A First Step into MiniZinc

Peter Stuckey

Overview

- A few simple example models in MiniZinc

MiniZinc

- MiniZinc is a modeling language being developed by NICTA with Univ of Melbourne and Monash University.
- Depending on the kind of model it can be solved with constraint programming or MIP or SAT or SMT techniques.
- It is a subset of the more powerful modeling language Zinc.

First Example: ToyProblem

- The problem:
- A toy manufacturer must determine how many bicycles, B, and tricycles, T, to make in a 40 hr week given that
  - the factory can produce 200 bicycles per hour or 140 tricycles
  - the profit for a bicycle is $25 and for a tricycle it is $30
  - no more than 6,000 bicycles and 4,000 tricycles can be sold in a week

$$\text{Maximise } 25B + 30T$$

$$\text{Subject to}$$

$$(1/200)B + (1/140)T \leq 40 \wedge$$

$$0 \leq B \leq 6000 \wedge 0 \leq T \leq 4000$$

A First MiniZinc Model

```
solve maximize 25*B + 30*T;

constraint 140*B+200*T <= 40*200*140;

var 0..6000: B;
var 0..4000: T;

output  ["B=\(B) T=\(T)\n"];
```

$$\text{Maximise } 25B + 30T$$

$$\text{Subject to}$$

$$(1/200)B + (1/140)T \leq 40 \wedge$$

$$0 \leq B \leq 6000 \wedge 0 \leq T \leq 4000$$

A First MiniZinc Model

```
var 0..6000: B;
var 0..4000: T;

constraint 140*B+200*T <= 40*200*140;

solve maximize 25*B + 30*T;

output  ["B=\(B) T=\(T)\n"];
```

$$\text{Maximise } 25B + 30T$$

$$\text{Subject to}$$

$$(1/200)B + (1/140)T \leq 40 \wedge$$

$$0 \leq B \leq 6000 \wedge 0 \leq T \leq 4000$$

A First MiniZinc Model

- We can run our MiniZinc model as follows

```
$ minizinc toyproblem.mzn
```

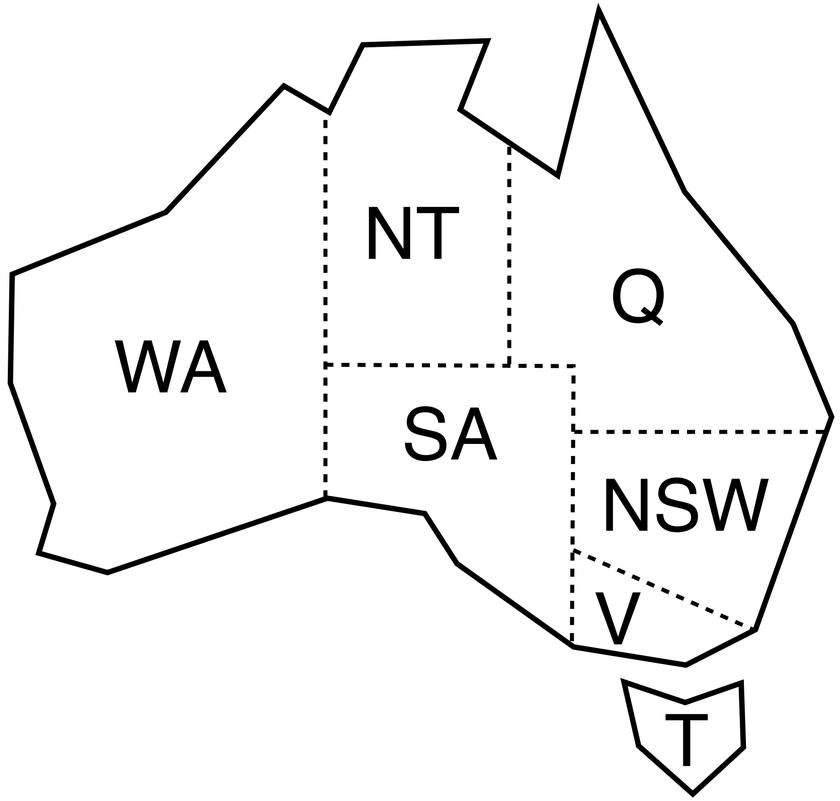
- This results in

```
B=6000 T=1400
----------
==========
```

- The line ---------- indicates a solution
- The line ========== indicates no better solution (that this is the best solution)
- MiniZinc models must end in .mzn
- There is also an IDE for MiniZinc

Second Example: AustColor

- Given a map of Australian states and territories
- Color it in so no two adjacent regions are colored the same.

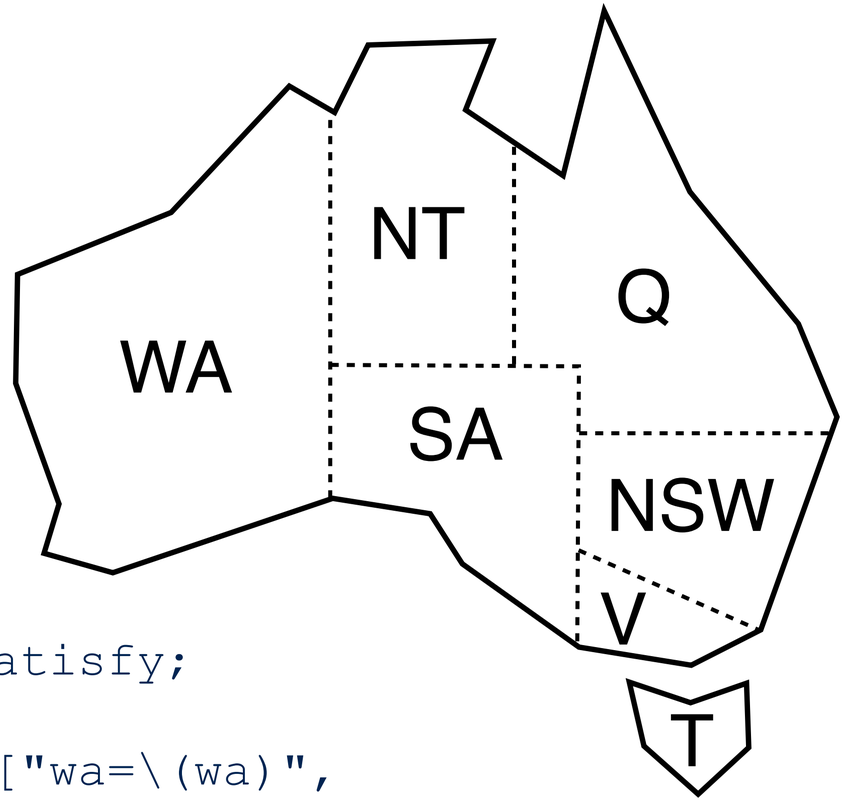


A Second MiniZinc Model

```
% Colouring Australia using 4 colors
int: nc = 4;

var 1..nc: wa;     var 1..nc: nt;
var 1..nc: sa;     var 1..nc: q;
var 1..nc: nsw;    var 1..nc: v;
var 1..nc: t;

constraint wa != nt;
constraint wa != sa;
constraint nt != sa;
constraint nt != q;
constraint sa != q;
constraint sa != nsw;
constraint sa != v;
constraint q != nsw;
constraint nsw != v;
```




```
solve satisfy;

output ["wa=\(wa)",
    " nt=\(nt)",
    " sa=\(sa)\n",
    "q=\(q)",
    " nsw=\(nsw)",
    " v=\(v)\n",
    "t=\(t)\n"];
```

A Second MiniZinc Model

- We can run our MiniZinc model as follows

```
$ minizinc aust_color.mzn
```

- This results in

```
wa=1 nt=3 sa=2
q=1 nsw=3 v=1
t=1
----------
```

- We can change the model to use 2 colors by instead using the line

```
int: nc = 2;
```

- This results in

```
=====UNSATISFIABLE=====
```

Overview

- Two examples models

- Optimization
  - ToyProblem

- Satisfaction
  - AustColor

EOF