Course Overview

Environment

Deterministic

Stochastic

Problem Type

Logic

Planning

Constraint Satisfaction

Variables + Constraints

Search

Arc Consistency

Search

Logics

STREPS

Bayesian Networks

Decision Networks

Markov Processes

Course Module

Representation

Reasoning

Technique

Now focus on CSPs
Lecture Overview

Arc consistency
- Recap
- GAC algorithm
- Complexity analysis
- Domain splitting
Arc Consistency

Definitions:
An arc \( <x, r(x,y)> \) is arc consistent if for each value \( x \) in \( \text{dom}(X) \) there is some value \( y \) in \( \text{dom}(Y) \) such that \( r(x,y) \) is satisfied.

A network is arc consistent if all its arcs are arc consistent.

Not arc consistent: No value in domain of B that satisfies \( A \prec B \) if \( A=3 \)

Arc consistent: Both \( B=2 \) and \( B=3 \) have ok values for \( A \) (e.g. \( A=1 \))
Arc Consistency

Arc consistent:
For each value in $\text{dom}(C)$, there is one in $\text{dom}(A)$ that satisfies $A > C$ (namely $A=3$)

Not arc consistent:
No value in domain of $B$ that satisfies $A < B$ if $A=3$
Arc Consistency

Not arc consistent anymore:
For C=2, there is no value in \( \text{dom}(A) \) that satisfies \( A \succ C \)
Which arcs need to be reconsidered?

- When we reduce the domain of a variable $X$ to make an arc $\langle X,c \rangle$ arc consistent, which arcs do we need to reconsider?

- You do not need to reconsider other arcs:
  - If arc $\langle Y,c \rangle$ was arc consistent before, it will still be arc consistent.
  - If an arc $\langle X,c' \rangle$ was arc consistent before, it will still be arc consistent.
  - Nothing changes for arcs of constraints not involving $X$.

Every arc $\langle Z,c' \rangle$ where $c' \neq c$ involves $Z$ and $X$: $Z_1$, $Z_2$, $Z_3$, $T$, $H$, $E$, $c_1$, $c_2$, $c_3$, $X$, $c$, $Y$, $c_4$, $A$. 
Lecture Overview

- Arc consistency
  - Recap
  - GAC algorithm
  - Complexity analysis
  - Domain splitting
Arc consistency algorithm (for binary constraints)

Procedure \texttt{GAC}(V,\text{dom},C)

Inputs

- \(V\): a set of variables
- \(\text{dom}\): a function such that \(\text{dom}(X)\) is the domain of variable \(X\)
- \(C\): set of constraints to be satisfied

Output

- arc-consistent domains for each variable

Local

- \(D_X\) is a set of values for each variable \(X\)
- \(\text{TDA}\) is a set of arcs

1: for each variable \(X\) do
2: \(D_X \leftarrow \text{dom}(X)\)
3: \(\text{TDA} \leftarrow \{ \langle X, c \rangle \mid X \in V, c \in C \text{ and } X \in \text{scope}(c) \}\)
4: while \((\text{TDA} \neq \{\})\) do
5: \(\text{select } \langle X, c \rangle \in \text{TDA}\)
6: \(\text{TDA} \leftarrow \text{TDA} \setminus \{ \langle X, c \rangle \}\)
7: \(\text{ND}_X \leftarrow \{x \mid x \in D_X \text{ and } \exists y \in D_Y \text{ s.t. (x, y) satisfies } c\}\)
8: if \((\text{ND}_X \neq D_X)\) then
9: \(\text{TDA} \leftarrow \text{TDA} \cup \{ \langle Z, c' \rangle \mid X \in \text{scope}(c'), c' \neq c, Z \in \text{scope}(c') \setminus \{X\} \}\)
10: \(D_X \leftarrow \text{ND}_X\)
11: return \(\{D_X\mid X \text{ is a variable}\}\)
Arc Consistency Algorithm: Interpreting Outcomes

Three possible outcomes (when all arcs are arc consistent):

1. Each domain has a single value, e.g.
   http://www.cs.ubc.ca/~mack/CS322/AIspace/simple-network.xml
   (Download the file and load it as a local file in Alspace consistency applet)
   And “Scheduling Problem 1” in Alspace.
   We have a (unique) solution.

2. At least one domain is empty, e.g.
   http://www.cs.ubc.ca/~mack/CS322/AIspace/simple-infeasible.xml
   (All values are ruled out for this variable.)
   No solution!

3. Some domains have more than one value, e.g.
   built-in example “Simple Problem 2” or “Scheduling Problem 2”
   There may be one solution, many solutions, or none.
   Need to solve this new CSP (usually simpler) problem:
   same constraints, domains have been reduced
Arc Consistency Algorithm: Complexity

- Worst-case complexity of arc consistency procedure on a problem with N variables
  - let $d$ be the max size of a variable domain
  - let $c$ be the number of constraints

- How often will we prune the domain of variable V? $O(d)$ times
- How many arcs will be put on the ToDoArc (TDA) list when pruning domain of variable V?
  - $O(\text{degree of variable V})$
  - In total, across all variables: sum of degrees of all variables = $\ldots$ 2*number of constraints, i.e. 2*c
- Together: we will only put $O(dc)$ arcs on the ToDoArc list (2c arcs originally on TDA)
- Checking consistency is $O(d^2)$ for each of them
- Overall complexity: $O(cd^3)$
- Compare to $O(d^N)$ of DFS! Arc consistency is MUCH faster.
Lecture Overview

- Arc consistency
  - Recap
  - GAC algorithm
  - Complexity analysis
- Domain splitting
Can we have an arc consistent network with non-empty domains that has no solution?

YES

NO

• Example: vars A, B, C with domain \{1, 2\} and constraints $A \neq B$, $B \neq C$, $A \neq C$

• Or see Alspace CSP applet Simple Problem 2
Domain splitting (or case analysis)

• Arc consistency ends: Some domains have more than one value → may or may not have a solution
  A. Apply Depth-First Search with Pruning or
  B. Split the problem in a number of disjoint cases:

    CSP with \( \text{dom}(X) = \{x_1, x_2, x_3, x_4\} \) becomes

    \( \text{CSP}_1 \) with \( \text{dom}(X) = \{x_1, x_2\} \) and
    \( \text{CSP}_2 \) with \( \text{dom}(X) = \{x_3, x_4\} \)

• Solution to CSP is the union of solutions to \( \text{CSP}_i \)
Whiteboard example for domain splitting

- ...

15
Domain splitting

- Each smaller CSP is easier to solve
  - Arc consistency might already solve it
- For each subCSP, which arcs have to be on the ToDoArcs list when we get the subCSP by splitting the domain of X?
  - arcs $<Z, r(Z,X)>$
  - arcs $<Z, r(Z,X)>$ and $<X, r(Z,X)>$
  - All arcs

```
A_1  T  H  E  S
A_2  C_1
A_3  C_2
X  C_3
Y  C_4
```
Domain splitting in action

• Trace it on “simple problem 2”
Searching by domain splitting

CSP, apply AC
If domains with multiple values
Split on one

CSP₁, apply AC
If domains with multiple values
Split on one

CSP₂, apply AC
If domains with multiple values.....Split on one

How many CSPs do we need to keep around at a time?
With depth m and 2 children at each split: $O(2^m)$. It’s a DFS.
Learning Goals for today’s class

• Define/read/write/trace/debug the **arc consistency algorithm**. Compute its complexity and assess its possible outcomes

• Define/read/write/trace/debug **domain splitting** and its integration with arc consistency

• Coming up: local search, Section 4.8