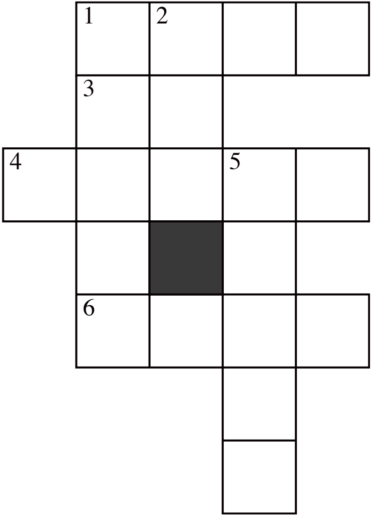
Constraint satisfaction revisited

- A Constraint Satisfaction problem consists of:
  - a set of variables
  - a set of possible values, a domain for each variable
  - a set of constraints amongst subsets of the variables
- The aim is to find a set of assignments that satisfies all constraints, or to find all such assignments.

Example: crossword puzzle



at, be, he, it, on,
eta, hat, her, him,
one,
desk, dove, easy,
else, help, kind,
soon, this,
dance, first, fuels,
given, haste, loses,
sense, sound, think,
usage

Dual Representations

Two ways to represent the crossword as a CSP

- First representation:
  - nodes represent word positions: 1-down…6-across
  - domains are the words
  - constraints specify that the letters on the intersections must be the same.
- Dual representation:
  - nodes represent the individual squares
  - domains are the letters
  - constraints specify that the words must fit

Representations for image interpretation

- First representation:
  - nodes represent the chains and regions
  - domains are the scene objects
  - constraints correspond to the intersections and adjacency
- Dual representation:
  - nodes represent the intersections
  - domains are the intersection labels
  - constraints specify that the chains must have same marking

Variable Elimination

- Idea: eliminate the variables one-by-one passing their constraints to their neighbours

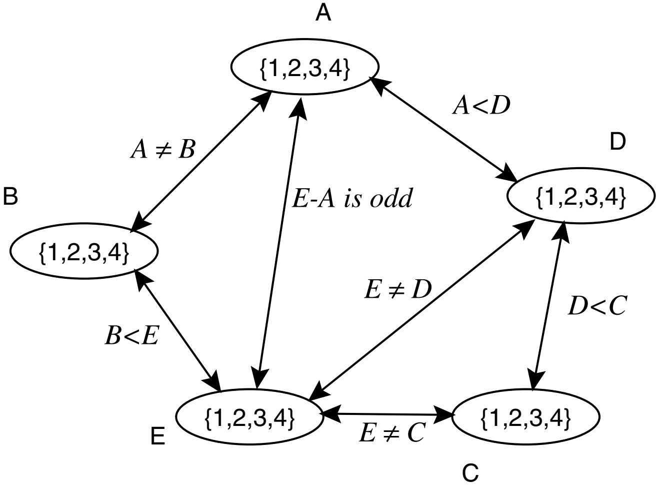
Variable Elimination Algorithm:

- If there is only one variable, return the intersection of the (unary) constraints that contain it
- Select a variable X
- Join the constraints in which X appears, forming constraint R_1
- Project R_1 onto its variables other than X, forming R_2
- Replace all of the constraints in which X_i appears by R_2
- Recursively solve the simplified problem, forming R_3
- Return R_1 joined with R_3

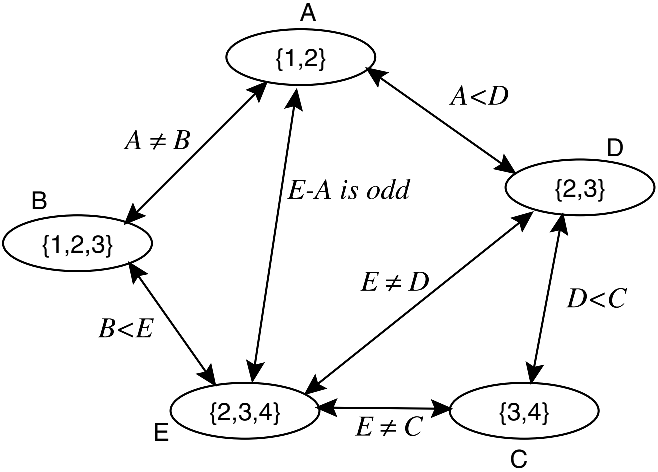
Variable elimination (cont.)

- When there is a single variable remaining, if it has no values, the network was inconsistent.
- The variables are eliminated according to some elimination ordering
- Different elimination orderings result in different size intermediate constraints.

Example network



Example: arc-consistent network



Example: eliminating C

$r_1 : C \neq E$	C	E
	3	2
	3	4
	4	2
	4	3

$r_2 : C > D$	C	D
	3	2
	4	2
	4	3

$r_3 : r_1 \bowtie r2$	C	D	E
	3	2	2
	3	2	4
	4	2	2
	4	2	3
	4	3	2
	4	3	3

$r_4 : \pi_{\{D,E\}} r_3$	D	E
	2	2
	2	3
	2	4
	3	2
	3	3

$\hookrightarrow$ new constraint

Resulting network after eliminating C