Lecture 1

Introduction

Constraint Programming

- Alternative approach to programming
- Combination of reasoning and computing
- Constraint on a sequence of variables: a relation on their domains
- Constraint Satisfaction Problem (CSP): a finite set of constraints

Constraint programming approach:

- Formulate your problem as CSP
- Solve the chosen representation using
  - domain specific methods, or
  - general methods

Solving CSPs

- Determine whether it has a solution (is consistent)
- Find a solution
- Find all solutions
- Find an *optimal* solution
- Find all *optimal* solutions

Domain Specific vs. General Methods

Domain specific methods:

Special purpose algorithms (constraint solvers), for example

- Program for solving systems of linear equations
- Package for linear programming
- Implementation of the unification algorithm

General Methods:

- Constraint propagation algorithms
- Search methods

Applications

- Interactive graphic systems
  (to express geometric coherence for scene analysis)
- Operations research problems
  (various optimization problems)
- Molecular biology
  (DNA sequencing, construction of 3D models of proteins)
- Electrical engineering (location of faults in the circuits, computing the circuit layouts, testing the design, verification)
- Natural language processing
  (construction of efficient parsers)
- Computer algebra
  (solving and/or simplifying equations over various algebraic structures)

Some Recent Applications

- Generation of coherent music radio programs
- Software engineering: design recovery and code optimization
- Selecting and scheduling satellite observations

Outline (of Today's Lecture)

- Define formally Constraint Satisfaction Problems (CSPs)
- Modeling: representing a problem as CSP
- Clarify various aspects of modeling:
  - in general there are several natural representations
  - some representations straightforward, some non-trivial
  - some representations rely on a “background” theory
- Show the generality of the notion of a CSP

Constraint Satisfaction Problem (CSP)

Given:

- Variables $Y := y_1, ..., y_k$
- Domains $D_1, ..., D_k$

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

Given:

- Variables $x_1, ..., x_n$
- Domains $D_1, ..., D_n$

Constraint Satisfaction Problem (CSP):

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

$\mathcal{C}$ – constraints, each on a subsequence of $x_1, ..., x_n$

$(d_1, ..., d_n) \in D_1 \times ... \times D_n$ is a solution to the CSP
if for every constraint $C \in \mathcal{C}$ on $x_{i_1}, ..., x_{i_m}$

$$(d_{i_1}, ..., d_{i_m}) \in C$$

Example: SEND + MORE = MONEY

Replace each letter by a different digit so that

$$\begin{array}{r} SEND \\ + \underline{MORE} \\ MONEY \end{array}$$

is a correct sum.

Unique solution:

$$\begin{array}{r} 9567 \\ + \underline{1085} \\ 10652 \end{array}$$

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 equality constraint:

$$\begin{aligned} & 1000 \cdot S + 100 \cdot E + 10 \cdot N + D \\ + & 1000 \cdot M + 100 \cdot O + 10 \cdot R + E \\ = & 10000 \cdot M + 1000 \cdot O + 100 \cdot N + 10 \cdot E + Y \end{aligned}$$

- 5 equality constraints:

$$D + E = 10 \cdot C_1 + Y$$
$$C_1 + N + R = 10 \cdot C_2 + E$$
$$C_2 + E + O = 10 \cdot C_3 + N$$
$$C_3 + S + M = 10 \cdot C_4 + O$$
$$C_4 = M$$

where $C_1, ..., C_4 \in [0..1]$ “carry” variables

Alternatives for Disequality Constraints

- 28 disequality constraints:

  $x \neq y$ for $x, y \in \{S, E, N, D, M, O, R, Y\}$, $x < y$

- 1 disequality constraint:

  `all_different`(S, E, N, D, M, O, R, Y)

- Modeling it as an IP (integer programming) problem:
  For $x, y \in \{S, E, N, D, M, O, R, Y\}$ transform $x \neq y$ to

  $x - y \leq 10 - 11 z_{x,y}$

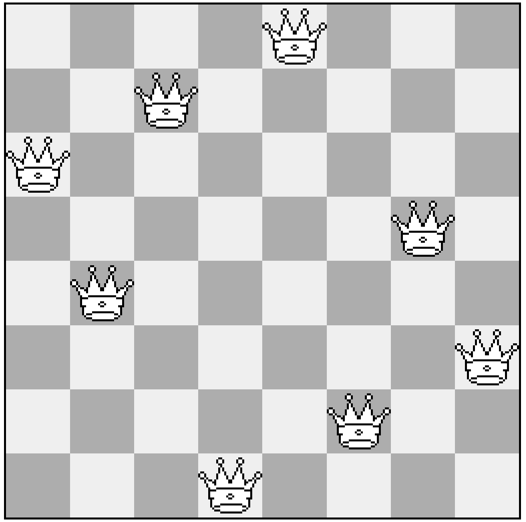
  $y - x \leq 11 z_{x,y} - 1$

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

  Disadvantage: 28 new variables

N Queens

Place n queens on an $n \cdot n$ chess board so that they do not attack each other.



Variables: $x_1, ..., x_n$

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)

Zebra Puzzle

A small street has five differently colored houses on it.

Five men of different nationalities live in them.

Each of them has a different profession, likes a different drink, and has a different pet animal.

Zebra Puzzle, ctd

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?

Zebra Puzzle, ctd

25 Variables:

- red, green, white, yellow, blue
- english, spaniard, japanese, italian, norwegian
- dog, snails, fox, horse, zebra
- painter, sculptor, diplomat, violinist, doctor
- tea, coffee, milk, juice, water

Domains: [1..5]

Constraints:

`all_different`(red, green, white, yellow, blue)

`all_different`(english, spaniard, japanese, italian, norwegian)

`all_different`(dog, snails, fox, horse, zebra)

`all_different`(painter, sculptor, diplomat, violinist, doctor)

`all_different`(tea, coffee, milk, juice, water)

Constraints, ctd

- The Englishman lives in the red house:
  english = red
- spaniard = dog
- japanese = painter
- italian = tea
- The Norwegian lives in the first house on the left:
  norwegian = 1
- green = coffee
- The green house is on the right of the white house:
  green = white + 1

Constraints, ctd

- sculptor = snails
- diplomat = yellow
- milk = 3
- The Norwegian lives next door to the blue house:
  |norwegian – blue| = 1
- violinist = juice
- The fox is in the house next to the doctor's:
  |fox – doctor| = 1
- |horse – diplomat| = 1

Crossword Puzzles

Fill the crossword grid with words from

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

Variables: $x_1, ..., x_8$

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

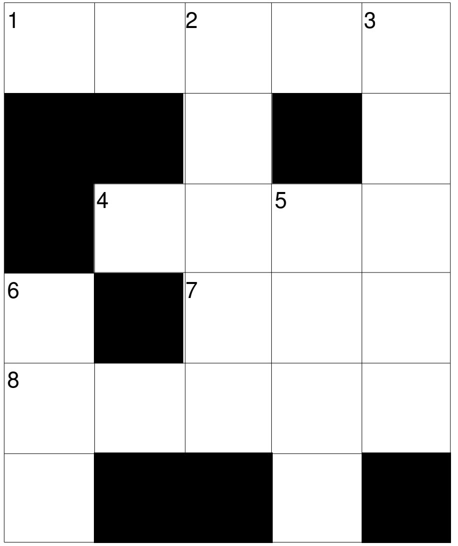
Constraints: one per crossing

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

(HOSES, STEER), (LASER, SAILS),

(LASER, SHEET), (LASER, STEER)}

etc.



Unique Solution

1 H	O	2 S	E	3 S
		A		T
	4 H	I	5 K	E
6 A		7 L	E	E
8 L	A	S	E	R
E			L	

Qualitative Temporal Reasoning

The meeting ran non-stop the whole day.

Each person stayed at the meeting for a continuos period of time.

The meeting began while Mr Jones was present and finished while Ms White was present.

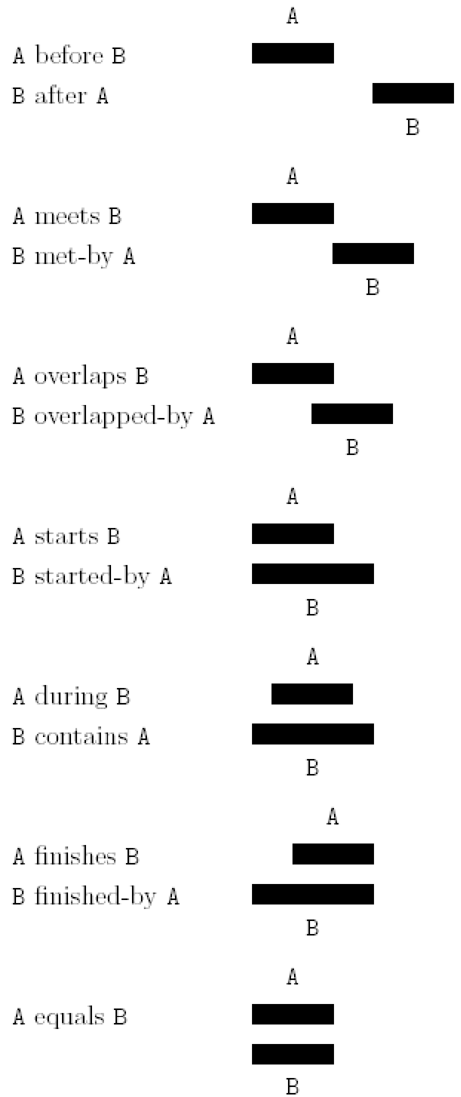
Ms White arrived after the beginning of the meeting.

Director Smith was also present, but he arrived after Jones had left.

Mr Brown talked to Ms White in the presence of Smith.

Could Jones and White possibly have talked during this meeting?

13 Temporal Relations (Allen 1983)



Composition Table

- Consider three events, `A`, `B`, and `C`
  Given the temporal relations between `A`, `B` and between `B`, `C`, what is the temporal relation between `A` and `C`?

- (Allen 1983) defines a 13 $\times$ 13 table:
  Example: if `A overlaps B` and `B before C`, then `A before C`
  This yields entry `allen(overlaps, before, before)`
  (In total 409 entries)

The Composition Table, Part 1

	before	after	meets	met-by	overlaps	overl.-by
before	before	*TEMP*	before	before meets overlaps starts during	before	before meets overlaps starts during
after	*TEMP*	after	during finishes after met-by overl.-by	after	during finishes after met-by overl.-by	after
meets	before	after met-by overl.-by started-by contains	before	finishes finished-by equals	before	overlaps starts during
met-by	before overlaps meets contains finished-by	after	starts started-by equals	after	during finishes overl.-by	after
overlaps	before	after met-by overl.-by started-by contains	before	overl.-by started-by contains	before meets overlaps	*R-OVERLAP*
overl.-by	before meets overlaps contains finished-by	after	overlaps contains finished-by	after	*R-OVERLAP*	after met-by overl.-by
starts	before	after	before	met-by	before	during

	before	after	meets	met-by	overlaps	overl.-by
started-by	before meets overlaps contains finished-by	after	overlaps contains finished-by	met-by	overlaps contains finished-by	overl.-by
during	before	after	before	after	before meets overlaps starts during	during finishes after met-by overl.-by
contains	before meets overlaps contains finished-by	after met-by overl.-by contains started-by	overlaps contains finished-by	overl.-by started-by contains	overlaps contains finished-by	overl.-by started-by contains
finishes	before	after	meets	after	overlaps starts during	after met-by overl.-by
finished-by	before	after met-by overl.-by started-by contains	meets	overl.-by started-by contains	overlaps	overl.-by started-by contains
equals	before	after	meets	met-by	overlaps	overl.-by

The Composition Table, Part 2

	starts	started-by	during	contains	finishes	finished-by	equals
before	before	before	before meets overlaps starts during	before	before meets overlaps starts during	before	before
after	during finishes after met-by overl.-by	after	during finishes after met-by overl.-by	after	after	after	after
meets	meets	meets	overlaps starts during	before	overlaps starts during	before	meets
met-by	during finishes overl.-by	after	during finishes overl.-by	after	met-by	met-by	met-by
overlaps	overlaps	overlaps contains finished-by	overlaps starts during	before meets overlaps contains finished-by	overlaps starts during	before meets overlaps	overlaps
overl.-by	during finishes overl.-by	after met-by overl.-by	during finishes overl.-by	after meets overl.-by started-by contains	overl.-by	overl.-by started-by contains	overl.-by
starts	starts	starts	during	before	during	before	starts

	starts	started-by	during	contains	finishes	finished-by	equals
started-by	starts started-by equals	started-by	during finishes overl.-by	contains	overl.-by	contains	started-by
during	during	during finishes after met-by overl.-by	during	*TEMP*	during	before meets overlaps starts during	during
contains	overlaps contains finished-by	contains	*R-OVERLAP*	contains	overl.-by contains started-by	contains	contains
finishes	during	after met-by overl.-by	during	after met-by overl.-by started-by contains	finishes	finishes finished-by equals	finishes
finished-by	overlaps	contains	overlaps starts during	contains	finishes finished-by equals	finished-by	finished-by
equals	starts	started-by	during	contains	finishes	finished-by	equals

Representation as CSP

- 5 events:
  - `M` (meeting)
  - `J` (Jones's presence)
  - `B` (Brown's presence)
  - `S` (Smith's presence)
  - `W` (White's presence)

- 10 variables, each associated with an ordered pair of events and each with a domain:
  TEMP := {`before, after, meets, met-by, overlaps, overlapped-by, starts, started-by, during, contains, finishes, finished-by, equals`}
  REAL-OVERLAP := *TEMP* - {`before, after, meets, met-by`}

Representation as CSP, ctd

- Constraints:
  - $x_{J,M} \in$ {overlaps, contains, finished-by}
  - $x_{M,W} \in$ {overlaps}
  - $x_{M,S} \in$ *REAL-OVERLAP*
  - $x_{J,S} \in$ {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
  If the constraint $x_{J,W} \in$ *REAL-OVERLAP* is added, is the CSP consistent?

Allen's Temporal Constraints

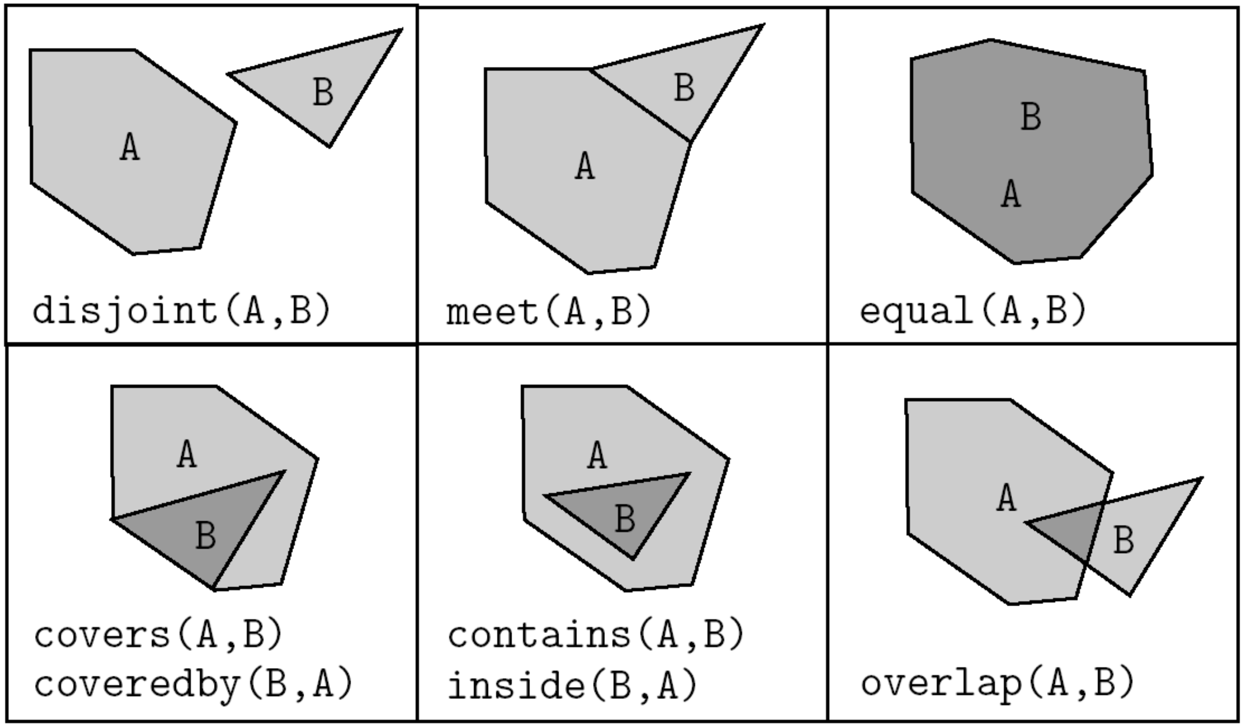
- `allen`: the composition table as a ternary relation (409 triples)
- For each ordered triple `A`, `B`, `C` of the events:
  a constraint $C_{A,B,C}$ on the variables $x_{A,B}$, $x_{B,C}$, $x_{A,C}$
  $C_{A,B,C} \coloneqq \texttt{allen} \cap (D_{A,B} \times D_{B,C} \times D_{A,C})$
  where
  $x_{A,B} \in D_{A,B}$
  $x_{B,C} \in D_{B,C}$
  $x_{A,C} \in D_{A,C}$

Qualitative Spatial Reasoning

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?

8 Spatial Relations



RCC8 $\coloneqq$ { `disjoint`, `meet`, `equal`, `covers`, `coveredby`, `contains`, `inside`, `overlap` }

The Composition Table for RCC8

	disjoint	meet	equal	inside	coveredby	contains	covers	overlap
disjoint	RCC8	disjoint meet inside coveredby overlap	disjoint	disjoint meet inside coveredby overlap	disjoint meet inside coveredby overlap	disjoint	disjoint	disjoint meet inside coveredby overlap
meet	disjoint meet contains covers overlap	disjoint meet equal coveredby covers overlap	meet	inside coveredby overlap	meet inside	disjoint	disjoint meet	disjoint meet inside coveredby overlap
equal	disjoint	meet	equal	inside	coveredby	contains	covers	overlap
inside	disjoint	disjoint	inside	inside	inside	RCC8	disjoint meet inside coveredby overlap	disjoint meet inside coveredby overlap
coveredby	disjoint	disjoint meet	coveredby	inside	inside coveredby	disjoint meet contains covers overlap	disjoint meet equal coveredby covers overlap	disjoint meet overlap coveredby overlap
contains	disjoint meet contains covers overlap	contains covers overlap	contains	equal inside coveredby contains covers overlap	contains covers overlap	contains	contains	contains covers overlap
covers	disjoint meet contains covers overlap	meet contains covers overlap	covers	inside coveredby overlap	equal coveredby covers overlap	contains	contains covers	contains covers overlap
overlap	disjoint meet contains covers overlap	disjoint meet contains covers overlap	overlap	inside coveredby overlap	inside coveredby overlap	disjoint meet contains covers overlap	disjoint meet contains covers overlap	RCC8

Representation as CSP

- 5 spatial objects: H1, H2, G1, G2, R
- 10 variables with domains, each associated with an ordered pair of spatial objects:
  - $x_{H1,G1} \in \{\texttt{inside}, \texttt{coveredby}\}$
  - $x_{H2,G2} \in \{\texttt{inside}\}$
  - $x_{H1,H2} \in \{\texttt{disjoint}\}$
  - $x_{H1,R} \in \{\texttt{meet}\}$
  - $x_{H2,R} \in \{\texttt{meet}\}$
  - $x_{G1,G2} \in \{\texttt{disjoint}, \texttt{meet}\}$
  - $x_{H1,G2} \in \{\texttt{disjoint}, \texttt{meet}\}$
  - $x_{G1,H2} \in \{\texttt{disjoint}, \texttt{meet}\}$
  - $x_{G1,R} \in$ RCC8
  - $x_{G2,R} \in$ RCC8

Constraints

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

Constrained Optimization Problem (COP)

- Given:
  - a CSP
    $P := \langle C ; x_1 \in D_1, \ldots, x_n \in D_n \rangle$
  - an objective function
    $obj : Sol \rightarrow R$

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

- Task: Find a solution *d* to P for which the value *obj*(*d*) is optimal (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; volumes $a_1, ..., a_n$; values $b_1, ..., b_n$

Variables: $x_1, ..., x_n$

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.

Optimum Golomb Ruler with 5 Marks

A Golomb ruler with 5 marks: 0, 1, 4, 9, 11

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

In fact, this is an optimum Golomb ruler with 5 marks.

The largest known optimum Golomb ruler has 21 marks and is of length 333.

Representations as a COP

- 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 $j \neq l$

Representation 1

Variables: $x_1, ..., x_m$

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: $x_1, ..., x_m$, $z_{i,j}$ for each pair i, j

Domains: $\mathbb{N}$ for $x_1, ..., x_m$

$\mathbb{N} \setminus \{0\}$ for $z_{i,j}$

Constraints:

- $z_{i,j} = x_\mathrm{j} - x_i$ for each pair i, j
- $z_{ij} \neq z_{k,l}$ for all different pairs i, j and k, l

Representation 4

Replace the disequality constraints by a single `all_different` on $z_{i,j}$

Different Representations as CSP

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 (of Today's Lecture)

- Define formally Constraint Satisfaction Problems (CSPs)
- Modeling: representing a problem as CSP
- Clarify various aspects of modeling:
  - in general there are several natural representations
  - some representations straightforward, some non-trivial
  - some representations rely on a “background” theory
- Show the generality of the notion of a CSP

Schedule

Lectures		Tutorials
Oct, 14th, 2009	Introduction	
Oct, 21st, 2009	CP in a Nutshell	Oct, 28th, 2009
Nov, 4th, 2009	Complete Constraint Solvers	Nov, 25th, 2009
Nov, 11th, 2009	Local Consistency	Dec, 2nd, 2009
Dec, 9th, 2009	Incomplete Constraint Solvers	Dec, 16th, 2009
Jan, 6th, 2010	Constraint Propagation	Jan, 13th, 2010
Jan, 20th, 2010	Search	Jan, 27th, 2010