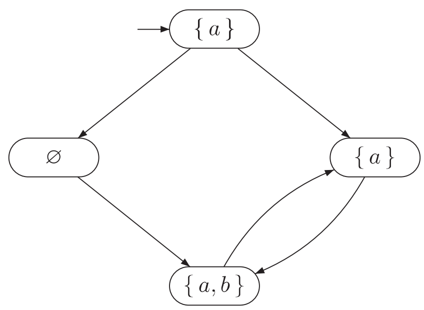
in the monograph by Francez [155]. A recent characterization of fairness in terms of topology, language theory, and game theory has been provided by Völzer, Varacca, and Kindler [415].

3.8 Exercises

Exercise 3.1. Give the traces on the set of atomic propositions $\{ a, b \}$ of the following transition system:



Exercise 3.2. On page 97, a transformation is described of a transition system TS with possible terminal states into an "equivalent" transition system TS^* without terminal states. Questions:

(a) Give a formal definition of this transformation $TS \mapsto TS^*$

(b) Prove that the transformation preserves trace-equivalence, i.e., show that if TS_1, TS_2 are transition systems (possibly with terminal states) such that $Traces(TS_1) = Traces(TS_2)$, then $Traces(TS_1^*) = Traces(TS_2^*)$.[8]

Exercise 3.3. Give an algorithm (in pseudocode) for invariant checking such that in case the invariant is refuted, a *minimal* counterexample, i.e., a counterexample of minimal length, is provided as an error indication.

Exercise 3.4. Recall the definition of AP-deterministic transition systems (Definition 2.5 on page 24). Let TS and TS' be transition systems with the same set of atomic propositions AP. Prove the following relationship between trace inclusion and finite trace inclusion:

(a) For AP-deterministic TS and TS':

$$Traces(TS) = Traces(TS') \text{ if and only if } Traces_{fin}(TS) = Traces_{fin}(TS').$$

[8]If TS is a transition system with terminal states, then $Traces(TS)$ is defined as the set of all words $trace(\pi)$ where π is an initial, maximal path fragment in TS.

(b) Give concrete examples of TS and TS' where at least one of the transition systems is not AP-deterministic, but

$$Traces(TS) \not\subseteq Traces(TS') \quad \text{and} \quad Traces_{fin}(TS) = Traces_{fin}(TS').$$

EXERCISE 3.5. Consider the set AP of atomic propositions defined by $AP = \{ x = 0, x > 1 \}$ and consider a nonterminating sequential computer program P that manipulates the variable x. Formulate the following informally stated properties as LT properties:

(a) false

(b) initially x is equal to zero

(c) initially x differs from zero

(d) initially x is equal to zero, but at some point x exceeds one

(e) x exceeds one only finitely many times

(f) x exceeds one infinitely often

(g) the value of x alternates between zero and two

(h) true

(This exercise has been adopted from [355].) Determine which of the provided LT properties are safety properties. Justify your answers.

EXERCISE 3.6. Consider the set $AP = \{ A, B \}$ of atomic propositions. Formulate the following properties as LT properties and characterize each of them as being either an invariance, safety property, or liveness property, or none of these.

(a) A should never occur,

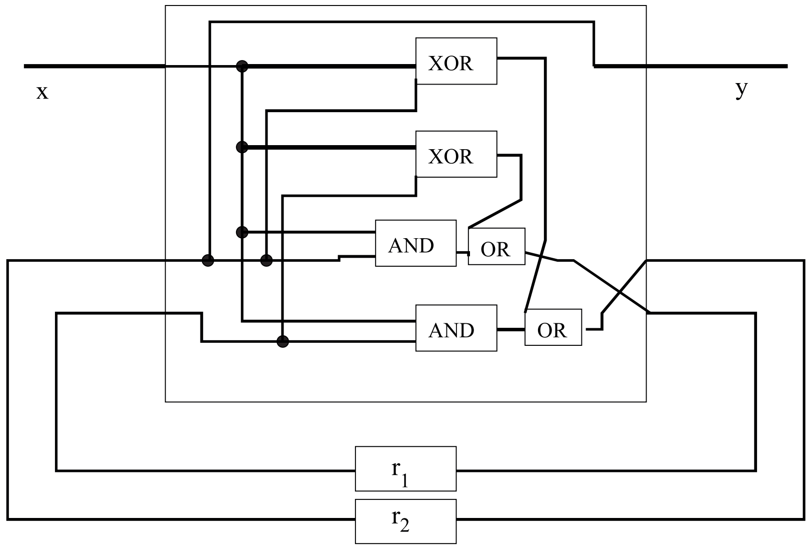
(b) A should occur exactly once,

(c) A and B alternate infinitely often,

(d) A should eventually be followed by B.

(This exercise has been inspired by [312].)

EXERCISE 3.7. Consider the following sequential hardware circuit:



The circuit has input variable x, output variable y, and registers r_1 and r_2 with initial values $r_1 = 0$ and $r_2 = 1$. The set AP of atomic propositions equals $\{ x, r_1, r_2, y \}$. Besides, consider the following informally formulated LT properties over AP:

P_1 : Whenever the input x is continuously high (i.e., x=1), then the output y is infinitely often high.

P_2 : Whenever currently r_2=0, then it will never be the case that after the next input, r_1=1.

P_3 : It is never the case that two successive outputs are high.

P_4 : The configuration with x=1 and r_1=0 never occurs.

Questions:

(a) Give for each of these properties an example of an infinite word that belongs to P_i. Do the same for the property $\left(2^{AP}\right)^\omega \setminus P_i$, i.e., the complement of P_i.

(b) Determine which properties are satisfied by the hardware circuit that is given above.

(c) Determine which of the properties are safety properties. Indicate which properties are invariants.

  (i) For each safety property P_i, determine the (regular) language of bad prefixes.

  (ii) For each invariant, provide the propositional logic formula that specifies the property that should be fulfilled by each state.

Exercise 3.8. Let LT properties P and P' be equivalent, notation $P \cong P'$, if and only if $\mathit{pref}(P) = \mathit{pref}(P')$. Prove or disprove: $P \cong P'$ if and only if $\mathit{closure}(P) = \mathit{closure}(P')$.

Exercise 3.9. Show that for any transition system *TS*, the set *closure*(*Traces*(*TS*)) is a safety property such that $TS \models closure(Traces(TS))$.

Exercise 3.10. Let P be an LT property. Prove: $pref(closure(P)) = pref(P)$.

Exercise 3.11. 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.

Answer the same question for P and P' being safety properties.

Exercise 3.12. Prove Lemma 3.38 on page 125.

Exercise 3.13. Let $AP = \{ a, b \}$ and let P be the LT property of all infinite words $\sigma = A_0 A_1 A_2 \ldots \in \left(2^{AP}\right)^{\omega}$ such that there exists $n \geqslant 0$ with $a \in A_i$ for $0 \leqslant i < n$, $\{ a, b \} = A_n$ and $b \in A_j$ for infinitely many $j \geqslant 0$. Provide a decomposition $P = P_{safe} \cap P_{live}$ into a safety and a liveness property.

Exercise 3.14. Let TS_{Sem} and TS_{Pet} be the transition systems for the semaphore-based mutual exclusion algorithm (Example 2.24 on page 43) and Peterson's algorithm (Example 2.25 on page 45), respectively. Let $AP = \{ wait_i, crit_i \mid i = 1, 2 \}$. Prove or disprove:

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

If the property does not hold, provide an example trace of one transition system that is not a trace of the other one.

Exercise 3.15. Consider the transition system *TS* outlined on the right and the sets of actions $B_1 = \{ \alpha \}$, $B_2 = \{ \alpha, \beta \}$, and $B_3 = \{ \beta \}$. Further, let E_b, E_a and E' be the following LT properties: