<div style="border: 1px solid black; padding: 10px;">

School of Cognitive and Computing Sciences

First Year School Course — CG019

COGNITIVE MODELLING

Computer Class Week 2: Summer Term 2002

Benedict du Boulay and Steve Torrance

</div>

There are two separate exercises for this class. You will be expected to write up your experiments with the system as part of a laboratory report for assessment (see the Course Outline).

1 Controlling a Blocks World

The purpose of this exercise is to enable you to experiment with a program that accepts commands and questions in English and relates these to a simple world of coloured blocks.

By the end of the exercise you should be able to (i) understand the stages of analysis between input and response and the different representations involved at each stage; (ii) understand the effects of different kinds of mal-formed input on the response of the system.

1.1 Getting Started

1. Log into the Unix system. You'll need to know your password. When you are logged in you'll see a row of icons along the bottom of the screen. Click on the one showing a picture of computer to obtain a window containing a Unix prompt e.g. `tsunx:`.

2. Start up the blocks program by typing
   `pop11 +gblocks`
   This will start up two windows. Use the mouse to get the windows in convenient places on the screen.

3. type `help`
   followed by pressing the $< RETURN >$ key in the window inviting this.

1.2 Finishing and Keeping a Record For the Laboratory Report

You will find it useful for your Report to have the record of your interactions with the system as your own file. *At the end of the session with this program* to cause a file named `myblocksrecord` to be written:

1. Press the $< ENTER >$ key and type `name myblocksrecord`

2. Press the $< RETURN >$ key

3. Press the $< ENTER >$ key and type `xx`

4. Press the $< RETURN >$ key

This will put a file called `myblocksrecord` containing a record of the interactions during this session into your file area and exit from the program back to the Unix prompt.

To print out that file on the dot matrix printer named "kgb" in 3D3 type the Unix command
`lp -d kgb myblocksrecord`

1.3 Finishing Without a Record

Type `bye` to the Blocks program and press $< RETURN >$.

1.4 Activities

Try out a range of commands and questions. Note the stages through which the analysis goes. Find out different ways in which the program can fail to understand what you typed or fail to execute your command. You will have to draw the parse-trees by hand to record them, though note that a textual representation is provided in the record file.

2 Buggy

The Buggy program pretends to be one of several children with misconception(s) about subtraction. It first invites you to give it a range of sums and shows you what answer the child would give. It then invites you to predict the answers the child would have given when it gives you the sums to do. The idea is for you to sufficiently understand the child's misconceptions so that you can predict how he or she would react.

By the end of the exercise you should be able to (i) mimic the incorrect subtraction behaviour of a range of children. (ii) understand the level of detail at which the underlying cognitive model of subtraction is constructed.

2.1 Getting Started

At the Unix prompt type
`pop11` $< RETURN >$
`lib buggy` $< RETURN >$
`go();` $< RETURN >$

and then follow the instructions of the program. When you are ready to be tested on your understanding of the child's misconception just press $< RETURN >$ in response to the prompt

```
Testing child b
        Top number:    ?
```

You should aim to understand the difficulties of all four of the available children named a, b, c and d. Keep a record by hand of what you do and how the system responds.

2.2 Finishing

Press $< RETURN >$ in response to the prompt
`Which child would you like to test:  a, b, c, or d`
and then type
`bye` $< RETURN >$
to get back to Unix.

Make sure to logout from Unix before leaving the terminal by clicking the EXIT icon at the bottom of the screen below the globe.