Part 1: Model Checking

In lectures we saw Temporal Logics produced from natural language requirements (using FRET) so that system properties could be model checked. We also saw Linear Temporal Logic (LTL) used to express properties of systems for verification within the SPIN model checker, as in the examples on Moodle in the Spin examples (Part II) folder (e.g. target/safety1.pml and taret/semTL.pml).

For example, we model checked the file `safety1.pml` as follows:

```
> spin -a safety1.pml
> gcc -o pan pan . c
> ./ pan -N alwaysZero
> ./ pan -N atMostOne
```

Checking the property `alwaysZero` reported an error as the LTL property did not hold (it is invalid). Checking the property `atMostOne` reported no error as the LTL property did hold (it is valid).

Q1 [11 marks] Consider the following Promela model:

```
byte x = 0;
bool b = f a l s e
active proctype P () {
      do
         :: x < 20 -> x = 20; b = true
         :: x >= 0 -> if
                      :: x < 30 -> x ++
                      :: e l s e -> x = 15
                   fi
      od
}
```

Consider the following properties, each of which might or might not hold:

1. b will be true at some point.
2. x will always be ≥ 15.
3. At some point, x will be 15.
4. At some point, x will be 16.
5. From some point on, x will always be ≥ 16.
6. x will infinitely often be 16.
7. If b will never be true, then x will infinitely often be 16.

(a) Formulate each of the properties 1 - 7 in Linear Temporal Logic adding them to the Promela program above.
(b) For each of the properties 1 - 7, explain whether or not the property is valid in the transition system given by the above Promela model. Explain your answer.

Submission: Your submission should consist of a file called X_1.pml (where X is your student ID) containing the Promela program above with LTL representing properties 1-7, and an explanation of how you interacted with SPIN to obtain your solution. This explanation should be included as comments within the .pml file submitted.

Q2 [11 marks] Consider the following PROMELA model:

```
byte mode = 1;
byte count = 0;
```

```
active proctype m () {

endLoop :
     if
         :: mode = 1
         :: mode = 2
     fi ;
     do
        :: mode == 1 && count < 30 -> count ++
        :: mode == 2 -> count = 0; goto endLoop
        :: mode == 3 -> break
        :: e l s e -> goto endLoop;
     od;
       count = 0
 }


active proctype n () {
     do
        :: mode = 3
     od
  }
```

(a) Formalise the following properties in LTL and indicate for each whether it is valid or not valid with respect to the above PROMELA model. Assume the scheduler guarantees weak-fairness.

1. count is never greater-or-equal than 30.
2. If in some state count becomes greater than 0, it remains strictly positive until, eventually, mode becomes greater than 1 (Note: "until, eventually," means the strong until operator U, rather than weak until W).
3. If count is greater-than 0 in some state it will eventually be reset to 0 at some later point.
4. mode will eventually become 3.

(b) Explain the effect if weak fairness is no longer assumed i.e. For which of the four properties from (a) does this change the validity/invalidity status of the property?

Submission: Your submission should consist of a file called X_2.pml (where X is your student ID) containing the Promela program above with LTL representing properties 1-4, and an explanation of how you interacted with SPIN to obtain your solution. This explanation should be included as comments within the .pml file submitted.

PART II: Büchi Automata and Model Checking

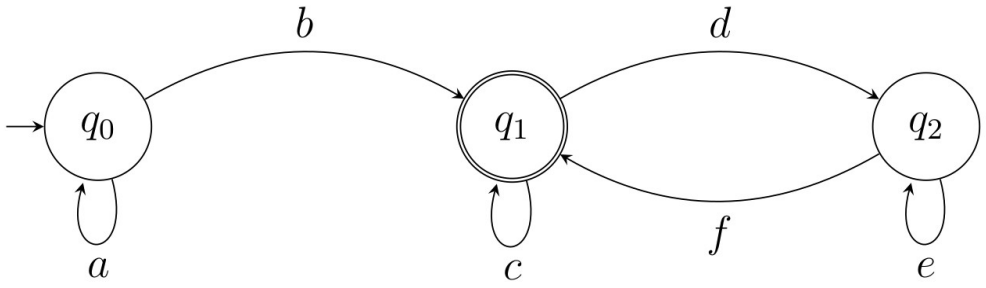
Submission: Your submission for Q3 should be a pdf document X_3.pdf (where X is your student ID). It may be a handwritten document scanned and uploaded as pdf.

Q3 [11 marks]. In lectures we discussed examples illustrating how Büchi automaton represent LTL formula e.g. **G F** r (expressed as **[] <> r)** over AP = {r, s}.

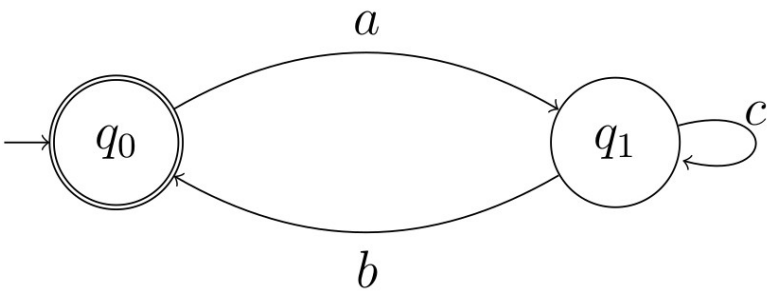
Buchi automata may also be used to represent languages. For example the ω expression

$$a^*b(c + (de^*f))^\omega$$

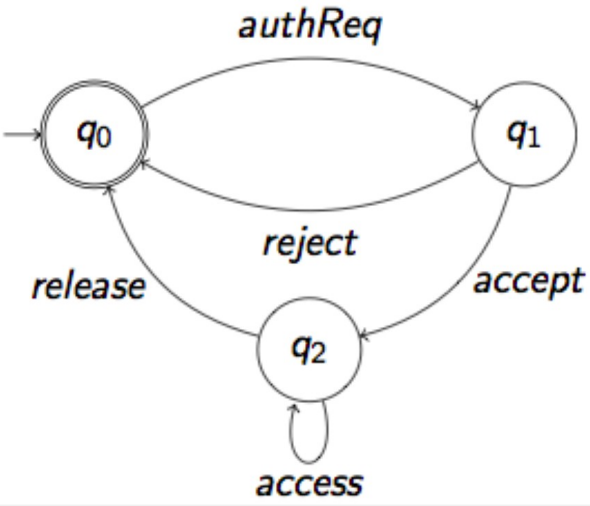
describes the language accepted by the Büchi automaton below. Note that a* represents 0 or more a's and $(a)^\omega$ denotes an infinite sequence of a's. In Büchi automaton, start-states are represented as arrows entering a state and end-states are represented with a double circle.



(a) Give the ω expression describing the language accepted by the following Büchi automaton. Explain your solution.



(b) Which language is accepted by the following Büchi automata? Explain your solution.



(c) Draw a Buchi automaton accepting exactly the language given by the ω-expression below. Explain your solution.

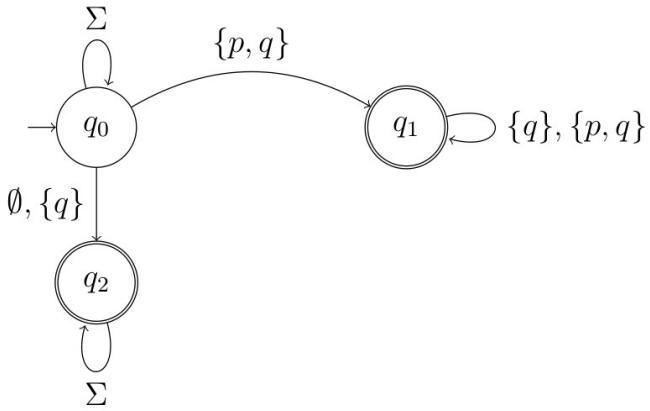
$$(a^*b^*c)^\omega$$

(d) Draw a Büchi automaton recognising exactly the language represented by the ω-expression below. Explain your solution.

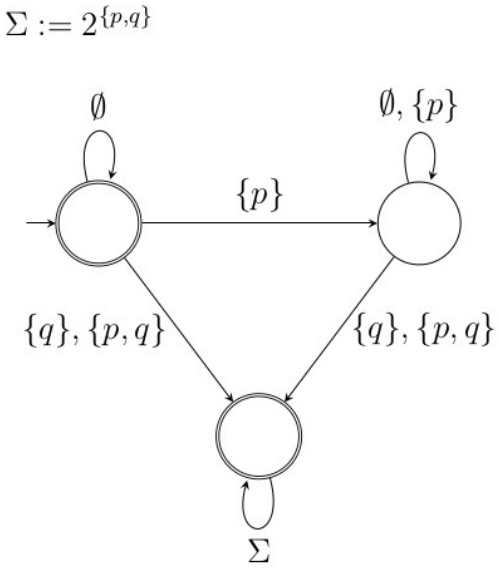
$$a\ ((aa)^*bb)^\omega$$

(e) Consider the LTL formula $\mathbf{F}(p \rightarrow \mathbf{G}\ q)$.
Does the following Büchi automaton accept exactly those runs satisfying the above formula? Explain your answer.

$\Sigma := \{\varnothing, \{p\}, \{q\}, \{p, q\}\}$



(f) Give an LTL formula that is satisfied by exactly those runs which are accepted by the following Buchi automaton. Explain your answer.



(g) Give a Büchi automaton that accepts exactly those runs satisfying the LTL formula below. Explain your answer.

G p ∨ **F** (p ∧ q)