Domain Splitting

CPSC 322 – CSP 4

Textbook §4.6

February 4, 2011

Discussion of feedback

- Pace
  - 2 “fine”, 1 “could go faster”
  - 2: recap too long, 3: “sometimes rushed later (as a consequence)”
- Coloured card questions
  - Some more explanation would be good
  - More consistent: get everyone to vote”
- Which parts are most important?
  - Definitions + algorithms. Examples are for illustration
- Hard concepts:
  - Arc consistency: today + work in AIspace + practice exercise
  - Alternative formulation of CSP as graph search: after class

Discussion of feedback

- Midterm: review & sample questions?
  - Midterm date confirmed: Mon, Feb 28, 3pm (1 to 1.5 hours, TBD)
  - Sample midterm has been on WebCT for ~2 weeks
    - Topics: everything up to (including all of) CSP, plus planning (partially or all of it, TBD closer to midterm)
  - Should we do a midterm review session?
- More explanation of practice exercises?
  - I'll show where they are in WebCT
  - If you have trouble with them, please come to office hours
- How will what we learn eventually be applied in making an intelligent agent?
  - Game AI: lots of search
  - Reasoning under constraints is core to making intelligent decisions
    - With CSPs, we're right in the middle of it!

Course Overview

Course Module: *Representation* / Reasoning Technique

Environment

Problem Type		Deterministic	Stochastic	
Static	Constraint Satisfaction	***Variables + Constraints*** Arc Consistency Search		
Static	Logic	***Logics*** Search	***Bayesian Networks*** Variable Elimination	Uncertainty
Sequential	Planning	***STRIPS*** Search	***Decision Networks*** Variable Elimination ***Markov Processes*** Value Iteration	Decision Theory

We'll now focus on CSP

Lecture Overview

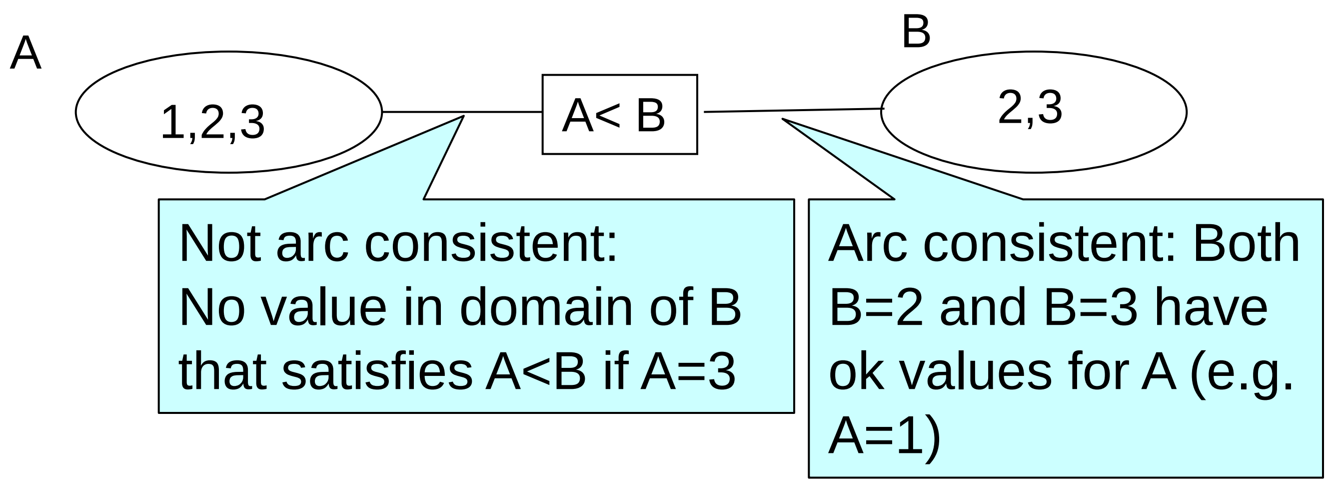
- Arc consistency
  - Recap
  - Complexity analysis
  - Domain Splitting
- Intro to Local Search

Arc Consistency

Definition:

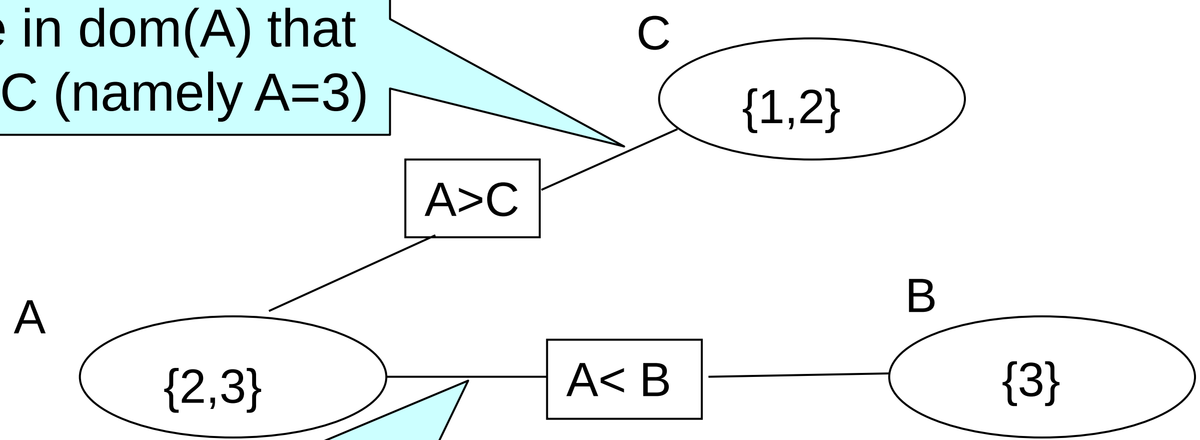
An arc <x, r(x,y)> is arc consistent if for each value x in dom(X) there is some value y in dom(Y) such that r(x,y) is satisfied.

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



Arc Consistency

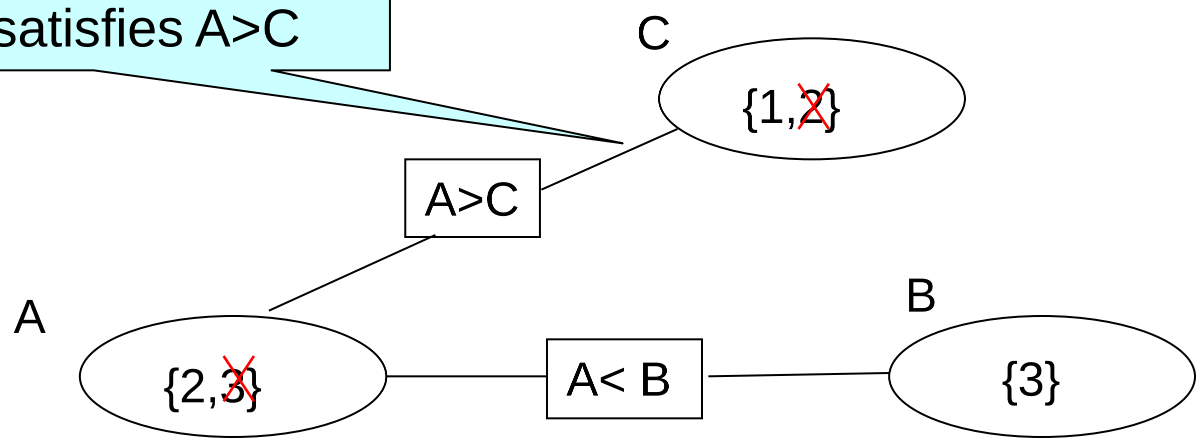
Arc consistent:
For each value in dom(C), there is one in dom(A) that satisfies A>C (namely A=3)



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

Arc Consistency

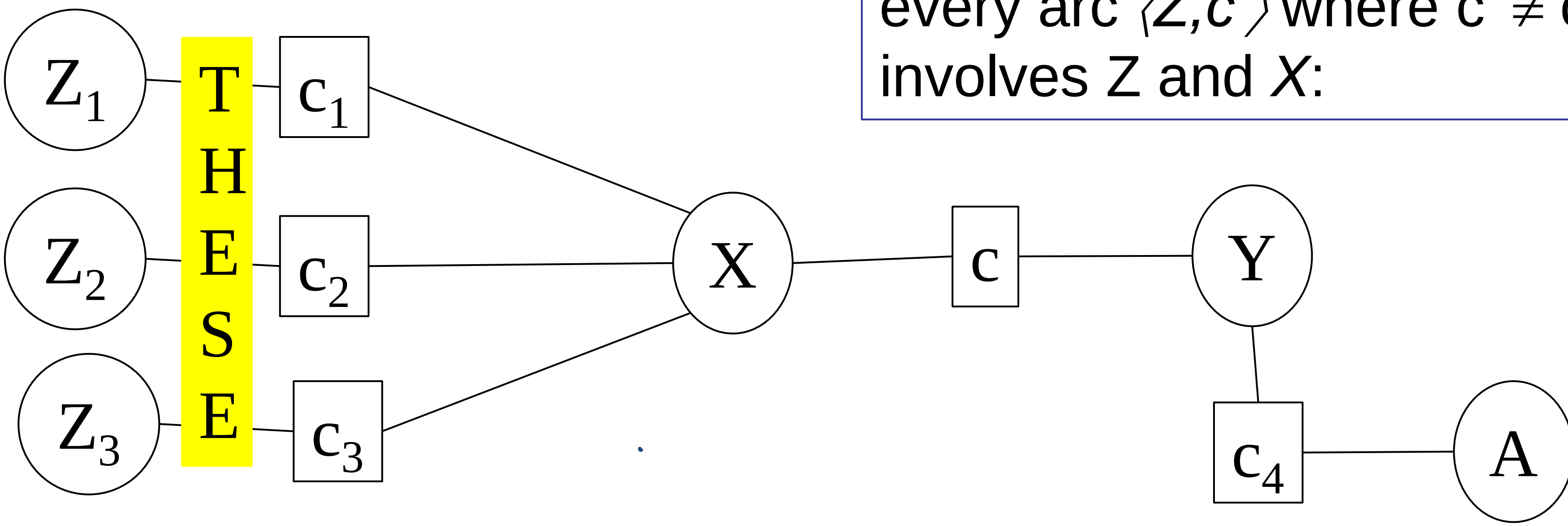
Not arc consistent anymore:
For C=2, there is no value in dom(A) that satisfies A>C



Which arcs need to reconsidered?

- When we reduce the domain of a variable X to make an arc ⟨X,c⟩ arc consistent, which arcs do we need to reconsider?

every arc ⟨*Z,c'*⟩ where c' ≠ c involves Z and *X*:



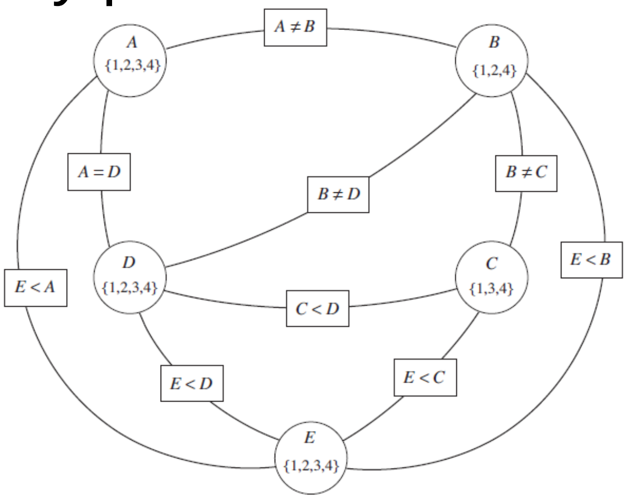
- You do not need to reconsider other arcs
  - If an arc ⟨X,c'⟩ was arc consistent before, it will still be arc consistent
  - Nothing changes for arcs of constraints not involving X

Lecture Overview

- Arc consistency
  - Recap
  - Complexity analysis
  - Domain Splitting
- Intro to Local Search

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 list when pruning domain of variable V?
    - O(degree of variable V)
    - In total, across all variables: sum of degrees of all variables = …
      - 2*number of constraints, i.e. 2*c
  - Together: we will only put O(dc) arcs on the ToDoArc list
  - 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
  - Complexity analysis
  - Domain Splitting
- Intro to Local Search

Can we have an arc consistent network with 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 AIspace CSP applet Simple Problem 2

Domain splitting (or case analysis)

- Arc consistency ends: Some domains have more than one value $\rightarrow$ 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 dom(X) = $\{x_1, x_2, x_3, x_4\}$ becomes

  CSP_1 with dom(X) = $\{x_1, x_2\}$ and
  CSP_2 with dom(X) = $\{x_3, x_4\}$

- Solution to CSP is the union of solutions to CSP_i

Whiteboard example for domain splitting

- ...

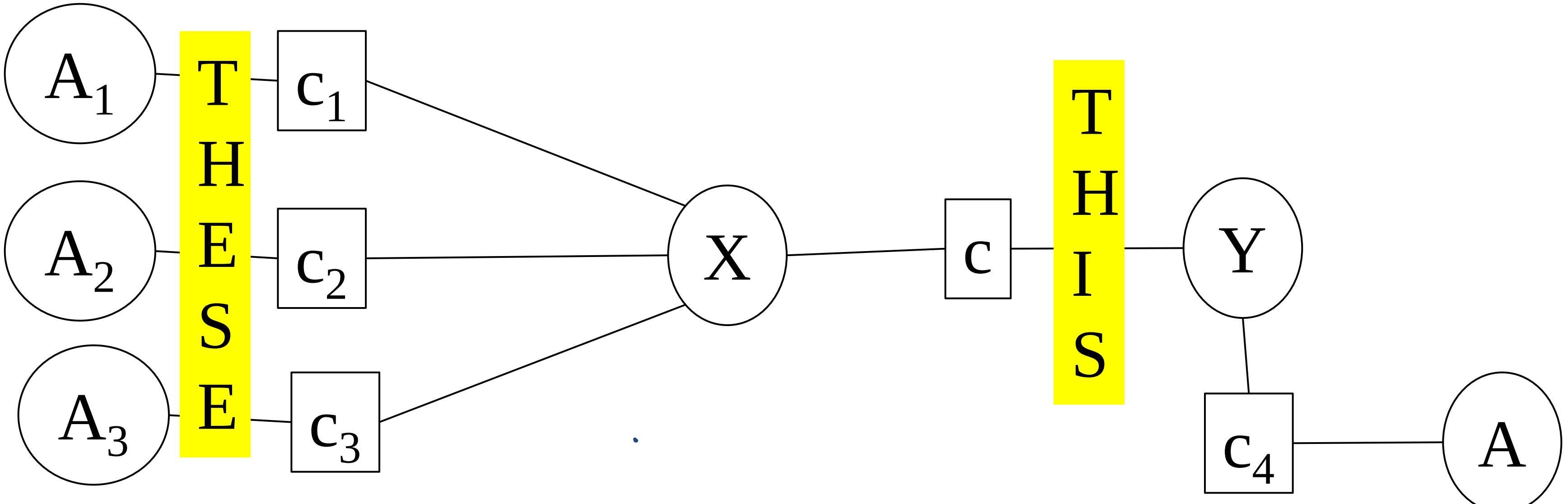
Domain splitting

- 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



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

CSP$_1$, apply AC

If domains with multiple values
Split on one

CSP$_n$, 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 b children at each split: O(bm). 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

---

- Assignment 1 is due on Monday
- Local search practice exercise is on WebCT
- Programming assignment (part of assignment #2) is available on WebCT (due Wednesday, Feb 23rd)
- Coming up: local search, Section 4.8