Discrete Optimization

Constraint Programming: Part III

Goals of the Lecture

- Illustrating the rich modeling language of constraint programming
- Key aspect of constraint programming
  - ability to state complex, idiosyncratic constraints

Magic Series

- A series S = (S_0,...,S_n) is magic if S_i represents the number of occurrences of i in S

	0	1	2	3	4
Occurrences	2	1	2	0	0

- Can you find a magic series?

Magic Series and Reification

```
int n = 5;
range D = 0..n-1;
var{int} series[D] in D;
solve {
   forall(k in D)
     series[k] = sum(i in D) (series[i]=k);
}
```

- Reification
  - the ability to transform a constraint into a 0/1 variable
  - the variable has the value 1 if the constraint is true and 0 otherwise

Magic Series and Reification

```
series[0] = (series[0]=0)+(series[1]=0)+(series[2]=0)+(series[3]=0)+(series[4]=0);
series[1] = (series[0]=1)+(series[1]=1)+(series[2]=1)+(series[3]=1)+(series[4]=1);
series[2] = (series[0]=2)+(series[1]=2)+(series[2]=2)+(series[3]=2)+(series[4]=2);
series[3] = (series[0]=3)+(series[1]=3)+(series[2]=3)+(series[3]=3)+(series[4]=3);
series[4] = (series[0]=4)+(series[1]=4)+(series[2]=4)+(series[3]=4)+(series[4]=4);
```

	0	1	2	3	4
Occurrences	?	?	?	?	?

Magic Series and Reification

```
series[0] = (series[0]=0)+(series[1]=0)+(series[2]=0)+(series[3]=0)+(series[4]=0);
series[1] = (series[0]=1)+(series[1]=1)+(series[2]=1)+(series[3]=1)+(series[4]=1);
series[2] = (series[0]=2)+(series[1]=2)+(series[2]=2)+(series[3]=2)+(series[4]=2);
series[3] = (series[0]=3)+(series[1]=3)+(series[2]=3)+(series[3]=3)+(series[4]=3);
series[4] = (series[0]=4)+(series[1]=4)+(series[2]=4)+(series[3]=4)+(series[4]=4);
```

- But what if series[1] != 1?

```
1         =                     (series[2]=0)+(series[3]=0)+(series[4]=0);
series[1] = 1 + (series[1]=1)+(series[2]=1)+(series[3]=1)+(series[4]=1);
series[2] =     (series[1]=2)+(series[2]=2)+(series[3]=2)+(series[4]=2);
series[3] =     (series[1]=3)+(series[2]=3)+(series[3]=3)+(series[4]=3);
series[4] =     (series[1]=4)+(series[2]=4)+(series[3]=4)+(series[4]=4);
```

Reification

- What is happening behind the scene?

```
int n = 5;
range D = 0..n-1;
var{int} series[D] in D;
solve {
   forall(k in D) {
      var{int} b[D] in 0..1;
      forall(i in D)
         booleq(b[i],series[i],k);
      series[k] = sum(i in D) b[i];
   }
}
```

- Constraint booleq(b,x,v) holds
  if (b=1 and x=v) or (b=0 and x≠v).

$$\text{booleq}(b,x,v) \Leftrightarrow (b=1 \wedge x=v) \vee (b=0 \wedge x \neq v)$$

Stable Marriages




- Basic assumptions
  - Every woman should provide a ranking of the men
  - Every man provides a ranking of the women




Stable Marriages

- A marriage between Hugh and Angelina is stable provided that
  - If Angelina prefers another man, say George, over Hugh, then George must prefer his spouse over Angelina
  - If Hugh prefers another woman, say Julia, over Angelina, then Julia must prefer her spouse over Hugh

- These stability rules make the marriage stable!

Stable Marriages

- What are the decision variables?

Stable Marriages

- Data and decision variables

`wrank[Hugh,Julia]` is the ranking of Julia in Hugh's preferences

```
enum Men = {George,Hugh,Will,Clive};
enum Women = {Julia,Halle,Angelina,Keira};

int wrank[Men,Women];
int mrank[Women,Men];
...

var{Women} wife[Men];
var{Men}   husband[Women];
```

`mrank[Julia,Hugh]` is the ranking of Hugh in Julia's preference

Stable Marriages

```
solve {
   forall(m in Men)
      husband[wife[m]] = m;
   forall(w in Women)
      wife[husband[w]] = w;

   ...
}
```

Stable Marriages

```
solve {
   forall(m in Men)
      husband[wi
   forall(w in Wo
      wife[husband[w]] = w;

   forall(m in Men, w in Women)
     wrank[m,w] < wrank[w,wife[m]] => mrank[w,husband[w]] < mrank[w,m];
   forall(w in Women, m in Men)
      mrank[w,m] < mrank[w,husband[m]] => wrank[m,wife[m]] < mrank[m,w];
}
```

m prefers w over his wife

Stable Marriages

```
solve {
   forall(m in Men)
      husband[wife[m]] = m;
   forall(w in Women)
      wife[husband[w]] = w;

   forall(m in Men, w in Women)
      wrank[m,w] < wrank[w,wife[m]] => mrank[w,husband[w]] < mrank[w,m];
   forall(w in Women, m in Men)
      mrank[w,m] < mrank[w,husband[m]] => wrank[m,wife[m]] < mrank[m,w];
}
```

w prefers her husband over m

Stable Marriages

```
enum Men = {George,Hugh,Will,Clive};
enum Women = {Julia,Halle,Angelina,Keira};
int wrank[Men,Women];
int mrank[Women,Men];
...
var{Women} wife[Men];
var{Men}   husband[Women];
solve {
   forall(m in Men)
      husband[wife[m]] = m;
   forall(w in Women)
      wife[husband[w]] = w;

   forall(m in Men, w in Women)
      wrank[m,w] < wrank[w,wife[m]] => mrank[w,husband[w]] < mrank[w,m];
   forall(w in Women, m in Men)
      mrank[w,m] < mrank[w,husband[m]] => wrank[m,wife[m]] < mrank[m,w];
}
```

Stable Marriages

- Two interesting features
  - Element constraint
    - useful in many applications
  - Logical combination of constraints
- The element constraint
  - ability to index an array/matrix with a variable or an expression containing variables
- Logical combination of constraints
  - can be handled by reification for instance

The Basic Element Constraint

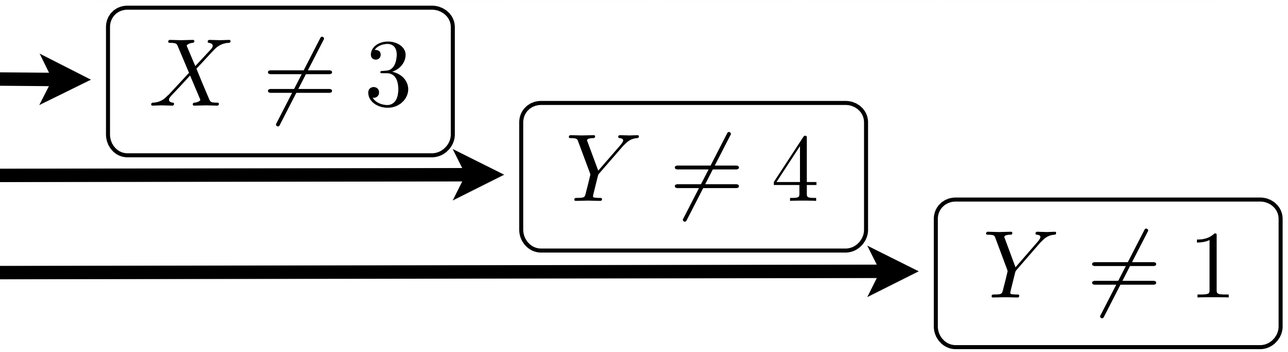
- x, y: variables
- c is an array of integers
- constraint x = c[y]

Y $\in \{0, 1, 2, 3, 4, 5\}$

X $\in \{3, 4, 5\}$

Array c

i	0	1	2	3	4	5
c[i]	3	4	5	5	4	3



Citations

66ème Festival de Venise (Mostra)(http://www.flickr.com/photos/nicogenin/3902889281/) by Nicolas Genin (http://www.flickr.com/photos/nicogenin/) CC BY-SA 2.0 (http://creativecommons.org/licenses/by-sa/2.0/deed.en)

Hugh Jackman (http://www.flickr.com/photos/58820009@N05/8294109236) by Eva Rinaldi (http://www.flickr.com/photos/evarinaldiphotography/) CC BY-SA 2.0 (http://creativecommons.org/licenses/by-sa/2.0/deed.en)

Wil Smith (http://www.flickr.com/photos/sandrascherer/6775228280/) by Sandra Scherer (http://www.flickr.com/people/sandrascherer/) CC BY-SA 2.0 (http://creativecommons.org/licenses/by-sa/2.0/deed.en)

Clive Owen, (http://www.flickr.com/photos/oneras/251106343/) by Mario Antonio Pena Zapateira (http://www.flickr.com/photos/oneras/) CC BY-SA 2.0 (http://creativecommons.org/licenses/by-sa/2.0/deed.en)

Julia Roberts 2011 Shankbone (http://www.flickr.com/photos/shankbone/5656729480/) by David Shankbone (http://www.flickr.com/people/shankbone/) CC BY 2.0 (http://creativecommons.org/licenses/by/2.0/deed.en)

Angelina Jolie (http://www.flickr.com/photos/gageskidmore/4860509634/) by Gage Skidmore (http://www.flickr.com/photos/gageskidmore/) CC BY-SA 2.0 (http://creativecommons.org/licenses/by-sa/2.0/deed.en)

Halle Berry 10 by German Marin [CC BY 3.0 (http://creativecommons.org/licenses/by/3.0)], via Wikimedia Commons (http://upload.wikimedia.org/wikipedia/commons/f/f8/Halle_Berry_10.jpg)

Keira Knightley (http://www.flickr.com/photos/tonyshek/7271339306/) by Tony Shek (http://www.flickr.com/photos/tonyshek/) CC BY-SA 2.0 (http://creativecommons.org/licenses/by-sa/2.0/deed.en)