Constraint Satisfaction Problems

Introduction
1 Introduction

- Constraint Satisfaction Problems
- Real World Applications
- Solving Constraints
- Contents of the lecture
Constraints

What is a constraint?

1 a: the act of constraining    b: the state of being checked, restricted, or compelled to avoid or perform some action . . .
c: a constraining condition, agency, or force . . .
2 a: repression of one’s own feelings, behavior, or actions    b: a sense of being constrained . . .

(from Merriam-Webster’s Online Dictionary)

Usage

- In programming languages, constraints are often used to restrict the domains of variables.
- In databases, constraints can be used to specify integrity conditions.
- In mathematics, a constraint is a requirement on solutions of optimization problems.
Examples:

- Latin squares
- Eight queens problem
- Sudoku
- Map coloring problem
- Boolean satisfiability
Problem:

- How can one fill an $n \times n$ table with $n$ different symbols
- ... such that each symbol occurs exactly once in each row and in each column?

There are 56 different *reduced* Latin squares of size 5, 9408 squares of size 6, 16.942.080 squares of size 7, 535.281.401.856 squares of size 8, ...
Sudoku

Problem:

- Fill a partially completed $9 \times 9$ grid such that
- ... each row, each column, and each of the nine $3 \times 3$ grids contains the numbers from 1 to 9.

```
 2 5 3 9 1 4 7 2 8
1 4 3 6 7 2 7 3 9
4 7 8 2 4 3 6 9 1
```

... (additional cells filled in the grid) ...
Eight queens puzzle

Problem:

- How can one put 8 queens on a standard chess board (8 × 8-board)
- …such that no queen can attack any other queen?

Solutions:

- The puzzle has 12 unique solutions (up to rotations and reflections)

- Old problem proposed in 1848.
- Various variants
  - knights (instead of queens)
  - 3D
  - n queens on an n × n-board
A solution of the 8-queens problem
Definition

A constraint network is defined by:

- a finite set of variables
- a (finite) domain of values for each variable
- a finite set of constraints (i.e., binary, ternary, ... relations defined between the variables)

Problem

Is there a solution of the network, i.e., an assignment of values to the variables such that all constraints are satisfied?
**$k$-Colorability**

**Problem:**
- Can one color the nodes of a given graph with $k$ colors such that all nodes connected by an edge have different colors?

Reformulated as a constraint network:
- **Variables:** the nodes of the graph
- **Domains:** “colors” $\{1, \ldots, k\}$ for each variable
- **Constraints:** nodes connected by an edge must have different values

This constraint network has a particular restricted form:
- only **binary** constraints
- domains are **finite**
Crossword puzzle

Problem instance:

- **Variables**: empty squares in a crossword puzzle;
- **Domains**: letters \( \{A, B, C, \ldots, Z\} \) for each variable;
- **Constraints**: relations defined by a given set of words that need (or are allowed) to occur in the completed puzzle.

```
  1  2  3  4  5  6  7  8  9 10 11 12 13  14 15 16 17 18 19 20 21 22 23 24 25
```

- **1**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **2**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **3**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **4**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **5**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **6**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **7**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **8**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **9**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **10**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **11**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **12**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **13**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **14**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **15**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **16**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **17**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **18**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **19**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **20**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **21**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **22**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **23**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **24**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
- **25**: [ ] [ ] [ ] [ ] [ ] [ ] [ ]
SAT

Given a propositional logic formula in CNF, is the formula satisfiable?

As a constraint satisfaction problem:

Problem instance (Boolean constraint network):

- **Variables**: (propositional) variables;
- **Domains**: truth values \( \{0, 1\} \) for each variable;
- **Constraints**: defined by a clause in the formula.

Example: \((x_1 \lor \neg x_2 \lor \neg x_3) \land (x_1 \lor x_2 \lor x_4)\)
Traveling salesperson problem (TSP):
Given a set of $n$ cities and distances $c_{ij}$ between city $i$ and city $j$, find the shortest route that visits all cities and finishes in the starting city.

TSP is not a constraint satisfaction problem, but a constraint optimization problem . . .
Vehicle routing problem (VRP):
Given a set of goods that need to be delivered from a central depot to customers, and given a fleet of trucks that can transport the goods: find an assignment of routes to the trucks that minimizes the total route cost.

Dozens of variants: Capacitated Vehicle Routing Problem (CVRP), ... with Pickup and Delivery (VRPPD), ... with time windows (CRPTW), ...
In practice, not only constraint satisfaction, but constraint optimization is required.

**Seminar topic assignment**

- Given $n$ students who want to participate in a seminar; $m$ topics are available to be worked on by students; each topic can be worked on by at most one student, and each student has preferences which topics s/he would like to work on;
- ... how to assign topics to students?
CSP/COP techniques can be used in

- **civil engineering** (design of power plants, water and energy supply, transportation and traffic infrastructure)
- **mechanical engineering** (design of machines, robots, vehicles)
- digital circuit **verification**
- automated timetabling
- air traffic control
- finance
Computational complexity

Theorem

*It is NP-hard to decide solvability of CSPs.*

Since $k$-colorability (SAT, 3SAT) is NP-complete, solvability of CSPs in general must be NP-hard.

Question: Is CSP solvability *in* NP?
Solving CSPs

- **Enumeration** of all assignments and testing
  \[\Rightarrow\] ... too costly

- **Backtracking** search
  \[\Rightarrow\] numerous different strategies, often “dead” search paths are explored extensively

- **Constraint propagation**: elimination of obviously impossible values

- Interleaving backtracking and constraint propagation: constraint propagation at each generated search node

- Many other search methods, e.g., local/stochastic search, etc.
Contents I

- *Introduction and mathematical background*
  - Sets, relations, graphs
  - Constraint networks and satisfiability
  - Binary constraint networks
  - Simple solution methods (backtracking, etc.)

- *Inference-based methods*
  - Arc and path consistency
  - \( k \)-consistency and global consistency

- *Search methods*
  - Backtracking
  - Backjumping
  - Comparing different methods
  - Stochastic local search
Contents II

- Global constraints
- Constraint optimization
- Selected advanced topics
  - Expressiveness vs complexity of constraint formalisms
  - Qualitative constraint networks
2 Organization

- Time, Location, Web
- Lecturers
- Exercises
- Course goals
- Literature
Lectures: Where, when, web page

Where
Bld. 51, Room SR 00 006

When
Monday, 10:15–12:00
Wednesday, 10:15–11:00 (+ exercises: 11:15–12:00)

No lectures
- 24-12-2014 – 06-01-2015 (Christmas break)

Web Page
http://www.informatik.uni-freiburg.de/~ki/teaching/ws1415/csp/
Lecturers

Prof. Bernhard Nebel
Bld. 52, Room 00-029
Phone: 0761/203-8221
Email: nebel@informatik.uni-freiburg.de

Dr. Christian Becker-Asano
Bld. 52, Room 00-042
Phone: 0761/203-8251
Email: basano@informatik.uni-freiburg.de

Dr. Stefan Wölfl
Bld. 52, Room 00-043
Phone: 0761/203-8228
Email: woelfl@informatik.uni-freiburg.de
**Exercises**

**Where**
Bld. 51, Room SR 00 006

**When**
Wednesday, 11:15–12:00
Course prerequisites & goals

Goals

- Acquiring skills in constraint processing
- Understanding the principles behind different solving techniques
- Being able to read and understand research literature in the area of constraint satisfaction
- Being able to complete a project (thesis) in this research area

Prerequisites

- Basic knowledge in the area of AI
- Basic knowledge in formal logic
- Basic knowledge in theoretical computer science
Exercises

Exercise assignments

- handed out on Wednesdays
- due on Wednesday in the following week (before the lecture)
- may be solved in groups of two students
- 50 % of reachable points are required for exam admission
Programming project

Implement a CSP solver . . .

- Implementation tasks are specified on a regular basis (depending on the progress of the lecture)
- Programming language
- Implementation should compile on a standard Linux computer (Ubuntu 13.08)
- We provide git repositories for source code
- Working solver is prerequisite for exam admission
- We will do a competition between solvers at the end of the lecture
Credit points

- 6 ECTS points

Exams

- (Oral or written) exam in February/March 2015
Projects and theses

Topics of theses resulting from this lecture:

- *Räumliche und zeitliche Constraints in beschreibungslogischen Wissensbasen*
- *Tableaux-Verfahren zur Lösung qualitativer CSPs*
- *Revisionsoperationen auf qualitativen Constraintnetzen*
- *Berechnung handhabbarer Klassen für qualitative räumliche Formalismen*
- *Fast procedures for the combination of qualitative constraint calculi*
Projects and theses

Topics of projects related to this lecture:

- *Ein Schwierigkeitsmaß für Sudoku-Puzzles*
- *Empirische Analyse von Konsistenz- und Suchalgorithmen*
- *Umsetzung eines CSP-Methoden-basierten Timetabling Algorithmus*
Acknowledgements

Lecture is based on slidesets of previous CSP lectures:

- Malte Helmert and Stefan Wölfli (summer term 2007)
- Bernhard Nebel and Stefan Wölfli (winter term 2009/10)
- Julien Hué and Stefan Wölfli (summer term 2012)

Special thanks to: Matthias Westphal, Robert Mattmüller, Gabriele Röger, Manuel Bodirsky
Literature

- Further readings will be given during the lecture.