Principles of Constraint Programming

Krzysztof R. Apt

Chapter 2
Constraint Satisfaction Problems: Examples
Objectives

• Define formally Constraint Satisfaction Problems (CSP),
• **Modeling**: representation of a problem as a CSP.
• Clarify various aspects of modeling:
  – in general several natural representations exist,
  – some representations straightforward, some non-trivial,
  – some representations rely on a “background” theory.
• Show the generality of the notion of a CSP.
Constraint Satisfaction Problems (CSP)

Given:

- **Variables** $Y := y_1, \ldots, y_k$,
- **Domains** $D_1, \ldots, D_k$,

**Constraint** $C$ on $Y$: subset of $D_1 \times \ldots \times D_k$.

Given:

- **Variables** $x_1, \ldots, x_n$,
- **Domains** $D_1, \ldots, D_n$,

**Constraint Satisfaction Problem (CSP):**

$$\{ C ; x_1 \in D_1, \ldots, x_n \in D_n \}$$

$C$ – constraints, each on a subsequence of $x_1, \ldots, x_n$.

$(d_1, \ldots, d_n) \in D_1 \times \ldots \times D_n$ is a **solution** to

$$\{ C ; x_1 \in D_1, \ldots, x_n \in D_n \}$$

if for every constraint $C' \in C$ on $x_{i_1}, \ldots, x_{i_m}$

$$(d_{i_1}, \ldots, d_{i_m}) \in C.$$
Example: \(\text{SEND} + \text{MORE} = \text{MONEY}\)

Replace each letter by a different digit so that

\[
\begin{align*}
\text{SEND} & \\
+ \text{MORE} & \\
\hline
\text{MONEY} & 
\end{align*}
\]

is a correct sum.

Unique solution:

\[
\begin{align*}
9567 & \\
+ 1085 & \\
\hline
10652 & 
\end{align*}
\]

**Variables:** \(S, E, N, D, M, O, R, Y\),

**Domains:**
- \([1..9]\) for \(S, M\),
- \([0..9]\) for \(E, N, D, O, R, Y\).
Alternatives for Equality Constraints

1. 1 equality constraint.

\[
1000 \cdot S + 100 \cdot E + 10 \cdot N + D + 1000 \cdot M + 100 \cdot O + 10 \cdot R + E = 10000 \cdot M + 1000 \cdot O + 100 \cdot N + 10 \cdot E + Y
\]

2. 5 equality constraints.

Use “carry” variables \( C_1, \ldots, C_4 \in [0..1] \):

\[
\begin{align*}
D + E &= 10 \cdot C_1 + Y, \\
C_1 + N + R &= 10 \cdot C_2 + E, \\
C_2 + E + O &= 10 \cdot C_3 + N, \\
C_3 + S + M &= 10 \cdot C_4 + O, \\
C_4 &= M.
\end{align*}
\]
Alternatives for Disequality Constraints

1. 28 disequality constraints.
   
   \[ x \neq y \text{ for } x, y \in \{S, E, N, D, M, O, R, Y\}, \]
   
   \[ x < y. \]

2. A single constraint for disequalities.

   For variables \(x_1, \ldots, x_n\) with domains \(D_1, \ldots, D_n\):
   
   \[
   \text{all} \_ \text{different}(x_1, \ldots, x_n) := \{(d_1, \ldots, d_n) \mid d_i \neq d_j \text{ for } i \neq j\}.
   \]

   Use
   
   \[
   \text{all} \_ \text{different}(S, E, N, D, M, O, R, Y).
   \]


   For \(x, y \in \{S, E, N, D, MO, R, Y\}\) transform \(x \neq y\) to

   \[
   x - y \leq 10 - 11z_{x,y},
   \]

   \[
   y - x \leq 11z_{x,y} - 1,
   \]

   where \(z_{x,y} \in [0..1]\).

   **Disadvantage:** 28 new variables.
Problem Place $n$ queens on the $n \cdot n$ chess board so that they do not attack each other.

Variables: $x_1, \ldots, x_n$,

Domains: $[1..n]$,

Constraints:

For $i \in [1..n - 1]$ and $j \in [i + 1..n]$:

- $x_i \neq x_j$ (rows),
- $x_i - x_j \neq i - j$
  (South-West – North-East diagonals),
- $x_i - x_j \neq j - i$
  (North-West – South-East diagonals).
A small street has five differently colored houses on it. Five men of different nationalities live in these five houses. Each man has a different profession, each man likes a different drink, and each has a different pet animal.
The Englishman lives in the red house.
The Spaniard has a dog.
The Japanese is a painter.
The Italian drinks tea.
The Norwegian lives in the first house on the left.
The owner of the green house drinks coffee.
The green house is on the right of the white house.
The sculptor breeds snails.
The diplomat lives in the yellow house.
They drink milk in the middle house.
The Norwegian lives next door to the blue house.
The violinist drinks fruit juice.
The fox is in the house next to the doctor’s.
The horse is in the house next to the diplomat’s.

Who has the zebra and who drinks water?
Variables:
- nationality: english, spaniard, japanese, italian, norwegian,
- pet: dog, snails, fox, horse, zebra,
- profession: painter, sculptor, diplomat, violinist, doctor,
- drink: tea, coffee, milk, juice, water,
- colour: red, green, white, yellow, blue.

Domains: [1...5].

Constraints:

\texttt{all\_different(red, green, white, yellow, blue)},
\texttt{all\_different(english, spaniard, japanese, italian, norwegian)},
\texttt{all\_different(dog, snails, fox, horse, zebra)},
\texttt{all\_different(painter, sculptor, diplomat, violinist, doctor)},
\texttt{all\_different(tea, coffee, milk, juice, water)}. 
• The Englishman lives in the red house:
  \( \text{english} = \text{red}, \)
• The Spaniard has a dog:
  \( \text{spaniard} = \text{dog}, \)
• The Japanese is a painter:
  \( \text{japanese} = \text{painter}, \)
• The Italian drinks tea:
  \( \text{italian} = \text{tea}, \)
• The Norwegian lives in the first house on the left:
  \( \text{norwegian} = 1, \)
• The owner of the green house drinks coffee:
  \( \text{green} = \text{coffee}, \)
• The green house is on the right of the white house:
  \( \text{green} = \text{white} + 1, \)
• The sculptor breeds snails:  
  \textit{sculptor} = \textit{snails},

• The diplomat lives in the yellow house:  
  \textit{diplomat} = \textit{yellow},

• They drink milk in the middle house:  
  \textit{milk} = 3,

• The Norwegian lives next door to the blue house:  
  |\textit{norwegian} − \textit{blue}| = 1,

• The violinist drinks fruit juice:  
  \textit{violinist} = \textit{juice},

• The fox is in the house next to the doctor’s:  
  |\textit{fox} − \textit{doctor}| = 1,

• The horse is in the house next to the diplomat’s:  
  |\textit{horse} − \textit{diplomat}| = 1.
Fill the crossword grid with the words from:

- HOSES, LASER, SAILS, SHEET, STEER,
- HEEL, HIKE, KEEL, KNOT, LINE,
- AFT, ALE, EEL, LEE, TIE.

**Variables**: $x_1, \ldots, x_8$,

**Domains**: $x_7 \in \{\text{AFT, ALE, EEL, LEE, TIE}\}$, etc.

**Constraints**: one per crossing

$C_{1,2} := \{(\text{HOSES, SAILS}), (\text{HOSES, SHEET}),$

$(\text{HOSES, STEER}), (\text{LASER, SAILS}),$

$(\text{LASER, SHEET}), (\text{LASER, STEER})\}$

etc.
Consider the following problem.

The **meeting** ran non-stop the whole day.
Each person stayed at the meeting for a continuous period of time.
The **meeting** began while **Mr Jones** was present and finished while **Ms White** was present.
**Ms White** arrived after the meeting has began.
In turn, **Director Smith** was also present but he arrived after **Jones** had left.
**Mr Brown** talked to **Ms White** in presence of **Smith**.
Could possibly **Jones** and **White** have talked during this meeting?
Consider three events, A, B and C. **Known:** Temporal relations

- AB between A and B,
- BC between B and C.

**Question:** What is the temporal relation BC between A and C?

Allen ’83 defined a 13 × 13 table.

**Example:** if A overlaps B and B is before C, then A is before C.

This yields entry

allen(overlaps, before, before).

In total 409 entries.
<table>
<thead>
<tr>
<th>before</th>
<th>after</th>
<th>meets</th>
<th>met-by</th>
<th>overlaps</th>
<th>overl.-by</th>
</tr>
</thead>
<tbody>
<tr>
<td>before</td>
<td>before</td>
<td>TEMP</td>
<td>before</td>
<td>before meets</td>
<td>before meets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overlaps</td>
<td>overlaps starts</td>
<td>overlaps starts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>during</td>
<td>during</td>
<td>during</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>finishes</td>
<td>finishes</td>
<td>finishes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>after</td>
<td>after</td>
<td>after</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>met-by</td>
<td>met-by</td>
<td>met-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overl.-by</td>
<td>overl.-by</td>
<td>overl.-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>started-by</td>
<td>started-by</td>
<td>started-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>contains</td>
<td>contains</td>
<td>contains</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>finishes</td>
<td>finishes</td>
<td>finishes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>equals</td>
<td>equals</td>
<td>equals</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overlaps</td>
<td>overlaps</td>
<td>overlaps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overl.-by</td>
<td>overl.-by</td>
<td>overl.-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>during</td>
<td>during</td>
<td>during</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>meets</td>
<td>meets</td>
<td>meets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overlaps</td>
<td>overlaps</td>
<td>overlaps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>starts</td>
<td>starts</td>
<td>starts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overl.-by</td>
<td>overl.-by</td>
<td>overl.-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>contains</td>
<td>contains</td>
<td>contains</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>met-by</td>
<td>met-by</td>
<td>met-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overlaps</td>
<td>overlaps</td>
<td>overlaps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>contains</td>
<td>contains</td>
<td>contains</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>finished-by</td>
<td>finished-by</td>
<td>finished-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overl.-by</td>
<td>overl.-by</td>
<td>overl.-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>contains</td>
<td>contains</td>
<td>contains</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>started-by</td>
<td>started-by</td>
<td>started-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>met-by</td>
<td>met-by</td>
<td>met-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overlaps</td>
<td>overlaps</td>
<td>overlaps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>contains</td>
<td>contains</td>
<td>contains</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overl.-by</td>
<td>overl.-by</td>
<td>overl.-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>contains</td>
<td>contains</td>
<td>contains</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>met-by</td>
<td>met-by</td>
<td>met-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>meets</td>
<td>meets</td>
<td>meets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overl.-by</td>
<td>overl.-by</td>
<td>overl.-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>contains</td>
<td>contains</td>
<td>contains</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overl.-by</td>
<td>overl.-by</td>
<td>overl.-by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>contains</td>
<td>contains</td>
<td>contains</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overlaps</td>
<td>overlaps</td>
<td>overlaps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>overl.-by</td>
<td>overl.-by</td>
<td>overl.-by</td>
</tr>
<tr>
<td></td>
<td>starts</td>
<td>started-by</td>
<td>during</td>
<td>contains</td>
<td>finishes</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>------------</td>
<td>--------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>before</td>
<td>before</td>
<td>before</td>
<td>before</td>
<td>before</td>
<td>before</td>
</tr>
<tr>
<td>after</td>
<td>during finishes after overl.-by</td>
<td>after</td>
<td>during finishes after overl.-by</td>
<td>after</td>
<td>after</td>
</tr>
<tr>
<td>meets</td>
<td>meets</td>
<td>meets</td>
<td>overlaps starts during</td>
<td>before</td>
<td>overlaps starts during</td>
</tr>
<tr>
<td>met-by</td>
<td>during finishes overl.-by</td>
<td>after</td>
<td>during finishes overl.-by</td>
<td>after</td>
<td>met-by</td>
</tr>
<tr>
<td>overlaps</td>
<td>overlaps contains finished-by</td>
<td>overlaps starts during</td>
<td>before meets overlaps contains finished-by</td>
<td>overlaps starts during</td>
<td>before meets overlaps</td>
</tr>
<tr>
<td>overl.-by</td>
<td>during finishes overl.-by</td>
<td>after met-by overl.-by</td>
<td>during finishes overl.-by</td>
<td>after meets overl.-by started-by contains</td>
<td>overl.-by</td>
</tr>
<tr>
<td>starts</td>
<td>starts</td>
<td>starts</td>
<td>started-by equals</td>
<td>during</td>
<td>before meets overlaps contains finished-by</td>
</tr>
<tr>
<td>started-by</td>
<td>started-by equals</td>
<td>started-by</td>
<td>during finishes overl.-by</td>
<td>contains</td>
<td>overl.-by contains</td>
</tr>
<tr>
<td>during</td>
<td>during</td>
<td>during finishes after met-by overl.-by</td>
<td>during</td>
<td>TEMP</td>
<td>during</td>
</tr>
<tr>
<td>contains</td>
<td>overlaps contains finished-by</td>
<td>contains</td>
<td>R-OVERLAP</td>
<td>contains</td>
<td>overl.-by contains started-by</td>
</tr>
<tr>
<td>finishes</td>
<td>during</td>
<td>after met-by overl.-by</td>
<td>during</td>
<td>after met-by overl.-by started-by contains</td>
<td>finishes</td>
</tr>
<tr>
<td>finished-by</td>
<td>overlaps contains</td>
<td>overlaps starts during</td>
<td>contains</td>
<td>overlaps contains</td>
<td>finished-by</td>
</tr>
<tr>
<td>equals</td>
<td>starts</td>
<td>started-by</td>
<td>during</td>
<td>contains</td>
<td>finishes</td>
</tr>
</tbody>
</table>
• 5 events:
  - M (meeting),
  - J (Jones’s presence),
  - B (Brown’s presence),
  - S (Smith’s presence),
  - W (White’s presence).

• This yields 10 variables, each associated with an ordered pair of events and each with a domain:

  \[ TEMP := \{ \text{before, after, meets, met-by, overlaps,} \]
  \[ \text{overlapped-by, starts, started-by, during,} \]
  \[ \text{contains, finishes, finished-by, equals}\}, \]

  \[ REAL-OVERLAP := TEMP - \{ \text{before, after, meets, met-by} \} \]

  - \( x_{J,M} \in \{ \text{overlaps, contains, finished-by} \} \),
  - \( x_{M,W} \in \{ \text{overlaps} \} \),
  - \( x_{M,S} \in REAL-OVERLAP \),
  - \( x_{J,S} \in \{ \text{before} \} \),
  - \( x_{B,S} \in REAL-OVERLAP \),
  - \( x_{B,W} \in REAL-OVERLAP \),
  - \( x_{S,W} \in REAL-OVERLAP \),
  - \( x_{J,B}, x_{J,W}, x_{M,B} \in TEMP \).

**Final question**

Use rather

  - \( x_{J,W} \in REAL-OVERLAP \).

Is the above CSP consistent?
• *allen*: the composition table as a ternary relation (409 entries).

• For each ordered triple $A, B, C$ of the events a constraint $C_{A,B,C}$ on the variables $x_{A,B}$, $x_{B,C}$, $x_{A,C}$:

$$C_{A,B,C} := \text{allen} \cap (D_{A,B} \times D_{B,C} \times D_{A,C}).$$

where

$x_{A,B} \in D_{A,B}$,

$x_{B,C} \in D_{B,C}$,

$x_{A,C} \in D_{A,C}$.

• In total 10 constraints.
Consider the following problem.

Two houses are connected by a road. The first house is surrounded by its garden or is adjacent to its boundary while the second house is surrounded by its garden.

What can we conclude about the relation between the second garden and the road?
RCC8 := \{\text{disjoint, meet, equal, covers, coveredby, contains, inside, overlap}\}.
<table>
<thead>
<tr>
<th>disjoint</th>
<th>meet</th>
<th>equal</th>
<th>inside</th>
<th>coveredby</th>
<th>contains</th>
<th>covers</th>
<th>overlap</th>
</tr>
</thead>
<tbody>
<tr>
<td>disjoint</td>
<td>RCC8</td>
<td>disjoint meet inside coveredby overlap</td>
<td>disjoint</td>
<td>disjoint meet inside coveredby overlap</td>
<td>disjoint</td>
<td>disjoint meet inside coveredby overlap</td>
<td>disjoint</td>
</tr>
<tr>
<td>meet</td>
<td>disjoint meet contains covers overlap</td>
<td>disjoint meet equal coveredby covers overlap</td>
<td>meet</td>
<td>inside coveredby overlap</td>
<td>meet</td>
<td>inside coveredby overlap</td>
<td>disjoint meet inside coveredby overlap</td>
</tr>
<tr>
<td>equal</td>
<td>disjoint</td>
<td>meet</td>
<td>equal</td>
<td>inside</td>
<td>coveredby</td>
<td>contains</td>
<td>covers</td>
</tr>
<tr>
<td>inside</td>
<td>disjoint</td>
<td>disjoint</td>
<td>inside</td>
<td>inside</td>
<td>inside</td>
<td>RCC8</td>
<td>disjoint meet inside coveredby overlap</td>
</tr>
<tr>
<td>coveredby</td>
<td>disjoint</td>
<td>disjoint</td>
<td>coveredby</td>
<td>inside</td>
<td>inside</td>
<td>coveredby</td>
<td>disjoint meet contains covers overlap</td>
</tr>
<tr>
<td>contains</td>
<td>disjoint</td>
<td>contains covers overlap</td>
<td>contains</td>
<td>equal inside coveredby contains covers overlap</td>
<td>contains covers overlap</td>
<td>contains</td>
<td>contains covers overlap</td>
</tr>
<tr>
<td>covers</td>
<td>disjoint meet contains covers overlap</td>
<td>meet</td>
<td>covers</td>
<td>inside coveredby overlap</td>
<td>equal coveredby covers overlap</td>
<td>contains</td>
<td>contains covers overlap</td>
</tr>
<tr>
<td>overlap</td>
<td>disjoint meet contains covers overlap</td>
<td>disjoint meet contains covers overlap</td>
<td>overlap</td>
<td>inside coveredby overlap</td>
<td>inside coveredby overlap</td>
<td>disjoint meet contains covers overlap</td>
<td>disjoint meet contains covers overlap</td>
</tr>
</tbody>
</table>
Representation as a CSP

• 5 spatial objects:
  – $H_1$ (house 1),
  – $G_1$ (garden 1),
  – $H_2$ (house 2),
  – $G_2$ (garden 2),
  – $R$ (road).

• 10 variables with domains, each associated with an ordered pair of spatial objects:
  – $x_{H_1,G_1} \in \{\text{inside, coveredby}\}$,
  – $x_{H_2,G_2} \in \{\text{inside}\}$,
  – $x_{H_1,H_2} \in \{\text{disjoint}\}$,
  – $x_{H_1,R} \in \{\text{meet}\}$,
  – $x_{H_2,R} \in \{\text{meet}\}$,
  – $x_{G_1,G_2} \in \{\text{disjoint, meet}\}$,
  – $x_{H_1,G_2} \in \{\text{disjoint, meet}\}$,
  – $x_{G_1,H_2} \in \{\text{disjoint, meet}\}$,
  – $x_{G_1,R} \in \text{RCC8}$,
  – $x_{G_2,R} \in \text{RCC8}$. 
• \(S_3\): the composition table as a ternary relation (193 triples).

• For each ordered triple \(A,B,C\) of the objects a constraint \(C_{A,B,C}\) on the variables \(x_{A,B}, x_{B,C}, x_{A,C}\):

\[
C_{A,B,C} := S_3 \cap (D_{A,B} \times D_{B,C} \times D_{A,C}).
\]

where

\[
x_{A,B} \in D_{A,B},
\]

\[
x_{B,C} \in D_{B,C},
\]

\[
x_{A,C} \in D_{A,C}.
\]

• In total 10 constraints.
Four labels

• +, mark **convex** edges,
  (takes 270 degrees to rotate)

• −, mark **concave** edges,
  (takes 90 degrees to rotate)

• → and ← mark **boundary** edges
  (formed by two planes one of which is hidden).

Examples
Representation as a CSP

Variables: edges,
Domains: \{+, −, →, ←\},
Constraints: junctions.
Four type of constraints: \( L, \text{fork}, T, \text{and} \ arrow. \)

Example:
\[
L := \{(→, ←), (←, →), (+, →), \\
(←, +), (−, ←), (→, −)\}.
\]

des for as a CSP:

\begin{align*}
\text{arrow}(AC, AE, AB), \\
\text{fork}(BA, BF, BD), \\
L(CA, CD), \\
\text{arrow}(DG, DC, DB), \\
L(EF, EA), \\
\text{arrow}(FE, FG, FB), \\
L(GD, GF).
\end{align*}
Also needed
\[ edge := \{(+, +), (-, -), (\rightarrow, \leftarrow), (\leftarrow, \rightarrow)\}. \]

\( edge \) captures the complementary character of \( \rightarrow \) and \( \leftarrow \).

Nine constraints:
- \( edge(AB, BA) \)
- \( edge(AC, CA) \)
- \( edge(CD, DC) \)
- \( edge(BD, DB) \)
- \( edge(AE, EA) \)
- \( edge(EF, FE) \)
- \( edge(BF, FB) \)
- \( edge(FG, GF) \)
- \( edge(DG, GD) \).
Constrained Optimization Problems

- **Given:**
  - a CSP
    \[ P := \langle C ; x_1 \in D_1, \ldots, x_n \in D_n \rangle, \]
  - a function
    \[ obj : Sol \rightarrow \mathbb{R} \]

- \((P, obj)\) a **constrained optimization problem (COP).**

- **Task:** Find a solution \(d\) to \(P\) for which the value \(obj(d)\) is optimal (below: maximal).
Example: Knapsack Problem

**Given:** a knapsack of a fixed *volume* and \( n \) objects, each with a *volume* and a *value*. Find a collection of these objects with *maximal total value* that fits in the knapsack.

Representation as a COP:

**Given:** knapsack *volume* \( v \) and \( n \) objects with *volumes* \( a_1, \ldots, a_n \) and *values* \( b_1, \ldots, b_n \).

**Variables:** \( x_1, \ldots, x_n \),

**Domains:** \( \{0, 1\} \),

**Constraint:**
\[
\sum_{i=1}^{n} a_i \cdot x_i \leq v,
\]

**Objective function:**
\[
\sum_{i=1}^{n} b_i \cdot x_i.
\]
Example: Golomb Ruler

- **Golomb ruler with** $m$ **marks**: an ordered sequence of $m$ natural numbers such that the distance between any two elements in this sequence is **unique**.
- The largest element of a Golomb ruler is its **length**.
- An **optimum Golomb ruler with** $m$ **marks**: a Golomb ruler with $m$ marks with a **minimal** length.
0, 1, 4, 9, 11

is a Golomb ruler with 5 marks. Indeed, the distances are:

- for elements one apart: 1, 3, 5, 2,
- for elements two apart: 4, 8, 7,
- for elements three apart: 9, 10,
- for elements four apart: 11.

0, 1, 4, 9, 11 is an optimum Golomb ruler with 5 marks.

Largest known optimum Golomb ruler has 21 marks and is of length 333.
Fix $m$.

- **Pair**: two numbers $i, j$ such that $1 \leq i < j \leq m$.
- **Pairs** $i, j$ and $k, l$ are
  - *different* if $i \neq k$ or $j \neq l$,
  - *disjoint* if $i \neq k$ and $j \neq l$.
- **Example**:
  - 1,3 and 1,4 are different but not disjoint.
  - 1,3 and 2,4 are disjoint (and so different).

**Representation 1**

**Variables**: $x_1, \ldots, x_m$,

**Domains**: $\mathbb{N}$,

**Constraints**:

- $x_i < x_{i+1}$ for $i \in [1..m-1]$,
- $x_j - x_i \neq x_l - x_k$ for all different pairs $i, j$ and $k, l$.

**Objective function**: $-x_n$. 
Representations as a COP, ctd

Representation 2

Constraints:

• \( x_i < x_{i+1} \) for \( i \in [1..m - 1] \),
• \( x_j - x_i \neq x_l - x_k \) for all disjoint pairs \( i, j \) and \( k, l \).

Representation 3

Variables: \( x_1, \ldots, x_m, z_{i,j} \) for each pair \( i, j \),

Domains:
\( \mathcal{N} \) for \( x_1, \ldots, x_m \),
\( \mathcal{N} \setminus \{0\} \) for \( z_{i,j} \),

Constraints:

• \( z_{i,j} = x_j - x_i \) for each pair \( i, j \),
• \( z_{i,j} \neq z_{k,l} \) for all different pairs \( i, j \) and \( k, l \).

We can replace here “different” by “disjoint”.

Representation 4

Replace the disequality constraints by a single \texttt{all\_different} constraint on the variables \( z_{i,j} \).
Less Contrived Examples

• *A Microcode Label Assignment Problem*
  – CSP representation: 187 finite integer domain variables,
  – IP representation: 2024 Boolean variables,

• *A Packing Problem*
  – CSP representation: 7 finite integer domain variables, 2 constraints,
  – IP representation: 42 Boolean variables, 18 constraints,

• *A Golf Scheduling Problem*
  – CP representation: 176 variables,
  – IP representation 1: 2574 variables,
  – IP representation 2: 592 variables.
Objectives

● Define formally **Constraint Satisfaction Problems (CSP)**,

● **Modeling**: representation of a problem as a CSP.

● Clarify various aspects of modeling:
  – in general several natural representations exist,
  – some representations straightforward, some non-trivial,
  – some representations rely on a “background” theory.

● Show the generality of the notion of a CSP.