Exercise 1 (5 Points, Resolution Calculus)
Consider the theory $T = \{\neg(p \land \neg q) \lor \neg(\neg s \land \neg t), \neg(t \lor q), u \rightarrow (\neg t \rightarrow (\neg s \land p))\}$
and use the resolution calculus to show that $T \models \neg u$.

Exercise 2 (8 Points, Resolution Calculus)
Consider the theory $T = \{c \lor d, (a \rightarrow (b \rightarrow c)), \neg c \leftrightarrow \neg d\}$.

(a) Use the Resolution calculus from Definition 2.37 to show that this theory entails $b \lor d$. Explain all the steps.

(b) Show that $T$ does not entail $b$.

(c) Is the following true: $T \vdash_{\text{Res}} b \lor d$? Prove or disprove.

(d) Is the following true: $T \vdash_{\text{Res}} c$? Prove or disprove.

Exercise 3 (6 Points, Verification)
Consider the following sequential hardware circuits:

(a) Give the transition systems for both hardware circuits. Assume that the initial values of the registers are $r_1 = 1$ and $r_2 = 0$.

(b) Determine the reachable part of the transition system of the synchronous product of the transition systems from (a).

Exercise 4 (12 Points, Verification)
The following program is a mutual exclusion protocol for two processes due to Pnueli. There is a single shared variable $s$ which is either 0 or 1, and initially 1. Besides, each process has a local Boolean variable $y$ that initially equals 0. The program text for process $P_i(i = 0, 1)$ is as follows:

```
while true do
    Noncritical section
    $(y_i, s) := (1, i);
```
wait until \((y_{i-1} = 0) \lor (s \neq i)\);

Critical section

\(y_i := 0\)

end while

Here, the statement \((y_i, s) := (1, i)\) is a multiple assignment in which variable \(y_i := 1\) and \(s := i\) is a single, atomic step.

a) Define the program graph of a process in Pnueli’s algorithm.

b) Determine the transition system for each process.

c) Construct their parallel composition. (Hint: Construct interleaving of program graphs and then the unfolding into a transition system. Compare with Figures 3.21 and 3.22 on Slides 201 and 202.)

d) Check whether the algorithm ensures mutual exclusion.

e) Check whether the algorithm ensures starvation freedom.

**Hint:** The last two questions may be answered by inspecting the transition system.

**Exercise 5 (0 Points, Extra (15P): LTL expressibility)**

We have seen that many interesting properties can be expressed with LTL formulae. In this exercise, we ask for a proof that a certain property *cannot be expressed with any LTL formula.*

We consider Example 3.39 of the lecture (Slide 270). *There is no LTL formula to express that “relation \(r\) is true at all even states (evenness)”.* Odd states do not matter: \(r\) can be true or not. Prove this statement.

**Hint:** Find two infinite series of models (paths) \(\lambda_j, \lambda'_j, j = 0, 1, \ldots\) such that the following holds: (1) for all \(j\) the property evenness is either true at \(\lambda_j\) or at \(\lambda'_j\), and (2) for all \(j\) and for all LTL formulae \(\varphi\) of depth at most \(j\), \(\lambda_j\) and \(\lambda'_j\) cannot be distinguished (i.e. \(\lambda_j \models \varphi\) if and only if \(\lambda'_j \models \varphi\) for all such \(\varphi\)). Define *depth* of a LTL formula so that it all works.

**Comment:** In fact, LTL can express all properties of *-free \(\omega\) regular languages. The property above is not of this kind: one needs the Kleene-* to express it for infinite words.