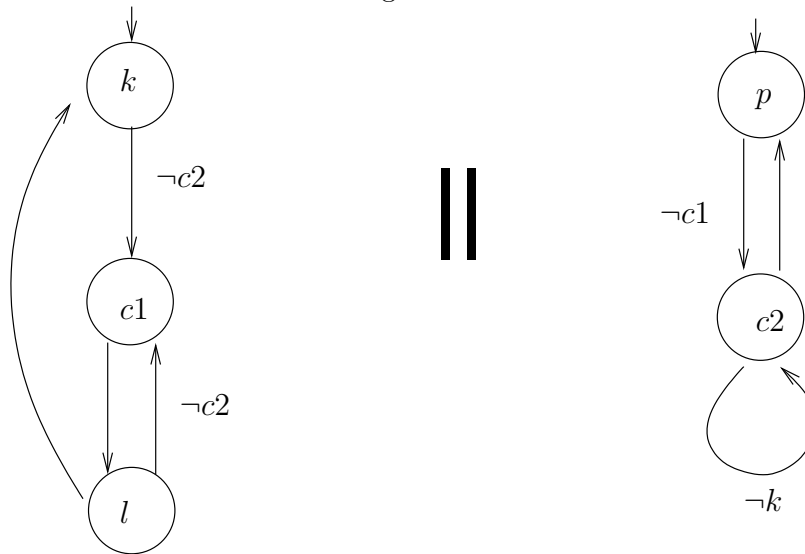


Assignment 1

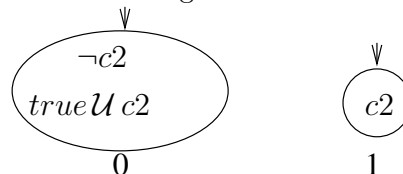
- a) Draw the transition diagram for the parallel system given in Figure 1. (4 pts.)

Figure 1:



- b) Using the *CTL* labeling model check algorithm, check whether or not $EFAFc1$ holds in the initial state of the system. (7 pts.)
- c) Using the *LTL* intersection of automata algorithm, check whether or not an (imaginary) formula, the negation of which being represented as a Street automaton, holds in the initial state of the system. The Street automaton atoms are given in Figure 2, the accepting condition is $\{\{0\}, \{1\}\}$. To complete the automaton, you only have to add the arrows induced by the formulae in the atoms. NB $c2$ is the only proposition of which the Street automaton gives the value, so, e.g., the Street automaton state with $c2$ combines with all states of the system that have $c2$, irrespective of the values of other propositions. (7 pts.)

Figure 2:



- d) Write the SMV code for the parallel system given in Figure 1, with the extra requirement that the system will not stay in $c2$ indefinitely. (7 pts.)

Assignment 2

- a) Give a transcript of the application of the search procedure (for formulae without equality and function symbols, see pp200–201 in Gallier) to the formula

$$(\exists x P(x)) \vee (\exists x Q(x)) \Rightarrow \exists x (P(x) \vee Q(x))$$

Indicate what rules are used and record the active variable information per round. (15 pts.)

b) The formula

$$(\exists x P(x)) \vee (\exists x Q(x)) \Rightarrow \forall x (P(x) \vee Q(x))$$

is not valid. Provide a first-order structure in which the formula does not hold. (10 pts.)

Assignment 3

a) Consider the following program, together with some annotation.

```

(| a = A ∧ b = B ∧ A > 0 ∧ B > 0 |)
(| invariant η : gcd(A, B) = gcd(a, b) |)
while (a ≠ b) {
  (| η ∧ a ≠ b |)
  if (a > b) {
    a = a - b;
  } else {
    b = b - a;
  }
  (| η |)
}
(| a = gcd(A, B) |)

```

Let us call this while-statement W . Give a complete derivation tree for

$$\vdash_{par} (| a = A \wedge b = B |) W (| a = \text{gcd}(A, B) |)$$

In your derivation, you may use B as an abbreviation for $a \neq b$ and P for the body of W , i.e., $\text{if } (a > b) \{ a = a - b; \} \text{ else } \{ b = b - a; \}$

For two non-zero integers x and y , the greatest common divisor, denoted as $\text{gcd}(x, y)$, is defined as the largest integer that is a divisor of both x and y . The following property holds for all non-zero x and y and $x \neq y$: $\text{gcd}(x - y, y) = \text{gcd}(x, y)$. (15 pts.)

- b) The property $\text{gcd}(x - y, y) = \text{gcd}(x, y)$ does not depend on whether $x > y$ or not. So what happens when we drop the test and the if-statement and always execute $a = a - b$? Is your proof still valid? Is the Hoare triple still valid? (4 pts.)
- c) Consider the program construct **repeat** P **until** B with meaning: execute P repeatedly until B holds; B is evaluated after each execution of P , so P will be executed at least once. Give one or more proof rules for this construct for partial correctness. (4 pts.)
- d) Give an example of the application of the rule by proving a Hoare triple containing a simple program with a repeat-until construct. You don't have to give the full derivation tree. Give a proof tableau (fully annotated program) and mention which proof rule is used and where ("justifications"). (2 pts.)

Assignment 4

- a) Using the Manna/Pnueli approach, prove that, under the usual assumption of justice for all transitions, for the parallel system given in Figure 1 it holds that $\Box \neg (c1 \wedge c2)$. (10 pts.)
- b) Explain why together with compassion for transition $k \rightarrow c1$ it holds that $k \Rightarrow \Diamond c1$. Give the name of the rule to prove this, the helpful transition required for the proof and the crucial clause in the rule for the proof. Also give the name of the rule to prove that clause (the proofs themselves are NOT required). (5 pts.)