2002 Paper 6 Question 7

Prolog for Artificial Intelligence

A simple D-type flip-flop is represented by the Prolog predicate `dff` whose definition is as follows:

```
dff(D, 0, Q, Q).
dff(D, 1, Q, D).
```

The first argument is the input to the flip-flop, the second is the clock with `0` representing a falling edge and `1` representing a rising edge. The third and fourth arguments are the previous and next states of the flip-flop. As can be seen the state of the flip-flop changes on a rising edge of the clock.

A clocked circuit consists of three d-type flip-flops with inputs and states (D_1, Q_1), (D_2, Q_2) and (D_3, Q_3). They are wired in such a way that

$$D_1 = (Q_1 \wedge Q_2) \vee (\overline{Q_1} \wedge \overline{Q_2})$$
$$D_2 = (\overline{Q_1} \wedge Q_3) \vee (Q_2 \wedge \overline{Q_3})$$
$$D_3 = (Q_1 \wedge Q_3) \vee (\overline{Q_2} \wedge \overline{Q_3})$$

(a) Using `s(Q1, Q2, Q3)` to represent the state of the circuit, define a predicate that will compute the state after the next rising edge of the clock. You may find it helpful to define predicates to represent *and*, *or* and *not* gates. [14 marks]

(b) Define a predicate `testcc(N, s(Q1,Q2,Q3), List)` that will compute the list of states (`List`) through which the circuit passes from the given initial state `s(Q1,Q2,Q3)` as a result of a sequence of `N` rising edges of the clock. [6 marks]