A functional program to play 3-D Noughts-and-Crosses

Functional programming (FP)

Miranda is a functional programming language that was developed by David Turner in 1985

- A program transforms input I to output O via a (typically complex) function: O = F(I)
  "complex" means "specified using a sophisticated logical formalism"
- Conceiving a program as a function offers a succint mathematical abstract view
- The Miranda interpreter is an ingenious evaluator of functions

Literate programming: human readability as a primary goal cf. what interaction offers over and above annotation

Input, output and interaction in FP

A standard function takes an input and returns an output without intervention

To capture *interaction*, as in *playing* "3D OXO", need to elaborate the notion of function and evaluation:

- Think of an entire game as converting "a stream of inputs" to a "stream of outputs"
- View as functional program (F) mapping stream of inputs (I) to stream of outputs (O)

Need to consider that

- the stream of inputs is not given in advance - it is determined as the game is played
- the stream of outputs has to be computed incrementally, so the player gets feedback

The Miranda interpreter evaluates F ("executes the program F") using *lazy evaluation*

How a "3D OXO" game is represented as a function:

The name of the functional program is "oxo". The function oxo is built up out of transactions, each of which is a function that *transforms a string that represents the entire history of the game so far into a string that represents the history of the game so far after a single move has been made (if legal) or ventured (if illegal)*

The mechanism for constructing oxo from the constituent transaction functions is explained in more detail in the italicised text added to the full listing of oxo.m as a "literate program".

Some key ideas regarding FP in relation to definitive programming (as in oxoJoy1994 for instance) are:

- *The string that represents the history of the game so far includes as sub-segments representations of all the board positions to date by strings of characters, separated by messages such as prompts, error and game status messages ... this approach abstracts away what - from an EM perspective - are very significant distinctions between observables such as board positions and error messages, and also abstracts away state change and agency.*

- *The difficulty of understanding the connection between the symbolic data references in the code and commonsense human-readable concepts (such as "who is the current player?") is illustrated in many ways. There are several examples of symbols in the program code and in the comments on the program that don't relate closely to their nearest external counterparts: e.g. 'squares' for 'cells', 'rows' for 'lines of cells', or* `inputdatum` *for tokens that only sometimes originate as direct player*

input (cf. `quickmove` which is generated "by the computer player"). The casual way in which terms of reference are framed is symptomatic of the mathematical mindset that functional programming invokes. It reflects the fact that almost all the nuances about the meanings of observables are dissolved in this style of programming, and the simple observables that we associate with a game get to be smeared into their traces when subjected to the standard and circumscribed behaviour in which they participate.

Reflections

- There are similarities between definitive programming and FP: functional programs can be viewed as definitive scripts
- Script construction is linked with emerging program comprehension
- It is difficult to connect variables in the FP script with observables in an actual OXO game:
    - A variable is associated with a highly artificial mode of observation that relies on the fact that ritualised patterns of interaction have been established
    - There is no scope for the open-ness to non-standard interaction that every day observables have
- Poor connection between components of functional program and familiar everyday observables undermines readability

Concluding thought

Where the detailed understanding or experiencing of a specific situation is concerned, formal specification of state, however sophisticated, does not have as much expressive power as an appropriately constructed interactive artefact.