ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Engineering and Networks Laboratory

HS 2010 Dr. K. Lampka

Discrete Event Systems

Solution to Exercise Sheet 14

1 Computation Tree Logic Model Checking

a) Ω_1 und Ω_6 are already in ENF. The remaining formulas are transformed as follows.

$$\Omega_2 \equiv \neg\exists(\text{true} \bigcup \neg\text{yellow})$$

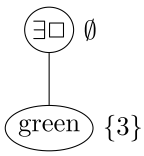
$$\Omega_3 \equiv \forall(\text{true} \bigcup \text{black}) \equiv \neg\exists(\neg\text{black} \bigcup \neg\text{true}) \wedge \neg\exists\Box\neg\text{black}$$

$$\Omega_4 \equiv \neg\exists(\neg\text{black} \bigcup \neg\text{black}) \wedge \neg\exists\Box\neg\text{black}$$

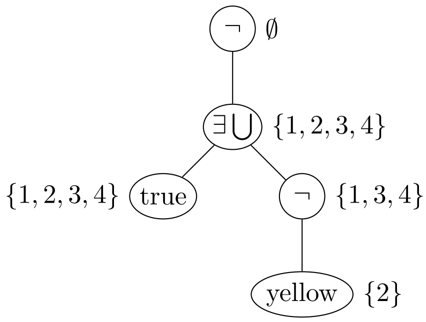
$$\Omega_5 \equiv \neg\exists\left(\neg(\exists \bigcirc \text{black}) \bigcup (\text{yellow} \wedge \neg(\exists \bigcirc \text{black}))\right) \wedge \neg\exists\Box\neg(\exists \bigcirc \text{black})$$

b) We first give the annotated syntax trees for the syntactically correct formulas. The calculation of the satisfiability sets is explained examplarily in **f)** for Ω_5. For the remaining formulas, the calculation works analogously.

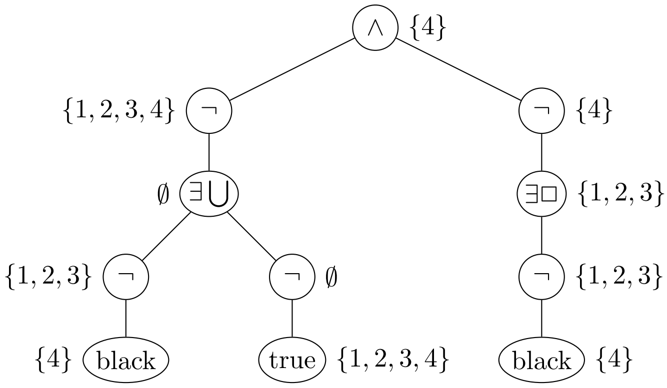
- $\Omega_1 = \exists\Box\text{green}$:



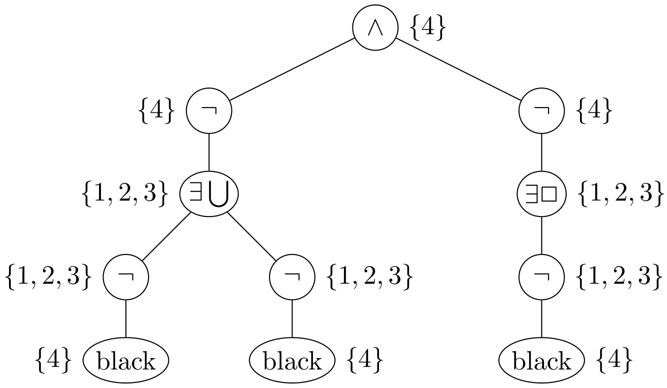
- $\Omega_2 = \neg\exists(\text{true} \bigcup \neg\text{yellow})$:



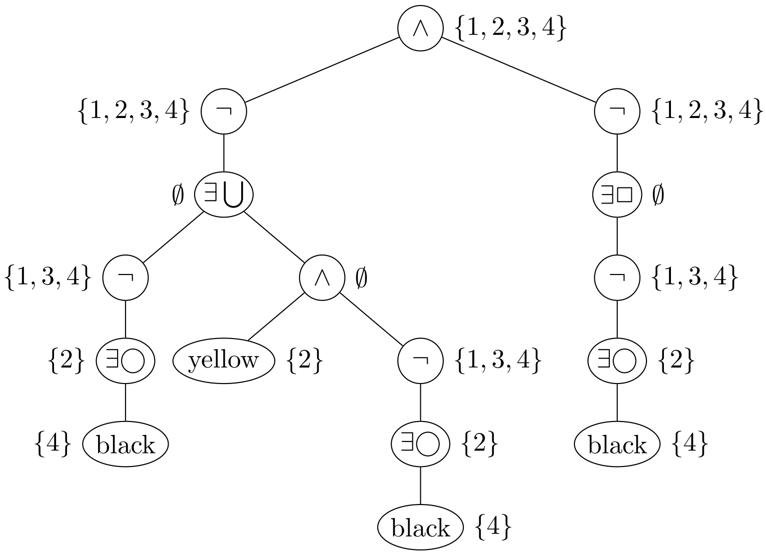
- $\Omega_3 = \neg\exists(\neg\text{black} \bigcup \neg\text{true}) \wedge \neg\exists\Box\neg\text{black}$:



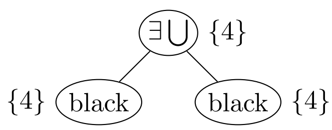
- $\Omega_4 = \neg\exists(\neg\text{black} \bigcup \neg\text{black}) \wedge \neg\exists\Box\neg\text{black}$:



- $\Omega_5 = \neg\exists\Big(\neg(\exists\bigcirc\text{black}) \bigcup \big(\text{yellow} \wedge \neg(\exists\bigcirc\text{black})\big)\Big) \wedge \neg\exists\Box\neg(\exists\bigcirc\text{black})$:



- $\Omega_6 = \exists(\text{black} \bigcup \text{black})$:



- We explain the calculation of the satisfiability sets Satisfy(ϕ) exemplarily for Ω_5. The Satisfy(ϕ)-sets are first determined for the leaves of the syntax tree and then the nodes of the tree are labelled in a bottom-up fashion.

- Satisfy(black) $= \{\vec{s} \in \{1,2,3,4\} \mid L(\vec{s}) = \text{black}\} = \{4\}$
- Satisfy(yellow) $= \{\vec{s} \in \{1,2,3,4\} \mid L(\vec{s}) = \text{yellow}\} = \{2\}$
- Satisfy($\exists \bigcirc$ black) $= \{\vec{s} \in \{1,2,3,4\} \mid \text{Post}(\vec{s}) \cap \text{Satisfy(black)} \neq \emptyset\} = \{2\}$
- Satisfy($\neg\exists \bigcirc$ black) $= \mathbb{S} \setminus$ Satisfy($\exists \bigcirc$ black) $= \{1,3,4\}$
- Satisfy(yellow $\wedge \neg\exists \bigcirc$ black) = Satisfy(yellow) $\cap$ Satisfy($\neg\exists \bigcirc$ black) $= \emptyset$
- Satisfy($\exists\Box(\neg\exists \bigcirc$ black)) =: Satisfy(ϕ):
  Here we have to do a fixed point calculation. We calculate the sets $\text{Satisfy}_i(\phi)$ until we reach a fixed point where we have $\text{Satisfy}_j(\phi) = \text{Satisfy}_i(\phi)$ for all $j > i$.
  * $\text{Satisfy}_0(\phi) = \text{Satisfy}(\neg\exists \bigcirc \text{black}) = \{1,3,4\}$
  * $\text{Satisfy}_1(\phi) = \{\vec{s} \in \{1,3,4\} \mid \text{Post}(\vec{s}) \cap \{1,3,4\} \neq \emptyset\} = \{1\}$
  * $\text{Satisfy}_2(\phi) = \{\vec{s} \in \{1\} \mid \text{Post}(\vec{s}) \cap \{1\} \neq \emptyset\} = \emptyset$

  Clearly, we have arrived at a fixed point since the set $\text{Satisfy}_3(\phi)$ (as well as all other sets $\text{Satisfy}_i(\phi)$ for $i > 3$) cannot hold more elements than $\text{Satisfy}_2(\phi)$. Hence, we have Satisfy($\exists\Box(\neg\exists \bigcirc$ black)) $= \emptyset$.
- Satisfy($\exists(\neg\exists \bigcirc$ black) $\bigcup$(yellow $\wedge \neg\exists \bigcirc$ black)) =: Satisfy(ϕ):
  Again, we have to do a fixed point calculation.
  * $\text{Satisfy}_0(\phi) = \text{Satisfy}(\text{yellow} \wedge \neg\exists \bigcirc \text{black}) = \emptyset$
  * $\text{Satisfy}_1(\phi) = \emptyset \cup \{\vec{s} \in \{1,3,4\} \mid \mathcal{P}ost(\vec{s}) \cap \emptyset \neq \emptyset\} = \emptyset$

  We have arrived at a fixed point because all subsequent sets $\text{Satisfy}_i(\phi)$ evaluate to the empty set. Hence we have Satisfy($\exists(\neg\exists\bigcirc$ black) $\bigcup$(yellow $\wedge\neg\exists\bigcirc$ black)) $= \emptyset$.
- Satisfy($\neg\exists(\neg\exists \bigcirc$ black) $\bigcup$(yellow $\wedge \neg\exists \bigcirc$ black)) $= \mathbb{S} \setminus \emptyset = \{1,2,3,4\}$
- Satisfy($\neg\exists\Box(\neg\exists \bigcirc$ black)) $= \mathbb{S} \setminus \emptyset = \{1,2,3,4\}$
- Satisfy($\neg\exists(\neg\exists\bigcirc$black) $\bigcup$(yellow$\wedge\neg\exists\bigcirc$black)$\wedge\neg\exists\Box(\neg\exists\bigcirc$black)) $= \{1,2,3,4\} \cap \{1,2,3,4\} = \{1,2,3,4\}$.

c) A formula Ω is satisfied by $\mathcal{K}$ if the intersection of $\mathbb{S}_0$ and Satisfy($\mathcal{K}$) at the root of the syntax tree is not empty. This is only the case for Ω_5. All other formulas are not satisfiable in $\mathcal{K}$.

d) We shall have a look at the computation tree.

A path Π_1 starting in state 1 falsifies Ω_2 and Ω_4 because already in state 1 we have $\neg$yellow as well as $\neg$black. The path $\Pi_2 = (1,3,2,1,3,2,\ldots)$ falsifies Ω_3 because this path never reaches state 4 and hence never fulfils black.

2 Timed Automata

Solutions can be found in the reference material for this exercise as given on the course website or at `http://disco.ethz.ch/lectures/hs10/des/exercises/UppaalSolutions.zip`.