Name:

Constraint-Based Models

Tutorial 8

The model files

- The model files that you will need are in:
  `/proj/courses/comppsych/Tutorial8/`
  Copy the entire directory to your account

For this tutorial we will experiment with the Competition-Integration model, a constraint-based model of (syntactic) ambiguity resolution developed by Michael Spivey and colleagues.

1. Running a model

The model in `model1.pl` specifies a toy model that consists of four constraints (constraint1 through constraint4) and two alternatives (alternative1 and alternative2). Run the model, and inspect the output:

```
swipl –f model1.pl
?- run_model.
```

a) Which alternative wins the competition? Why? And, after how many cycles?

b) Open `model1.pl` and compare the activation levels of `contraint1` prior to running the model to those after competition is resolved. What do you notice? Write down how this can happen.

c) To get a better grasp of the model's dynamics, set the set the value of `flag_verbose` to `true` in `model1.pl`. Now, reload and rerun the model. Identify the pivotal cycle for `contraint1`.

d) Try to alter the delta parameter. How does its value relate to the number of processing cycles required to resolve competition?

2. **Building a model**

The following table lists the constraints of the first region (e.g., *arrested by*) of a single item of the McRae et al. study discussed in the lecture.

	Weight	Main clause	Reduced relative
Thematic fit of initial NP	0.2763	5.8	0.6
Main clause bias	0.38205	0.92	0.08
Verb tense/voice	0.09165	0.495	0.505
by-bias	0.25	0.2	0.8

a) Open `model2.pl` and construct the input nodes for this item. See the comment block for `input_node/4`. Run the competition, and write down the outcome.

b) Is this an item from the "good agent" or the "good patient" condition and why?

c) What would the constraints for the other condition look like?

3. Analyzing a model

Let's now do some *pencil and paper* analysis of the Competition-Integration model. Assume that we have a model in which two alternatives are equally supported, e.g.:

```
input_node('constraint1','alternative1',1.0,0.5).
input_node('constraint1','alternative2',1.0,0.5).
```

a) Work out how long it will take for competition to end, given that $\Delta = 0.0075$.

b) What will happen to the constraint activations during competition?

c) Run `model3.pl` and verify your answers to (a) and (b). Now, add a slight imbalance to the constraints, what happens to this imbalance during competition?

d) Does your answer to (c) imply that prior to competition it is already known which alternative will win? Explain your answer.

e) **Bonus:** Provided your solution to (a), can you give a function $f(a,\Delta,c)$ that determines the maximum number of competition cycles, given the activation level of one of the alternatives a, the Δ parameter, and the current cycle c?