leq 0,\\
bucket_3 &: \quad x_3 \leq 5,\ x_1 + x_2 - x_3 \leq -10\\
bucket_2 &: \quad x_1 + 2x_2 \leq 0 \ || \ 3x_2 - x_1 \leq 5, x_1 + x_2 \leq -5\\
bucket_1 &: \quad || \ x_1 \leq 2.
\end{aligned}
$$

Figure 4.24: final buckets