Advanced consistency methods

Chapter 8
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 $[-5,7]$
  - $R_{xyz} := x + y = z$
  - $R_{ztl} := z + t = l$
  - 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,

$$\rho(S - x) \subseteq \pi_{S - x} R_S \otimes D_S$$
Enforcing relational arc-consistency

- If arc-consistency is not satisfied add:

\[ R_{S-x} \leftarrow R_{S-x} \cap \pi_{S-x} R_S \bigotimes D_S \]
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 become 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-consistency

- 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 \( S \cup T \) iff any consistent instantiation of the variables in \( S \cup T - \{x\} \) has an extension to a value in the domain \( D_x \), that satisfies \( R_S \) and \( R_T \) simultaneously;

\[
\rho(A) \subseteq \pi_A R_S \otimes R_T,
\]

\[
A = S \cup T - x
\]

- A pair of relations \( R_S \) and \( R_T \) is relational-path-consistent iff it is relational-path-consistent relative to every variable in \( S \cup T \). A network is relational-path-consistent iff every pair of its relations is relational-path-consistent.
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 there is no value of $z$ that simultaneously satisfies $x + y = z$ and $z + t = l$. To make the two constraints relational path-consistent relative to $z$ we should deduce the constraint $x + y + t = l$ and add it to the network.
Relational m-consistency

- Let $R_{S_1}, \ldots, R_{S_m}$ be $m$ distinct constraints.
- $R_{S_1}, \ldots, R_{S_m}$ are relational-m-consistent relative to $x$ in $U_{i=1}^m S_i$ iff any consistent instantiation of the variables in $A = U_{i=1}^m S_i - \{x\}$ has an extension to $x$ that satisfies $R_{S_1}, \ldots, R_{S_m}$ simultaneously;

$$\rho(A) \subseteq \pi_A \otimes_{i=1,m} R_{S_i} \otimes D_x$$

$$A = S_1 \cup \ldots S_m - x$$

- A set of relations $\{R_{S_1}, \ldots, R_{S_m}\}$ is relational-m-consistent iff it is relational-m-consistent relative to every variable in their scopes. A network is relational-m-consistent iff every set of $m$ relations is relational-m-consistent. A network is strongly relational-m-consistent if it is relational-i-consistent for every $i \leq m$. 
A set of relations $R_{S_1}, \ldots, R_{S_m}$ is relationally $(i,m)$-consistent iff for every subset of variables $A$ of size $i$, $A \in \bigcup_{j=1}^{m} S_j$, any consistent assignment to $A$ can be extended to an assignment to $\bigcup_{i=1}^{m} S_i - A$ that satisfies all $m$ constraints simultaneously.

A network is relationally $(i,m)$-consistent iff every set of $m$ relations is relationally $(i,m)$-consistent. A network is strong relational $(i,m)$-consistent iff it is relational $(j,m)$-consistent for every $j \leq i$. 
The extended composition of relation $R_{S_1}, \ldots, R_{S_m}$ relative to $A$ in $\bigcup_{i=1}^m S_i$, $EC_A (R_{S_1}, \ldots, R_{S_m})$, is defined by

$$EC_A (R_{S_1}, \ldots, R_{S_m}) = \pi_A (\Join_{i=1}^m R_{S_i})$$

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

- $D_x \leftarrow \pi_x (R_S \Join D_x)$
- $R_{S-x} \leftarrow \pi_{S-x} (R_S \Join D_x)$
Given an ordering $d = (x_1, \ldots, x_n)$, $R$ is $m$-directionally relationally consistent iff for every subset of constraints $R_{\{S_1\}, \ldots, R_{\{S_m\}}}$ where the latest variable is $x_l$, and for every $A$ in $\{x_1, \ldots, x_{l-1}\}$, every consistent assignment to $A$ can be extended to $x_l$ while simultaneously satisfying all these constraints.
Summary: directional i-consistency

- E: E ≠ D, E ≠ C, E ≠ B
- D: D ≠ C, D ≠ A
- C: C ≠ B
- B: A ≠ B
- A:

Adaptive: $R_{DCB}$

d-path: $R_{DC}, R_{DB}$

D-arc: $R_D, R_C, R_{CD}$
**Example: crossword puzzle**

\[
\begin{align*}
R_{1,2,3,4,5} &= \{(H, O, S, E, S), (L, A, S, E, R), (S, H, E, E, T), \\
&\quad (S, N, A, I, L), (S, T, E, E, R)\} \\
R_{3,6,9,12} &= \{(H, I, K, E), (A, R, O, N), (K, E, E, T), (E, A, R, N), \\
&\quad (S, A, M, E)\} \\
R_{5,7,11} &= \{(R, U, N), (S, U, N), (L, E, T), (Y, E, S), (E, A, T), (T, E, N)\} \\
R_{8,9,10,11} &= R_{3,6,9,12} \\
R_{10,13} &= \{(N, O), (B, E), (U, S), (I, T)\} \\
R_{12,13} &= R_{10,13}
\end{align*}
\]
Example: crossword puzzle, DRC_2

bucket(x₁) \rightarrow R_{1,2,3,4,5} \rightarrow H_{2,3,4,5}
bucket(x₂)
bucket(x₃) \rightarrow R_{3,6,9,12} \rightarrow H_{3,4,5}
bucket(x₄)
bucket(x₅) \rightarrow R_{5,7,11} \rightarrow H_{4,5,6,9,12}
bucket(x₆)
bucket(x₇)
bucket(x₈) \rightarrow R_{8,9,10,11} \rightarrow H_{5,6,9,12}
bucket(x₉)
bucket(x₁₀) \rightarrow R_{9,10,11} \rightarrow H_{6,7,9,11,12}
bucket(x₁₁)
bucket(x₁₂) \rightarrow R_{10,13} \rightarrow H_{7,9,11,12}
bucket(x₁₃)
H_{10,11,12} \rightarrow \text{Empty relation . . . exit.}
Complexity

- Even DRC_2 is exponential in the induced-width.

- Crossword puzzles can be made directional backtrack-free by DRC_2
Domain and constraint tightness

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

- **m-tightness**: $R_S$ of arity $r$ is $m$-tight if, for any variable $x_i \in S$ and any instantiation of the remaining $r-1$ variables in $S - x_i$, either there are at most $m$ extensions of to $x_i$ that satisfy $R_S$, or there are exactly $|D_i|$ such extensions.

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

- Example: $D_i = \{a,b,c\}$,
- $R_{\{x1,x2,x3\}} = \{(aaa),(aac),(abc),(acb),(bac),(bbb),(bca),(cab),(cba),(ccc)\}$
Inference for Boolean theories

- Resolution is identical to Extended 2 decomposition
- Boolean theories are 2-tight
- 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** $i \leftarrow n$ **down to** 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 $\{(\alpha \lor Q_i), (\beta \lor \neg Q_i)\} \subseteq bucket_i$.
   
   **if** $\gamma = \alpha \lor \beta$ 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

\[ \vert \text{bucket}_i \vert = O(\exp(w^*)) \]

DR time and space: \( O(n \exp(w^*)) \)
Directional Resolution ⇔ Adaptive Consistency

Knowledge compilation

bucket A: \( A \lor B \lor C \lor \neg A \lor B \lor E \)

bucket B: \( \neg B \lor C \lor D \lor B \lor C \lor E \)

bucket C: \( \neg C \lor C \lor D \lor E \)

bucket D: \( D \lor E \)

bucket E:

Directional Extension \( E_0 \)

Model generation

\( A = 0 \)
\( B = 1 \)
\( C = 0 \)
\( D = 1 \)
\( E = 0 \)
History

- 1960 – resolution-based Davis-Putnam algorithm

- 1962 – resolution step replaced by conditioning (Davis, Logemann and Loveland, 1962) to avoid memory explosion, resulting into a backtracking search algorithm known as Davis-Putnam (DP), or DPLL procedure.

- The dependency on induced width was not known in 1960.

- 1994 – Directional Resolution (DR), a rediscovery of the original Davis-Putnam, identification of tractable classes (Dechter and Rish, 1994).
Complexity of DR

Theorem 4.7.6 (complexity of DR) Given a theory $\varphi$ and an ordering of its variables $o$, the time complexity of algorithm DR along $o$ is $O(n \cdot 9^{w_o^*})$, and $E_o(\varphi)$ contains at most $n \cdot 3^{w_o^*+1}$ clauses, where $w_o^*$ is the induced width of $\varphi$'s interaction graph along $o$. □

● 2-cnfs and Horn theories

Theorem 4.7.7 Given a 2-cnf theory $\varphi$, its directional extension $E_o(\varphi)$ along any ordering $o$ is of size $O(n \cdot w_o^{*2})$, and can be generated in $O(n \cdot w_o^{*2})$ time.

Theorem 4.7.8 The consistency of Horn theories can be determined by unit propagation. If the empty clause is not generated, the theory is satisfiable. □
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 \(c \geq a\), then \((c,b) \in R_{ij}\), and if \((a,b) \in R_{ij}\) and \(c \leq b\), 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

\[
R_{12} = \begin{bmatrix}
0 & 1 & 1 \\
0 & 1 & 0 \\
1 & 1 & 0 \\
\end{bmatrix}
\]
Lemma: Let $F$ be a finite collection of $(0,1)$-row vectors that are row convex and of equal length. If every pair of rows have a non-zero intersection, then all of the rows have a non-zero entry in common.
Theorem: Let $R$ be a path consistent binary constraint network. If there exists 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.
Example:

- Cube 3-dimensional recognition
- Bi-valued binary constraints
- 2-colorability
Linear constraints

- inequalities of the form
  - $a x_i - b x_j = c$,
  - $a x_i - b x_j < c$,
  - $a x_i - b x_j \leq c$,
  - $a, b, \text{ and } c$ are integer constants.

- However, it can be shown that each element in the closure under composition, intersection, and transposition of the resulting set of $(0,1)$-matrices is row convex, provided that when an element is removed from a domain by arc consistency, the associated $(0,1)$-matrices are "condensed."

- Hence, we can guarantee that the result of path consistency will be row-convex and therefore minimal, and that the network will be globally consistent for any binary linear equation over the integers.
Theorem: [Booth and Lueker, 1976]: An m x n (0,1)-matrix specified by its f nonzero entries can be tested for whether permutation of the columns exists such that the matrix is row convex in O(m + n + f) steps.
Consider r-ary constraints over a subset of variables $x_1, \ldots, x_r$ of the form $a_1 x_1 + \ldots + a_r x_r \leq c$, $a_i$ 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

- Gaussian elimination with domain constraint is relational-arc-consistency
- Gaussian elimination of 2 inequalities is relational path-consistency

**Theorem:** directional path-consistency is complete for CNFs and for linear inequalities
**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. \hspace{1em} if $x_i$ has one value in its domain **then**
   \hspace{2em} substitute the value into each inequality in the bucket and put the resulting inequality in the right bucket.
4. \hspace{1em} else **for each** pair $\{\alpha, \beta\} \subseteq \text{bucket}_i$, compute $\gamma = \text{elim}_i(\alpha, \beta)$
   \hspace{2em} if $\gamma$ has no solutions, return $E_d(\varphi) = \emptyset$, “inconsistency”
   \hspace{2em} else add $\gamma$ to the appropriate lower bucket.
5. **return** $E_d(\varphi) \leftarrow \bigcup_i \text{bucket}_i$

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. $\square$
Example

\[
\begin{align*}
\text{bucket}_4 &: \quad 5x_4 + 3x_2 - x_1 \leq 5, \quad x_4 + x_1 \leq 2, \quad -x_4 \leq 0, \\
\text{bucket}_3 &: \quad x_3 \leq 5, \quad x_1 + x_2 - x_3 \leq -10 \\
\text{bucket}_2 &: \quad x_1 + 2x_2 \leq 0. \\
\text{bucket}_1 &: \\
\end{align*}
\]

Figure 4.23: initial buckets

\[
\begin{align*}
\text{bucket}_4 &: \quad 5x_4 + 3x_2 - x_1 \leq 5, \quad x_4 + x_1 \leq 2, \quad -x_4 \leq 0, \\
\text{bucket}_3 &: \quad x_3 \leq 5, \quad x_1 + x_2 - x_3 \leq -10 \\
\text{bucket}_2 &: \quad x_1 + 2x_2 \leq 0 \quad || \quad 3x_2 - x_1 \leq 5, \quad x_1 + x_2 \leq -5 \\
\text{bucket}_1 &: \quad || \quad x_1 \leq 2. \\
\end{align*}
\]

Figure 4.24: final buckets