The Chinese University of Hong Kong

Weeks 10 and 11

In-class Survey

- Does the following lex_lesseq constraint have an answer?

```
lex_lesseq([0..1, 0..1, 1, 1, 0..1, 0],
           [0,    0..1, 0, 1, 1,    1])
```

  - A: what the hell is lex_lesseq?
  - B: definitely has a solution
  - C: probably has a solution
  - D: probably doesn't have a solution
  - E: definitely has no solutions

In-class Survey

- Does the following lex_lesseq constraint have an answer?

```
lex_lesseq([0..1, 0..1, 1, 1, 0..1, 1],
           [0,    0,    1, 1, 0,    0])
```

  - A: what the hell is lex_lesseq?
  - B: definitely has a solution
  - C: probably has a solution
  - D: probably doesn't have a solution
  - E: definitely has no solutions

In-class Survey

- Does the following value_precede_chain constraint have an answer?

```
value_precede_chain([0,2,4,1,3],
              [0..3, 1..3, 4..5, 0..3])
```

  - A: what the hell is value_precede_chain?
  - B: definitely has a solution
  - C: probably has a solution
  - D: probably doesn't have a solution
  - E: definitely has no solutions

Magic Squares of Numbers

- Arrange the numbers 1..$n*n$ in an $n*n$ square where
  - Every row adds to the same amount
  - Every column adds to the same amount
  - Both full diagonals add to the same amount
  - e.g. $n = 3$

4	3	8
9	5	1
2	7	6

The Chinese University of Hong Kong

Magic Squares of Numbers

- How many solutions
  - for $n = 3$?
  - for $n = 4$?
  - for $n = 5$?!

The Chinese University of Hong Kong

Magic Squares of Numbers

- Add a symmetry breaking constraint to your model
- Count the number of solutions again
  - for $n = 3$
  - for $n = 4$
  - for $n = 5$

Lex Least Solution

- How do we get the single lex least solution
  - Think about the 2d array as written out as 1d
  - E.g. [4,3,8,9,5,1,2,7,6]
- What about lex greatest?

More Squares of Numbers

- Modify your model to maximize the sum of the corner squares of the solution
- What if you want to maximize the weighted sum of the top left corner (2x2)
  - 4 * top left
  - 2 * orthogonal neighbours
  - 1 * diagonal neighbour

4*4	3*2	8*0
9*2	5*1	1*0
2*0	7*0	6*0

Remember

- The symmetry breaking constraints must be correct
  - Your problem must have symmetries in the first place!!
  - You are really breaking the correct symmetries
  - If you are breaking more than one symmetry, your symmetry breaking constraints must be compatible with each other

Let's look at Survey 9

The Chinese University of Hong Kong

In-class Survey

- Given domains

  D(X) = { -3, -2, 0, 1, 4 } & D(Y) = { -4, -1, 0, 2, 3 }

  What would a domain propagator for "Y = abs(X)" return?

  - A: no change in domains
  - B: D(X) = {0,1,4}, D(Y) = { -4, -1, 0, 2, 3 }
  - C: fail domains: D(X) = D(Y) = {}
  - D: D(X) = {-3,-2,0,1}, D(Y) = { 0, 2, 3 }
  - E: D(X) = {-3,-2,0}, D(Y) = {0,2,3}

In-class Survey

- Given domains

  D(X) = { -3, -2, 0, 1, 4 } & D(Y) = { -4, -1, 0, 2, 3 }

  What would a bounds(Z) propagator for "Y = abs(X)" return?
  - A: no change in domains
  - B: D(X) = {0,1,4}, D(Y) = { -4, -1, 0, 2, 3 }
  - C: fail domains: D(X) = D(Y) = {}
  - D: D(X) = {-3,-2,0,1}, D(Y) = { 0, 2, 3 }
  - E: D(X) = {-3,-2,0}, D(Y) = {0,2,3}

In-class Survey

- Given domains

  D(X) = { -3, -2, 0, 1, 4 } & D(Y) = { -4, -1, 0, 2, 3 }

  What would a bounds(R) propagator for "Y = abs(X)" return?
  - A: no change in domains
  - B: D(X) = {0,1,4}, D(Y) = { -4, -1, 0, 2, 3 }
  - C: fail domains: D(X) = D(Y) = {}
  - D: D(X) = {-3,-2,0,1}, D(Y) = { 0, 2, 3 }
  - E: D(X) = {-3,-2,0}, D(Y) = {0,2,3}

Recall Propagation Engine

- Propagation applies propagators $f \in F$ repeatedly until all at fixpoint $f(D) = D$
  - assume $f(D) = D$ for $f \in Fo$

isolv(Fo, Fn, D)

$F := Fo \cup Fn$; $Q := Fn$

while ($Q \neq \{\}$)

  f := choose(Q) % select next propagator

  $Q := Q - \{f\}$; $D' := f(D)$;

  $Q := Q \cup$ new(f,F,D,D') % read props

  $D := D'$

return D

Recall Propagation Engine

- $x = 2y \wedge x = 3z$, $D(x) = [0..17]$, $D(y) = [0..9]$, $D(z) = [0..6]$
- Propagators: f1 = ($x = 2y$), f2 = ($x = 3z$): Q = {f1,f2}
- write Q, f, domains, new (assuming all with vars changed)

The Chinese University of Hong Kong

Recall Propagation Engine

- $x = 2y \wedge x = 3z$, $D(x) = [0..17]$, $D(y) = [0..9]$, $D(z) = [0..6]$
- Propagators: f1 = ($x = 2y$), f2 = ($x = 3z$): Q = {f1,f2}
- write Q, f, domains, new (assuming all with vars changed)

Q	*f*	*D*(*x*)	*D*(*y*)	*D*(*z*)	new
f1,f2	f1	[0..17]	[0..8]	[0..6]	{f1}
f2,f1	f2	[0..17]	[0..8]	[0..5]	{f2}
f1,f2	f1	[0..16]	[0..8]	[0..5]	{f1,f2}
f2,f1	f2	[0..15]	[0..8]	[0..5]	{f1,f2}
f1,f2	f1	[0..15]	[0..7]	[0..5]	{f1}
f2,f1	f2	[0..15]	[0..7]	[0..5]	{}
f1	f1	[0..14]	[0..7]	[0..5]	{f1,f2}
f2,f1	f2	[0..14]	[0..7]	[0..4]	{f2}
f1,f2	f1	[0..14]	[0..7]	[0..4]	{}
f2	f2	[0..12]	[0..7]	[0..4]	{f1,f2}
f1,f2	f1	[0..12]	[0..6]	[0..4]	{f1}
f2,f1	f2	[0..12]	[0..6]	[0..4]	{}
f1	f1	[0..12]	[0..6]	[0..4]	{}

In-class Survey

- Domain propagators are always idempotent
- We usually apply bounds propagation on arithmetic constraints,
- The "basic" bounds propagator for $x = 2y$ is not idempotent. Why?
- Design an idempotent bounds propagator for $x = 2y$

In-class Survey

- Given $D(x) = [0..17]$ and $D(y) = [0..9]$. What would your idempotent bounds propagator for $x = 2y$ give?
  - A: no change in domains
  - B: $D(x) = [0..17]$ and $D(y) = [0..8]$
  - C: $D(x) = [0..16]$ and $D(y) = [0..8]$
  - D: $D(x) = [0..17]$ and $D(y) = [0..9]$
  - E: $D(x) = [0..18]$ and $D(y) = [0..9]$

Recall Propagation Engine

- $x = 2y \wedge x = 3z$, $D(x) = [0..17]$, $D(y) = [0..9]$, $D(z) = [0..6]$
- Propagators: f1 = ($x = 2y$), f2 = ($x = 3z$): Q = {f1,f2}
- write Q, f, domains, new (assuming all with vars changed)
- Assuming idempotent propagators won't get replaced in Q by their own variable domain changes

Recall Propagation Engine

- $x = 2y \wedge x = 3z$, $D(x) = [0..17]$, $D(y) = [0..9]$, $D(z) = [0..6]$
- Propagators: f1 = ($x = 2y$), f2 = ($x = 3z$): Q = {f1,f2}
- write Q, f, domains, new (assuming all with vars changed)
- Assuming idempotent propagators won't get replaced in Q by their own variable domain changes

Q	*f*	*D*(*x*)	*D*(*y*)	*D*(*z*)	**new**
f1,f2	f1	[0..16]	[0..8]	[0..6]	{f2}
f2	f2	[0..15]	[0..8]	[0..5]	{f1}
f1	f1	[0..14]	[0..7]	[0..5]	{f2}
f2	f2	[0..12]	[0..7]	[0..4]	{f1}
f1	f1	[0..12]	[0..6]	[0..4]	{}

Let's look at Survey 10

The Chinese University of Hong Kong