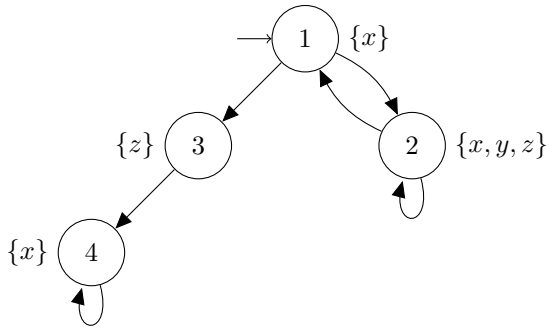
Checking CTL, and NuSMV excercise

No homework is mandatory this week.

Model checking CTL

Consider the following transition system:



Also consider the CTL-specification:
$\forall\Box(\exists\bigcirc z \vee \neg\exists\Diamond y)$

Show the steps of the CTL model checking algorithm for the given transition system and specification. Do this by giving the satisfaction set of every sub-formula, starting with the smallest sub-formulas, and finally for the formula itself. From the satisfaction sets, explain whether the formula holds, and why.

Pnueli's mutual exclusion protocol

We consider two processes, which run (infinitely long) in an interleaved way. The processes cannot both make a step at the same time. Each process has a noncritical and a critical section, which it enters alternatingly. The critical section can only be entered by one process at a time, and one process may therefore have to wait for the other process. This can safely be done by implementing both processes according to Pnueli's algorithm. Use a shared boolean variable s, and for both processes i (with $i \in \{0, 1\}$ a local variable y_i. Then each process i behaves as follows:

1: **loop**
2: noncritical section
3: $(y_i, s) \leftarrow (1, i)$
4: **while** $\neg(y_{1-i} = 0 \vee s \neq i)$ **do**
5: wait
6: **end while**
7: critical section
8: $y_i \leftarrow 0$
9: **end loop**

Note that line 3 contains an atomic assignment, i.e. a process assigns both variables at the same time, without the other process interrupting in between.

a) Create a NuSMV-model for these two processes. Modules may be useful.

b) Formalize the property of mutual exclusion, and check it.

c) Formalize the absence of starvation, i.e. that both processes will get to their critical section, and check it. If it does not hold, why not, and can you define a notion of fairness such that it does hold?

d) The atomic step in which two variables are assigned at once is often unrealistic. The assignment can be split in two ways, for both processes (first assign y_{1-i} or first assign s): check whether every combination of splitting still guarantees mutual exclusion.