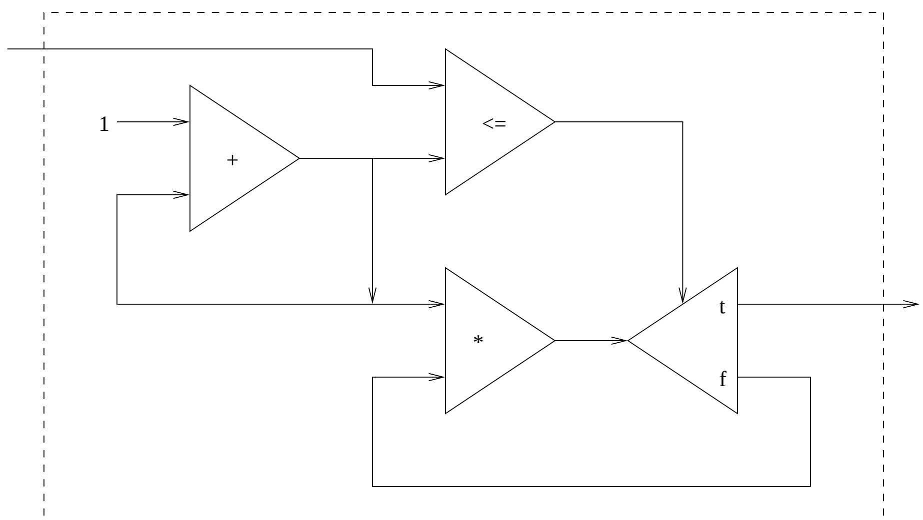
Department of Computer Science
Qualifying Examination
Visual Languages
Fall, 2001

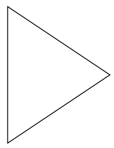
The following exam is open book and open notes. You may feel free to use whatever additional reference material you wish, but **no calculators** are allowed. Please note the following instructions.

- show your work whenever appropriate. There can be no partial credit unless I see how answers were obtained.
- be succinct. I will take points for facts that, while true, are not relevant to the question at hand.
- The questions are equally weighted.

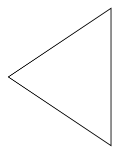
1. (50 points) The following figure shows a function written in a generic dataflow language:



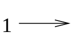
The function accepts a single token, and produces a single token. The intent is that this function should calculate the factorial of its input. The meanings of the boxes in the function are:



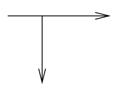
Function: takes one token from each of its inputs and produces a single output token, which is the result of applying the function to the inputs.



Selector: takes a data token from its left input and a logical token from its top input, and routes the data input to one of the outputs depending on the value of the logical token

1 

Constant: produces a single token, with the value of the number (1 in this case)



Duplicator: takes a single token on its input, and produces a copy of that token on each of its outputs.

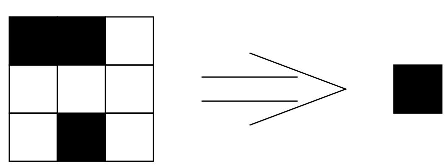
This function will not work as written.

(a) Give two reasons why the function will not successfully calculate the factorial.

(b) Fix one of the bugs. You may introduce new visual operators as necessary (but not a canned iterator or a canned factorial function).

(c) Discuss the generic dataflow language used in this problem (either as described above or as you modified it in part (b), if in fact you modified it) in terms of the following cognitive dimensions: visibility, consistency, premature commitment, error-proneness.

2. (50 points) Consider a rule-based visual language in which rules take the form of a 3×3 grid of cells on the left hand side, and a single cell on the right hand side. The left hand side is a pattern, and the cell in the center may change its value if the pattern matches. For example, the rule



means "if the cells immediately above, immediately below, and in the upper left diagonal relative to a white cell are black, and all of its other neighbors are white, turn the center cell black."

Conway's "Game of Life" is a very simple population model which is easily expressed in this language; its rules are described in English as follows:

- If a cell has 0 or 1 black neighbors, it will turn white (it dies of loneliness).
- If a cell has exactly two black neighbors it will stay whatever color it was previously.
- If a cell has exactly three black neighbors, it will turn black (they breed).
- If a cell has four or more black neighbors, it will turn white (it dies of overcrowding).

(a) One of the four cases listed does not require any rules to implement. Which is it?

(b) Draw a rule to serve as an example of each of the other three cases. For instance, nine rules are required to implement the first case; give one of the nine.

(c) Find a way to extend the given language so that you can express the Game of Life using fewer rules than would be necessary in the language as given (this is deliberately open-ended; there is not a unique correct answer. You are encouraged to use your imagination).