Name: _____ Matriculation Number: _____

Prof. Dr. Andreas Podelski, Tanja Schindler

Mock Exam for the Lecture
Cyber-Physical Systems – Discrete Models
WS 2017/18

In the exam, there will be 9 exercises to solve and you will have 90 minutes.

This mock exam consists of 18 exercises. You should be able to solve them within 180 minutes.

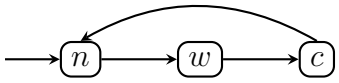
Exercise	Points
1. Regular Properties & Model Checking I	of 20
2. Regular Properties & Model Checking II	of 10
3. Regular Properties & Model Checking III	of 10
4. Regular Properties & Model Checking IV	of 10
5. Regular Properties & Model Checking V	of 10
6. Regular Properties & Model Checking VI	of 10
7. Linear-Time Properties I	of 10
8. Linear-Time Properties II	of 10
9. Linear-Time Properties III	of 10
10. Linear-Time Properties IV	of 10
11. Linear-Time Properties V	of 10
12. Linear-Time Properties VI	of 10
13. Modeling Concurrent Systems I	of 10
14. Modeling Concurrent Systems II	of 10
15. Modeling Concurrent Systems III	of 10
16. Modeling Concurrent Systems IV	of 10
17. Modeling Concurrent Systems V	of 10
18. Modeling Concurrent Systems VI	of 10
Total Points	of 190

Name: _____ Matriculation Number: _____

Exercise 1 (Regular Properties I) **20**

The goal of this exercise is to test your understanding of the model checking algorithm and to test your ability to manipulate the notions of transition systems, linear-time properties, Büchi automata and their product.

Consider the transition system TS of a mutual exclusion algorithm with only *one* process. (We take only one process in order to keep the exercise smaller.)



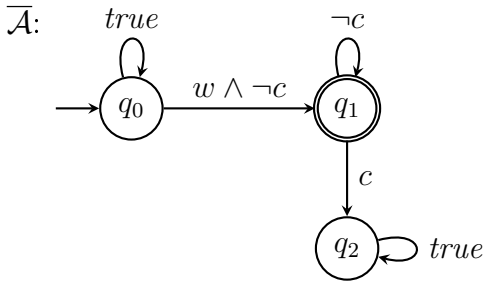
The state names n, w and c stand for "non-critical section", "waiting section" and "critical section", respectively. The set of atomic propositions is $AP = \{w, c\}$. The labeling function of the transition system labels the state n with $\{\}$, the state w with $\{w\}$ and the state c with $\{c\}$.
Let P_{live} be the following ω-regular property over $AP = \{w, c\}$:

> "Whenever the process is in its waiting location (w), it will eventually enter its critical section (c)."

(a) Depict a nonblocking NBA $\mathcal{A}$ for P_{live}.

(b) Check if $TS_{Sem} \not\models P_{live}$ holds by performing the following steps.

(i) Construct the reachable part of the product $TS \otimes \overline{\mathcal{A}}$. The nonblocking NBA $\overline{\mathcal{A}}$ for the complement property $\overline{P}_{live} = \left(2^{AP}\right)^{\omega} \setminus P_{live}$ is given below.



(ii) Check if this product, viewed as a Büchi automaton, has an accepting run (whether, viewed as a transition system, it satisfies the *persistence* property "eventually forever $\neg F$"). Remember, F is the set of accepting states of $\overline{\mathcal{A}}$.

In case the product satisfies the property, explain why this is the case. Otherwise, give an accepting run (a path in the transition system that shows the violation of the persistence property).

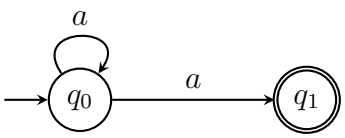
Name: _____ Matriculation Number: _____

Exercise 2 (Regular Properties II) **10**

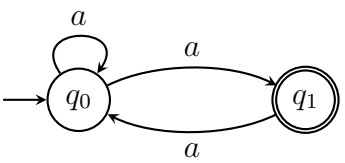
The goal of this exercise is to test your understanding of the role of determinism for Büchi automata and to test your ability to analyze the meaning of the acceptance condition of a Büchi automaton.

Consider the two NBA $\mathcal{A}_1$ and $\mathcal{A}_2$ depicted below.

$\mathcal{A}_1$



$\mathcal{A}_2$



(a) Give the deterministic automaton obtained by the determinisation of $\mathcal{A}_1$, viewed as NFA, and the deterministic automaton obtained by the determinisation of $\mathcal{A}_2$, viewed as NFA. Compare the two.

(b) Show that $\mathcal{L}_\omega(\mathcal{A}_1) \neq \mathcal{L}_\omega(\mathcal{A}_2)$.

Name: _____ Matriculation Number: _____

Exercise 3 (Regular Properties III) **10**

The goal of this exercise is to test your understanding of linear-time properties defined by regular sets of infinite words and to test your ability to construct a Büchi automaton.

Construct a Büchi automaton over the alphabet $\Sigma = \{a, b\}$ whose language consists of all ω-words that contain only finitely many b.

Name: _____ Matriculation Number: _____

Exercise 4 (Regular Properties IV) **10**

The goal of this exercise is to test your understanding of the relation between a safety property and a finite automaton over finite words, the relation which is used in the model checking for safety properties.

Let $AP = \{a, b, c\}$. Consider the following regular safety properties:

(P_1) If a becomes valid, afterward b stays valid ad infinitum or until c holds.

(P_2) Between two neighbouring occurrences of a, b always holds.

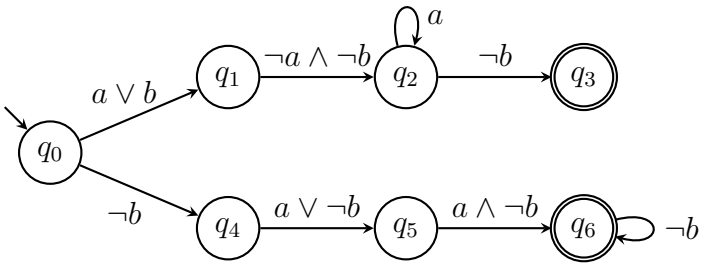
Construct an NFA $\mathcal{A}_i$ for each property P_i such that $\mathcal{L}(\mathcal{A}_i) = \text{BadPref}(P_i)$. You may use a symbolic NFA with propositional formulae over the set AP as transition labels.

Name: _____ Matriculation Number: _____

Exercise 5 (Regular Properties V) **10**

The goal of this exercise is to test your understanding of the notion of automata on finite prefixes of traces (the notion which is used in the model checking of safety properties) and test your ability to manipulate the notion of an alphabet where letters are sets of atomic propositions and the notion of symbolic automata where transitions are not labeled by letters but symbolic expressions.

Consider the following symbolic NFA $\mathcal{A}$ over the alphabet 2^{AP} with $AP = \{a, b\}$.



Which of the following words is accepted by $\mathcal{A}$? Give a short explanation.

(a) $w_1 = \{a\}\,\{\}\,\{a\}\,\{b\}$

(b) $w_2 = \{a\}\,\{\}\,\{a\}$

(c) $w_3 = \{b\}\,\{\}\,\{a, b\}\,\{a\}\,\{a\}$

(d) $w_4 = \{a\}\,\{a\}\,\{a\}\,\{\}\,\{a\}$

Name: _____ Matriculation Number: _____

Exercise 6 (Regular Properties VI) **10**

The goal of this exercise is to test your understanding of linear-time properties of concurrent systems and their formalization via regular expressions.

(a) For each of the following regular expressions over the alphabet $\Sigma = \{a, b\}$,

- describe informally what the regular expressions means, and
- find a different regular expression that describes the same language.

(i) $\emptyset a \emptyset$

(ii) $(a^* b^*)^*$

(iii) ε^{**}

(iv) a^{++}

(v) $a(ba)^* + (ab)^* a$

(b) Give regular expressions for the following languages over the alphabet $\Sigma = \{a, b\}$.

(i) $\mathcal{L}_1 = \{w \in \Sigma^* \mid \text{every } a \text{ in } w \text{ is immediately followed by } b\}$

(ii) $\mathcal{L}_2 = \{w \in \Sigma^* \mid w \text{ does not contain } bb\}$

(iii) $\mathcal{L}_3 = \{w \in \Sigma^* \mid w \text{ contains at least two } a\}$

(iv) $\mathcal{L}_4 = \{w \in \Sigma^* \mid w \text{ contains at most two } a\}$

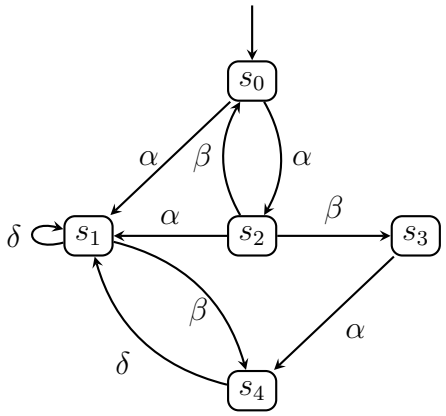
(v) $\mathcal{L}_5 = \{w \in \Sigma^* \mid w \text{ does not contain different characters}\}$

Name: _____ Matriculation Number: _____

Exercise 7 (Linear-Time Properties I) **10**

The goal of this exercise is to test your understanding of (the difference between the three notions of) fairness properties and to test your ability to manipulate the notions of fairness and realizability.

Consider the following transition system TS with actions $\{\alpha, \beta, \delta\}$ (and without atomic propositions):



Which of the following fairness assumptions $\mathcal{F}_i$ are realizable for TS? Explain why!

(a) $\mathcal{F}_1 = (\{\{\alpha\}\}, \{\{\delta\}\}, \{\{\alpha, \beta\}\})$

(b) $\mathcal{F}_2 = (\{\{\alpha, \delta\}\}, \{\{\alpha, \beta\}\}, \{\{\delta\}\})$

(c) $\mathcal{F}_3 = (\{\{\alpha, \delta\}, \{\beta\}\}, \{\{\alpha, \beta\}\}, \{\{\delta\}\})$

Recall the definition: Let TS be a transition system with set of actions Act. A fairness assumption $\mathcal{F}$ for Act is *realizable* for TS, if for every reachable state s it holds that $FairPaths_{\mathcal{F}}(s) \neq \emptyset$.

Name: _____ Matriculation Number: _____

Exercise 8 (Linear-Time Properties II) **10**

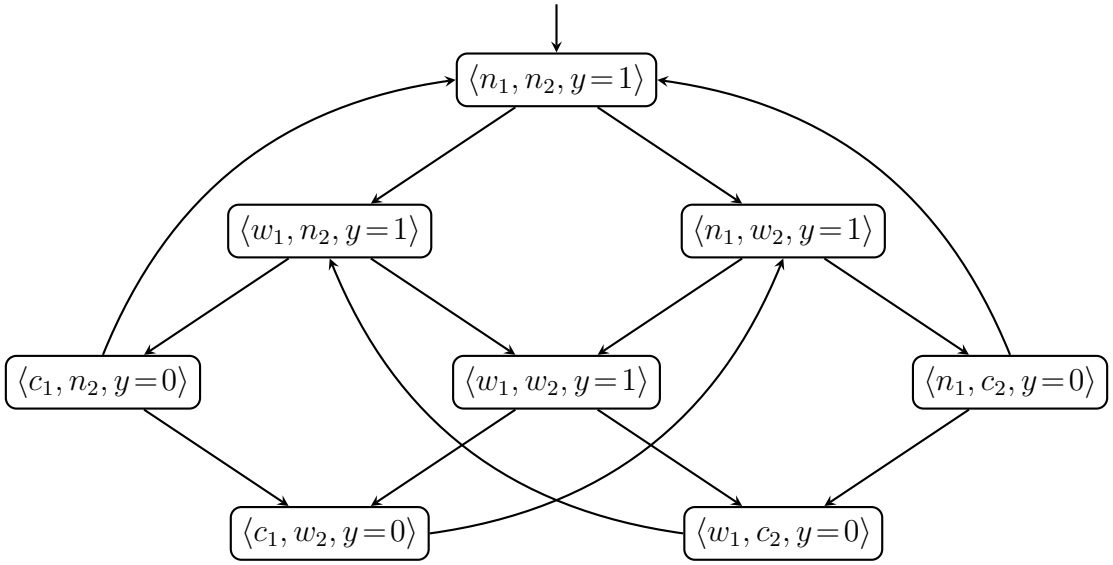
The goal of this exercise is to test your ability to manipulate the basic notions of (infinite!) traces as used in the definition of a liveness property.

Let TS_{Sem} and TS_{Pet} below be the transition systems for the semaphore-based mutual exclusion algorithm and Peterson's algorithm, respectively. Let $AP = \{w_1, c_1, w_2, c_2\}$. Prove or disprove:

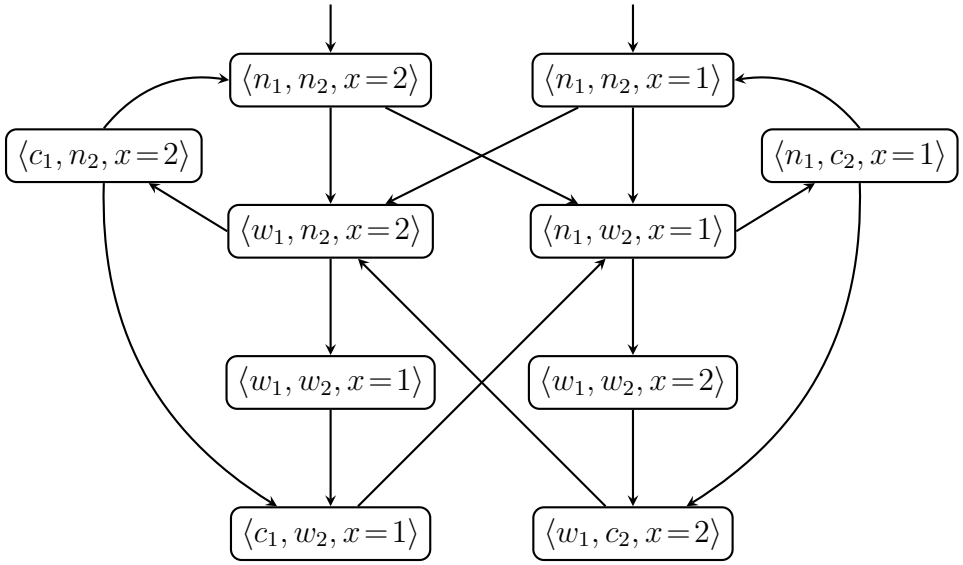
$$Traces(TS_{Sem}) = Traces(TS_{Pet}).$$

If the property does not hold, it is sufficient to give a trace of one transition system that is not a trace of the other transition system.

TS_{Sem}



TS_{Pet}



Name: _____ Matriculation Number: _____

Exercise 9 (Linear-Time Properties III) **10**

The goal of this exercise is to test your ability to manipulate the basic notions of sets, sets of traces, prefixes, etc. as used in the definition of a liveness property.

Let P and P' be liveness properties over AP. Prove or disprove the following claims:

(a) $P \cup P'$ is a liveness property.

(b) $P \cap P'$ is a liveness property.

Recall the definition:
An LT property P over AP is a liveness property iff $\mathit{pref}(P) = (2^{AP})^*$.

Name: _____ Matriculation Number: _____

Exercise 10 (Linear-Time Properties IV) **10**

The goal of this exercise is to test your ability to manipulate the basic notions of sets, sets of traces, prefixes, etc. as used in the definition of a safety property.

Prove that for the closure operator *closure* the following inclusion (resp. equality) holds for every linear-time property P.

(a) $P \subseteq \mathit{closure}(P)$

(b) $\mathit{closure}(P) = \mathit{closure}(\mathit{closure}(P))$ (the closure operator *closure* is idempotent)

Recall the definition: For an LT property P, the *closure* is defined by

$$\mathit{closure}(P) = \{\sigma \in (2^{AP})^{\omega} \mid \mathit{pref}(\sigma) \subseteq \mathit{pref}(P)\}$$

Name: _____ Matriculation Number: _____

Exercise 11 (Linear-Time Properties V) **10**

The goal of this exercise is to test your understanding of linear-time properties, and (the difference between) safety and liveness properties.

Consider the linear-time properties P_1 to P_7 below over the set of atomic propositions $AP = \{a, b\}$.

(a) For each of the properties, decide if the property is an invariant, a safety property, a liveness property, or neither. Explain why!

(b) For each property P that is neither a safety nor a liveness property, find a decomposition of $P = P_s \cap P_l$ into a safety and a liveness property. Use ω-regular expressions to describe P_s and P_l.

(P_1) Always (at any point of time) a or b holds.

(P_2) Always (at any point of time) a and b holds.

(P_3) Never b holds before a holds.

(P_4) Every time a holds there will be eventually a point of time where b holds.

(P_5) At exactly three points of time, a holds.

(P_6) If there are infinitely many points of time where a holds, then there are infinitely many points of time where b holds.

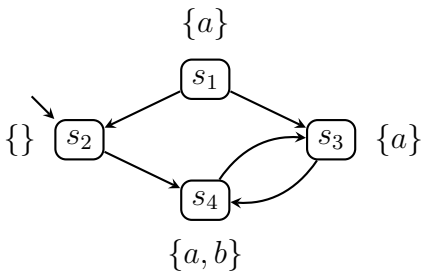
(P_7) There are only finitely many points of time where a holds.

Name: _____ Matriculation Number: _____

Exercise 12 (Linear-Time Properties VI) **10**

The goal of this exercise is to test your understanding of linear-time behavior, and (the difference between) paths and traces.

Consider the transition system below with the set of atomic propositions $AP = \{a, b\}$.



(a) Use regular expressions to describe the following sets:

(i) The set of initial finite path fragments of the transition system.

(ii) The set of paths of the transition system.

(iii) The set of initial finite traces of the transition system.

(iv) The set of traces of the transition system.

(b) Which of the four sets (i) to (iv) is finite and which is infinite?

Name: _____ Matriculation Number: _____

Exercise 13 (Modeling Concurrent Systems I) **10**

The goal of this exercise is to test your understanding of how to model communication protocols and their correctness properties, in the case of communication via channels.

Consider the following leader election algorithm:
For $n \in \mathbb{N}$, n processes $P_1, \ldots, P_n$ are located in a ring topology where each process is connected by an unidirectional channel to its neighbor in a clockwise manner. To distinguish the processes, each process is assigned a unique identifier id_i which is a natural number between 1 and n, i.e. $id_i \in \{1, \ldots, n\}$. Each process P_i has a local variable m_i.
The aim is to elect the process with the highest identifier as the leader within the ring. Each process executes the following algorithm:

Algorithm 1: Leader Election Algorithm for Process P_i

```
1  SEND (id_i);
2  while true do
3  |  RECEIVE (m_i);
4  |  if m_i = id_i then
5  |  |  STOP;
6  |  end
7  |  if m_i > id_i then
8  |  |  SEND (m_i);
9  |  end
10 end
```

(a) Model the leader election protocol for n processes as a channel system. Provide

(i) a graph that represents which processes communicate with each other, and

(ii) a graph that represents the behavior of each single process P_i (with $i \neq 1$, $i \neq n$) which has incoming channel $c_{i-1,i}$ and outgoing channel $c_{i,i+1}$. The instructions are of the form $c_{i,i+1}!m_i$ and $c_{i-1,i}?m_i$.

(b) Give an initial execution fragment of the system with three processes such that at least one process has executed the send statement within the body of the while loop. Assume that process P_i has identifier $id_i = i$ for $0 < i \leq 3$.

Name: _____ Matriculation Number: _____

Exercise 14 (Modeling Concurrent Systems II) **10**

The goal of this exercise is to test your understanding of shared memory, program graphs, concurrency, and interleaving.

We are given three (primitive) processes P_1, P_2, and P_3 with shared integer variable `x`. Process P_i executes ten times the assignment `x++`, which is realized using the three actions `LOAD(x)`, `INC(x)`, and `STORE(x)`. See the following pseudocode:

Algorithm 2: Process P_i

```
Data: x (global)
1 for i := 1 to 10 do
2 |   LOAD(x);
3 |   INC(x);
4 |   STORE(x);
5 end
```

Consider now the following parallel program P:

Algorithm 3: Parallel program P

```
Data: x (global)
1 x := 0;
2 P1‖P2‖P3;
```

(a) Does P have an execution that halts with the terminal value $\mathtt{x} = 2$? Explain!

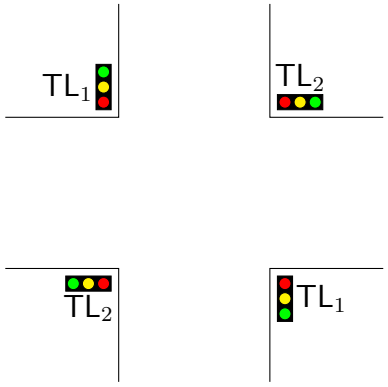
(b) Does P have an execution that halts with the terminal value $\mathtt{x} = 11$? Explain!

Name: _____ Matriculation Number: _____

Exercise 15 (Modeling Concurrent Systems III) **10**

The goal of this exercise is to test your understanding of how to model communication protocols and their correctness properties, in the case of communication via handshaking.

Consider the crossing of two roads with four traffic lights as depicted on the right. The two traffic lights labelled with TL_1 always show the same color, and likewise the two traffic lights labelled with TL_2 always show the same color. The traffic lights have three modes: `red`, `yellow`, and `green`, and they switch from `green` to `yellow`, from `yellow` to `red`, and from `red` to `green`.



(a) Create two transition systems TS_1 and TS_2 for the traffic lights, one for each direction of a crossing.

Insert suitable actions on which these systems can synchronize so that at least one of the lights is in the `red` mode in each state of the transition system $TS_1 \| TS_2$.

(b) Construct the transition system $TS_1 \| TS_2$. It is sufficient to draw the reachable part.

(c) Is the system safe? An informal argument is sufficient.

Name: _____ Matriculation Number: _____

Exercise 16 (Modeling Concurrent Systems IV) **10**

The goal of this exercise is to test your understanding of how to model communication protocols and their correctness properties, in the case of communication with shared variables.

Consider the following variant of Peterson's mutual exclusion algorithm, where the assignments to b_i and x are done in separate atomic steps, and x is assigned before b_i.

```
          while true {
              ......
req :     x := 2;
nc :      b1 := true;
wt :      wait until(x = 1 ∨ ¬b2);
cs :      ... critical section ...
          b1 := false;
              ......
          }
```

```
          while true {
              ......
req :     x := 1;
nc :      b2 := true;
wt :      wait until(x = 2 ∨ ¬b1);
cs :      ... critical section ...
          b2 := false;
              ......
          }
```

We say that the communication protocol *satisfies the mutual exclusion property* if there is no execution such that both processes are in the critical section at the same time.

Check if the mutual exclusion property is satisfied. You do not have to draw the program graphs or transition systems for the interleaving. Instead, if the variant satisfies the property, explain why. If it violates the property it is sufficient to give an execution that shows the violation.
Assume that initially b_1 and b_2 have the value `false`.

Name: _____ Matriculation Number: _____

Exercise 17 (Modeling Concurrent Systems V) **10**

The goal of this exercise is to test your understanding of how to model shared memory systems, interleaving of program graphs, concurrency, and interleaving of executions.

The two examples of parallel systems below seem to do the same. In both systems, the initial value of x is 1. Each line of the pseudocodes reflects an atomic statement.

(a) The first system is given in a high level language.

```
/* PROCESS 1 */        /* PROCESS 2 */
1: x := x+3;           1: x := 5*x;
```

(i) Draw the program graphs PG_1 and PG_2 for each process with $Var_1 = Var_2 = \{x\}$.

(ii) Draw the interleaving of the program graphs $PG_1 ||| PG_2$.

(iii) Which values can be finally stored in x?

(b) The second system is given in assembler.

```
/* PROCESS 1 */        /* PROCESS 2 */
1: LOAD x;             1: LOAD x;
2: ADDI 3;             2: MULTI 5;
3: STORE x;            3: STORE x;
```

Semantics of the assembler commands:

- `LOAD i` : Load the content of address i of the memory into an accumulator register ACC.
- `ADDI i` : Adds i to the content of register ACC and stores the result in ACC.
- `MULTI i` : Multiplies the content of register ACC with i and stores the result in ACC.
- `STORE i` : Store the content of register ACC at memory address i.

Here, the program graphs will be too big. You do not have to draw the program graphs.

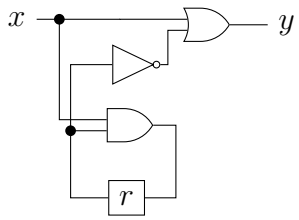
Which values can be finally stored in x?

Name: _____ Matriculation Number: _____

Exercise 18 (Modeling Concurrent Systems VI) **10**

The goal of this exercise is to test your understanding of how to model concurrent systems, in the case of hardware systems.

Consider the following sequential hardware circuit.



Provide the transition system of this hardware circuit. The states are the evaluations of the input x and the register r. The transitions represent the stepwise behavior of the circuit. The values of the input x change nondeterministically. The atomic propositions $\{X, Y, R\}$ stand for $x = 1$, $y = 1$ and $r = 1$, respectively. Initially the register r has the value 1 (`true`).

For your reference:  = AND gate, = OR gate,  = NOT gate