Basics of Verification

Written exam, November 16th, 2009

3 hours

The lecture notes are authorized, but no other documents.
Exercises are independent.
All answers should be rigorously and clearly justified.
The number in front of each question gives an indication on its length or difficulty.

1 LTL

We fix a set AP of atomic propositions containing $\{p, q\}$ and its associated alphabet $\Sigma = 2^{\mathrm{AP}}$.

[1] **a)** Consider the formulae $\varphi_1 = \mathsf{G}(p \to q)$ and $\varphi_2 = \mathsf{G}(p \to ((\neg p)\,\mathsf{R}\,q))$.
Prove or disprove that φ_1 implies φ_2, i.e., that $\varphi_1 \to \varphi_2$ is valid.
Prove or disprove that φ_2 implies φ_1.

[5] **b)** Consider the formulae $\alpha = p\,\mathsf{U}\,q$, $\beta = q\,\mathsf{U}\,p$, $\psi = (p \to \alpha) \wedge (q \to \beta)$ and $\varphi = \mathsf{G}\,\psi$. The aim is to compute the generalized Büchi automaton (GBA) $\mathcal{A}_\varphi$ associated with φ with the construction seen during the course. The following intermediary steps are mandatory:

(i) Write the formula φ in negative normal form.

(ii) Draw the reduction graph starting from $\{\varphi\}$. Do not forget the marks $!\alpha$ and $!\beta$ on the reduction rules. You should use the *additional* reduction rules in order to get a simpler graph.

(iii) Give the sets $\mathrm{Red}(\{\varphi\})$, $\mathrm{Red}_\alpha(\{\varphi\})$ and $\mathrm{Red}_\beta(\{\varphi\})$.

(iv) Draw the transitions starting from state $\{\varphi\}$ in the GBA $\mathcal{A}_\varphi$.

(v) Complete the construction and draw the automaton $\mathcal{A}_\varphi$.
Indicate clearly the acceptance sets T_α and T_β.

[1] **c)** Show that the following sets are expressible in $\mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$: $\Sigma_p^* \cdot \Sigma_{\neg p}^\omega$ and $\Sigma_p^n \cdot \Sigma_{\neg p}^\omega$ for all $n \geq 0$.

[4] **d)** Show that the set $(\Sigma_p^2)^* \cdot \Sigma_{\neg p}^\omega$ is not expressible in $\mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$.
Hint: Define a sequence of words $(w_n)_{n\geq 0}$ with $w_n \in \Sigma_p^{n+2} \cdot \Sigma_{\neg p}^\omega$ and verifying:
for all $\varphi \in \mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$ and all $n \geq 0$, if φ has at most n occurrences of X then
$$w_n, 0 \models \varphi \text{ if and only if } w_n, 1 \models \varphi.$$

[1] **e)** Prove or disprove that the language $(\Sigma_p \cdot \Sigma_{\neg p})^\omega$ is expressible in $\mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$.

[3] **f)** Prove or disprove that the language $(\Sigma_p \cdot \Sigma)^\omega$ is expressible in $\mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$.

2 CTL and CTL*

[3] **a)** For $n \geq 1$ we define the CTL* formula $\varphi_n = \mathsf{A}(\mathsf{X}^n\, p \vee \mathsf{F}\, q)$.
Show that φ_1 is equivalent to the CTL formula $\psi_1 = q \vee \mathsf{AX}(p \vee \mathsf{AF}\, q)$.
Prove that for all $n \geq 1$, φ_n is equivalent to some CTL formula ψ_n that you should define.

[2] **b)** Prove or disprove that the CTL* formula $\mathsf{A}(\mathsf{X}\, p \vee (q\, \mathsf{U}\, r))$ can be expressed in CTL.

[5] **c)** Prove or disprove that the CTL* formula $\mathsf{A}((p\, \mathsf{U}\, q) \vee (p'\, \mathsf{U}\, q'))$ can be expressed in CTL.
Advice: Keep this question for the end, if you have some time left.

3 Specifications and CTL

The aim is to produce formal specifications for a lift system. The system consists of n floors numbered from 1 to n. The cabin has one door which may be open or closed.

The formal specifications should use the following boolean variables:

- $\texttt{at}_i$ which is true if the cabin is *at* floor i ($1 \leq i \leq n$),
- `closed` which is true if the door of the cabin is *closed*,
- `up` which is true if the last direction of travel was *up*, and false otherwise,
- $\texttt{call}_i$ which is true if there is a request for floor i ($1 \leq i \leq n$).
  A call for some floor i is satisfied when the cabin is at floor i with its door open.

[6] **a)** Give formal specifications in CTL for the following:

(i) Whenever the cabin is moving between consecutive floors, the door should be and should stay closed.

(ii) The lift system stays *idle* if and only if there are no pending calls. By *idle* we mean that the cabin stays at the same floor and its door stays either open or closed.

(iii) The cabin never moves in a direction toward which there are no pending calls.

(iv) The cabin travels in its current direction satisfying all calls until no more exist in the current direction.

[6] **b)** Assuming that the lift system satisfies the above specifications, prove or disprove that every call will be eventually satisfied.

We assume now a new boolean variable $\frac{2}{3}$-`full` which is true if the cabin has reached two-thirds of its capacity. We also assume that $\texttt{call}_i$ is refined in two boolean variables $\texttt{cabinCall}_i$ and $\texttt{landingCall}_i$ which are true respectively if a call for floor i was placed inside the cabin or on the floor: $\texttt{call}_i = \texttt{cabinCall}_i \vee \texttt{landingCall}_i$.

[3] **c)** Give formal specifications for the following policies:

(i) When the cabin is $\frac{2}{3}$-full then the lift system answers cabin calls as before but it does not answer landing calls anymore.

(ii) Give priority to calls from or to floor n (assumed to be the *executive* floor).

(iii) Combination of the above two policies.

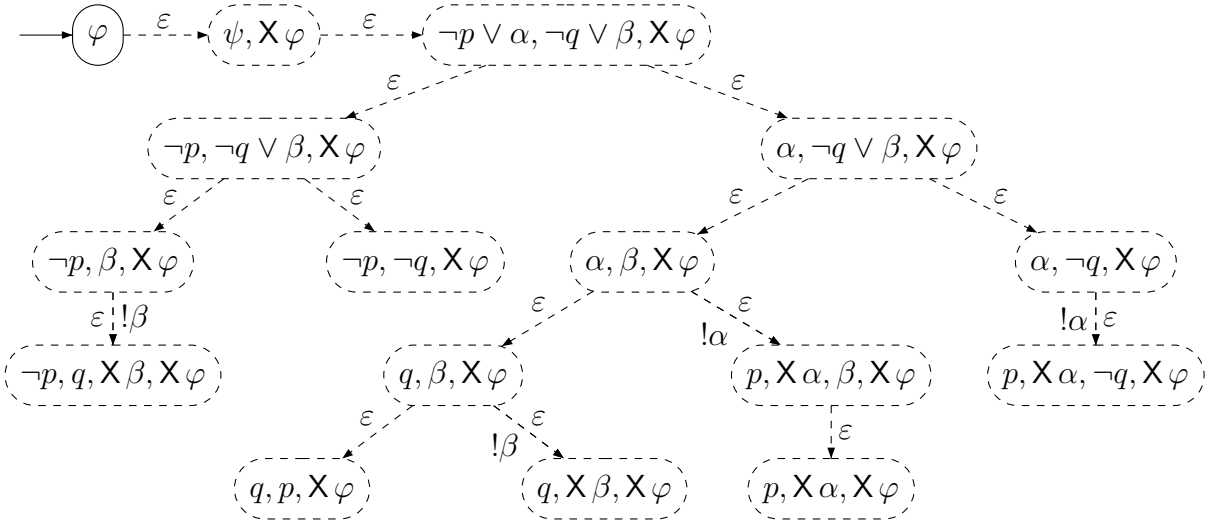
Solution of Exercise 1

[1] **a)** φ_1 does not imply φ_2: Consider $w = \{p,q\}\emptyset^\omega$. We have $w,0 \models \varphi_1$ but $w,0 \not\models \varphi_2$. Indeed, $w,0 \models p$ but $w,0 \not\models (\neg p) \wedge q$ and $w,1 \not\models q$. Hence, $w,0 \not\models (\neg p)\,\mathsf{R}\,q$.

φ_2 implies φ_1: Indeed, $(\neg p)\,\mathsf{R}\,q = q \wedge (\neg p \vee \mathsf{X}((\neg p)\,\mathsf{R}\,q))$ which implies q.

[5] **b)** (i) $\varphi = \mathsf{G}((\neg p \vee (p\,\mathsf{U}\,q)) \wedge (\neg q \vee (q\,\mathsf{U}\,p)))$

(ii)



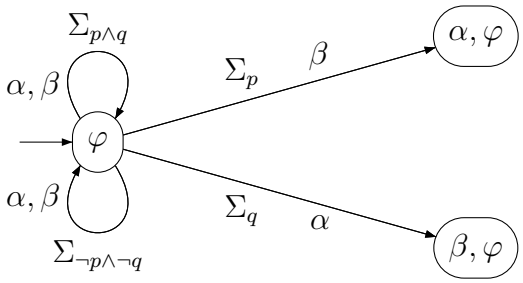
(iii)

$\mathrm{Red}(\{\varphi\}) = \{\{\neg p, \neg q, \mathsf{X}\varphi\}, \{p, q, \mathsf{X}\varphi\}, \{p, \neg q, \mathsf{X}\alpha, \mathsf{X}\varphi\}, \{p, \mathsf{X}\alpha, \mathsf{X}\varphi\}\{\neg p, q, \mathsf{X}\beta, \mathsf{X}\varphi\}, \{q, \mathsf{X}\beta, \mathsf{X}\varphi\}\}$

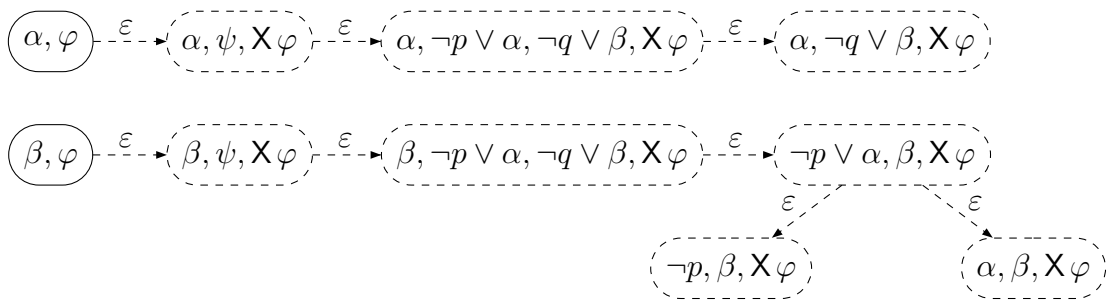
$\mathrm{Red}_\alpha(\{\varphi\}) = \{\{\neg p, \neg q, \mathsf{X}\varphi\}, \{p, q, \mathsf{X}\varphi\}, \{\neg p, q, \mathsf{X}\beta, \mathsf{X}\varphi\}, \{q, \mathsf{X}\beta, \mathsf{X}\varphi\}\}$

$\mathrm{Red}_\beta(\{\varphi\}) = \{\{\neg p, \neg q, \mathsf{X}\varphi\}, \{p, q, \mathsf{X}\varphi\}, \{p, \neg q, \mathsf{X}\alpha, \mathsf{X}\varphi\}, \{p, \mathsf{X}\alpha, \mathsf{X}\varphi\}\}$

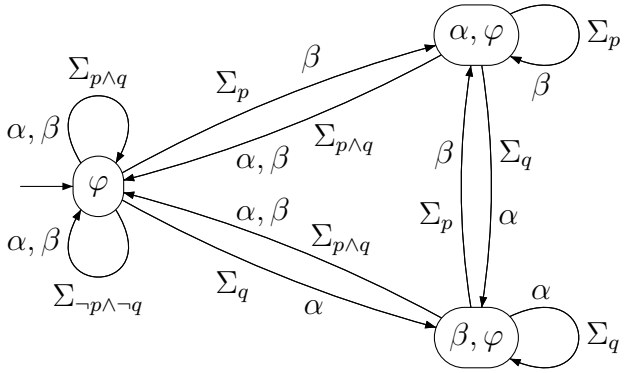
(iv)



(v) The reduction from the two new states adds the following part to the reduction graph computed in (ii):



The final GBA is:



[1] **c)** The language $\Sigma_p^* \cdot \Sigma_{\neg p}^\omega$ is expressed by $p \,\mathsf{U}\, (\mathsf{G}\, \neg p)$.

For $n \geq 0$, the language $\Sigma_p^n \cdot \Sigma_{\neg p}^\omega$ is expressed by $\mathsf{X}^n(\mathsf{G}\, \neg p) \wedge \bigwedge_{0 \leq i < n} \mathsf{X}^i\, p$.

[4] **d)** For $n \geq 0$, let $w_n = \{p\}^{n+2}\emptyset^\omega \in \Sigma_p^{n+2} \cdot \Sigma_{\neg p}^\omega$.
We prove by induction *on the formula* that for all $\varphi \in \mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$ and all $n \geq 0$, if φ has at most n occurrences of X then

$$w_n, 0 \models \varphi \text{ if and only if } w_n, 1 \models \varphi.$$

- If $\varphi \in \mathrm{AP}$ then for all $n \geq 0$ we have $w_n, 0 \models \varphi$ iff $\varphi \in \{p\}$ iff $w_n, 1 \models \varphi$.
- Negation and disjunction are trivial since we have an equivalence.
- If $\varphi = \mathsf{X}\, \varphi_1$ has at most n occurrences of X then
  $w_n, 0 \models \varphi$ iff $w_n, 1 \models \varphi_1$ iff $w_{n-1}, 0 \models \varphi_1$ iff (by induction) $w_{n-1}, 1 \models \varphi_1$ iff $w_n, 1 \models \varphi$.
- If $\varphi = \varphi_1 \,\mathsf{U}\, \varphi_2$ has at most n occurrences of X then
  $w_n, 0 \models \varphi$ iff $w_n, 0 \models \varphi_2$ or ($w_n, 0 \models \varphi_1$ and $w_n, 1 \models \varphi$) iff (by induction)
  $w_n, 1 \models \varphi_2$ or ($w_n, 1 \models \varphi_1$ and $w_n, 1 \models \varphi$) iff $w_n, 1 \models \varphi$.

Assume that the set $(\Sigma_p^2)^* \cdot \Sigma_{\neg p}^\omega$ can be expressed by a formula $\varphi \in \mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$.
Let $n \geq 2$ be an even upper bound on the number of occurrences of X in φ.
We have $w_n \in (\Sigma_p^2)^* \cdot \Sigma_{\neg p}^\omega$ hence $w_n, 0 \models \varphi$.
From the lemma above we deduce $w_n, 1 \models \varphi$ and $w_{n-1}, 0 \models \varphi$.
This is a contradiction since $w_{n-1} \notin (\Sigma_p^2)^* \cdot \Sigma_{\neg p}^\omega$.

[1] **e)** The language $(\Sigma_p \cdot \Sigma_{\neg p})^\omega$ is expressed by $p \wedge \mathsf{G}(p \rightarrow \mathsf{X}\, \neg p) \wedge \mathsf{G}(\neg p \rightarrow \mathsf{X}\, p)$.

[3] **f)** For $n \geq 0$, let $w_n = \{p\}^{n+2}(\{p\}\emptyset)^\omega \in \Sigma^\omega$.
Exactly as for **d)** we can prove that for all $\varphi \in \mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$ and all $n \geq 0$, if φ has at most n occurrences of X then

$$w_n, 0 \models \varphi \text{ if and only if } w_n, 1 \models \varphi.$$

Assume that the set $(\Sigma_p \cdot \Sigma)^\omega$ can be expressed by a formula $\varphi \in \mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$.
Let $n \geq 2$ be an even upper bound on the number of occurrences of X in φ.
We have $w_n \in (\Sigma_p \cdot \Sigma)^\omega$ hence $w_n, 0 \models \varphi$.
From the lemma above we deduce $w_n, 1 \models \varphi$ and $w_{n-1}, 0 \models \varphi$.
This is a contradiction since $w_{n-1} \notin (\Sigma_p \cdot \Sigma)^\omega$.

Solution of Exercise 2

Note that all formulae in this exercise are actually in CTL$^+$ hence they have equivalent CTL versions (See TD 4).

Throughout this exercise, M is an arbitrary Kripke structure.

[3] **a)** We show that $\mathsf{A}(\alpha \vee \beta) \equiv \alpha \vee \mathsf{A}\,\beta$ if α is a *state* formula. Let s be a state of M.

- Assume $M, s \models \alpha$. Then $M, s \models \mathsf{A}(\alpha \vee \beta)$ and $M, s \models \alpha \vee \mathsf{A}\,\beta$.

- Assume $M, s \not\models \alpha$. Then, for all run σ of M with $\sigma(0) = s$ we have $M, \sigma, 0 \not\models \alpha$. Therefore, $M, s \models \mathsf{A}(\alpha \vee \beta)$ iff $M, s \models \mathsf{A}\,\beta$ iff $M, s \models \alpha \vee \mathsf{A}\,\beta$.

We prove now that $\varphi_1 \equiv \psi_1$. We use the classical identity $\mathsf{AX}\,\alpha \equiv \mathsf{AX}\,\mathsf{A}\,\alpha$.

$$\begin{aligned}\varphi_1 = \mathsf{A}(\mathsf{X}\,p \vee \mathsf{F}\,q) &\equiv \mathsf{A}(\mathsf{X}\,p \vee q \vee \mathsf{X}\,\mathsf{F}\,q) \equiv q \vee \mathsf{A}(\mathsf{X}\,p \vee \mathsf{X}\,\mathsf{F}\,q) \equiv q \vee \mathsf{AX}(p \vee \mathsf{F}\,q) \\ &\equiv q \vee \mathsf{AX}\,\mathsf{A}(p \vee \mathsf{F}\,q) \equiv q \vee \mathsf{AX}(p \vee \mathsf{AF}\,q) = \psi_1\end{aligned}$$

For $n > 1$, we define $\psi_n = q \vee \mathsf{AX}\,\psi_{n-1} \in$ CTL.
We show by induction on n that for all $n \geq 1$, $\varphi_n \equiv \psi_n$.
The case $n = 1$ was already proved. For $n > 1$, we have:

$$\begin{aligned}\varphi_n = \mathsf{A}(\mathsf{X}^n\,p \vee \mathsf{F}\,q) &\equiv \mathsf{A}(\mathsf{X}^n\,p \vee q \vee \mathsf{X}\,\mathsf{F}\,q) \equiv q \vee \mathsf{A}(\mathsf{X}^n\,p \vee \mathsf{X}\,\mathsf{F}\,q) \equiv q \vee \mathsf{AX}(\mathsf{X}^{n-1}\,p \vee \mathsf{F}\,q) \\ &\equiv q \vee \mathsf{AX}\,\mathsf{A}(\mathsf{X}^{n-1}\,p \vee \mathsf{F}\,q) = q \vee \mathsf{AX}\,\varphi_{n-1} \equiv q \vee \mathsf{AX}\,\psi_{n-1} = \psi_n\end{aligned}$$

[2] **b)** We show that the CTL* formula $\varphi = \mathsf{A}(\mathsf{X}\,p \vee (q\,\mathsf{U}\,r))$ is equivalent to the CTL formula $\psi = r \vee \mathsf{AX}\,p \vee (q \wedge \mathsf{AX}(p \wedge \mathsf{A}\,q\,\mathsf{U}\,r))$.

We use the classical identity $\mathsf{A}(\alpha \wedge \beta) \equiv \mathsf{A}\,\alpha \wedge \mathsf{A}\,\beta$.

$$\begin{aligned}\varphi = \mathsf{A}(\mathsf{X}\,p \vee (q\,\mathsf{U}\,r)) &\equiv \mathsf{A}(\mathsf{X}\,p \vee r \vee (q \wedge \mathsf{X}(q\,\mathsf{U}\,r))) \\ &\equiv r \vee \mathsf{A}(\mathsf{X}\,p \vee (q \wedge \mathsf{X}(q\,\mathsf{U}\,r))) \\ &\equiv r \vee \mathsf{A}((q \vee \mathsf{X}\,p) \wedge (\mathsf{X}\,p \vee \mathsf{X}(q\,\mathsf{U}\,r))) \\ &\equiv r \vee [\mathsf{A}(q \vee \mathsf{X}\,p) \wedge \mathsf{AX}(p \vee (q\,\mathsf{U}\,r))] \\ &\equiv r \vee [(q \vee \mathsf{AX}\,p) \wedge \mathsf{AX}\,\mathsf{A}(p \vee (q\,\mathsf{U}\,r))] \\ &\equiv r \vee [(q \vee \mathsf{AX}\,p) \wedge \mathsf{AX}(p \vee \mathsf{A}\,q\,\mathsf{U}\,r)] \\ &\equiv r \vee (q \wedge \mathsf{AX}(p \vee \mathsf{A}\,q\,\mathsf{U}\,r)) \vee (\mathsf{AX}\,p \wedge \mathsf{AX}(p \vee \mathsf{A}\,q\,\mathsf{U}\,r)) \\ &\equiv r \vee (q \wedge \mathsf{AX}(p \vee \mathsf{A}\,q\,\mathsf{U}\,r)) \vee \mathsf{AX}\,p = \psi\end{aligned}$$

Note that $r \vee [(q \vee \mathsf{AX}\,p) \wedge \mathsf{AX}(p \vee \mathsf{A}\,q\,\mathsf{U}\,r)]$ is already a CTL formula.

[5] **c)** We show that the CTL* formula $\varphi = \mathsf{A}((p\,\mathsf{U}\,q) \vee (p'\,\mathsf{U}\,q'))$ is equivalent to the CTL formula $\psi = \mathsf{A}((p \wedge p')\,\mathsf{U}\,((\mathsf{A}\,p\,\mathsf{U}\,q) \vee (\mathsf{A}\,p'\,\mathsf{U}\,q')))$.

We show first that φ implies ψ. Let s be a state of M and assume that $M, s \models \varphi$. Let σ be a run of M with $\sigma(0) = s$. We have $M, \sigma, 0 \models \mathsf{F}\,q \vee \mathsf{F}\,q'$ hence there is a maximal $k \geq 0$ such that for all $0 \leq j < k$, $M, \sigma, j \models p \wedge p' \wedge \neg(q \vee q')$. We show that $M, \sigma, k \models (\mathsf{A}\,p\,\mathsf{U}\,q) \vee (\mathsf{A}\,p'\,\mathsf{U}\,q')$. If $M, \sigma, k \models q \vee q'$ this is clear. So assume that $M, \sigma, k \not\models q \vee q'$. By maximality of k we get $M, \sigma, k \not\models p \wedge p'$. For instance $M, \sigma, k \not\models p'$ and we prove that $M, \sigma, k \models \mathsf{A}\,p\,\mathsf{U}\,q$. So consider a run σ' with $\sigma'(0) = \sigma(k)$. Then, $\sigma'' = \sigma(0) \cdots \sigma(k-1)\sigma'$ is a run of M starting from s. For all $j \leq k$ we have $M, \sigma, j \models \neg q'$

and also $M, \sigma, k \models \neg p'$ hence $M, \sigma'', 0 \not\models p' \,\mathsf{U}\, q'$. We deduce that $M, \sigma'', 0 \models p \,\mathsf{U}\, q$. Since for all $j < k$ we have $M, \sigma, j \models \neg q$ we deduce that $M, \sigma'', k \models p \,\mathsf{U}\, q$. Therefore, $M, \sigma', 0 \models p \,\mathsf{U}\, q$ as desired. We have shown that $M, \sigma, k \models \mathsf{A}\, p \,\mathsf{U}\, q$. Since for all $j < k$ we have $M, \sigma, j \models p \wedge p'$ we deduce that $M, \sigma, 0 \models (p \wedge p') \,\mathsf{U}\, ((\mathsf{A}\, p \,\mathsf{U}\, q) \vee (\mathsf{A}\, p' \,\mathsf{U}\, q'))$. Finally, we have shown that $M, s \models \psi$.

Conversely, we prove that ψ implies φ. Let s be a state of M and assume that $M, s \models \psi$. Let σ be a run of M with $\sigma(0) = s$. Let $k \geq 0$ with $M, \sigma, k \models (\mathsf{A}\, p \,\mathsf{U}\, q) \vee (\mathsf{A}\, p' \,\mathsf{U}\, q')$ and $M, \sigma, j \models p \wedge p'$ for all $j < k$. We deduce that $M, \sigma, k \models (p \,\mathsf{U}\, q) \vee (p' \,\mathsf{U}\, q')$. Therefore, we also have $M, \sigma, 0 \models (p \,\mathsf{U}\, q) \vee (p' \,\mathsf{U}\, q')$. Hence, we have shown that $M, s \models \varphi$.

Solution of Exercise 3

In this exercise, there are more or less implicit assumptions which are made explicit first. The cabin is always at precisely one floor and it may only move to the next or previous floor:

$$\psi_1 = \mathsf{AG} \bigvee_i (\mathtt{at}_i \wedge \neg \bigvee_{j \neq i} \mathtt{at}_j) \wedge \mathsf{AG} \bigwedge_i (\mathtt{at}_i \rightarrow \mathsf{AX}(\mathtt{at}_{i-1} \vee \mathtt{at}_i \vee \mathtt{at}_{i+1}))$$

where $\mathtt{at}_0 = \mathtt{at}_{n+1} = \bot$. Calls are not cancelled before they are answered:

$$\psi_2 = \mathsf{AG} \bigwedge_j [(\mathtt{at}_j \wedge \neg\mathtt{closed}) \rightarrow \neg\mathtt{call}_j] \wedge [\mathtt{call}_j \rightarrow \mathsf{AX}(\mathtt{call}_j \vee (\mathtt{at}_j \wedge \neg\mathtt{closed}))]$$

The history predicate $\mathtt{up}$ behaves as follows:

$$\psi_3 = \mathsf{AG} \bigwedge_i [(\mathtt{at}_i \wedge \mathtt{up}) \rightarrow \mathsf{AX}(\neg\mathtt{up} \leftrightarrow \mathtt{at}_{i-1})] \wedge [(\mathtt{at}_i \wedge \neg\mathtt{up}) \rightarrow \mathsf{AX}(\mathtt{up} \leftrightarrow \mathtt{at}_{i+1})]$$

[6] **a)** (i) Whenever the cabin is moving between consecutive floors, the door should be and should stay closed:

$$\varphi_1 = \mathsf{AG} \bigwedge_i ((\mathtt{at}_i \wedge \mathsf{EX}\, \neg\mathtt{at}_i) \rightarrow \mathtt{closed}) \wedge (\mathtt{at}_i \rightarrow \mathsf{AX}(\neg\mathtt{at}_i \rightarrow \mathtt{closed}))$$

(ii) The lift system stays *idle* if and only if there are no pending calls. We first define some macros:

$$\mathtt{pendingCall} = \bigvee_i \mathtt{call}_i$$

$$\mathtt{possiblyIdle} = \bigvee_i [\mathtt{at}_i \wedge \mathtt{closed} \wedge \mathsf{EX}(\mathtt{at}_i \wedge \mathtt{closed})] \vee [\mathtt{at}_i \wedge \neg\mathtt{closed} \wedge \mathsf{EX}(\mathtt{at}_i \wedge \neg\mathtt{closed})]$$

$$\mathtt{certainlyIdle} = \bigvee_i [\mathtt{at}_i \wedge \mathtt{closed} \wedge \mathsf{AX}(\mathtt{at}_i \wedge \mathtt{closed})] \vee [\mathtt{at}_i \wedge \neg\mathtt{closed} \wedge \mathsf{AX}(\mathtt{at}_i \wedge \neg\mathtt{closed})]$$

The desired specification is

$$\varphi_2 = \mathsf{AG}(\mathtt{pendingCall} \rightarrow \neg\mathtt{possiblyIdle}) \wedge \mathsf{AG}(\neg\mathtt{pendingCall} \rightarrow \mathtt{certainlyIdle})$$

(*iii*) The cabin never moves in a direction toward which there are no pending calls:

$$\varphi_3 = \mathsf{AG} \bigwedge_i ((\mathtt{at}_i \wedge \mathsf{EX}\, \mathtt{at}_{i+1}) \rightarrow \bigvee_{j>i} \mathtt{call}_j) \wedge ((\mathtt{at}_i \wedge \mathsf{EX}\, \mathtt{at}_{i-1}) \rightarrow \bigvee_{j<i} \mathtt{call}_j)$$

(*iv*) The cabin travels in its current direction satisfying all calls until no more exist in the current direction. This is the most difficult specification and we will split it in 3 parts.

First, a natural requirement would be that if a call is pending for the current floor it should be answered immediately. This can be written $\mathsf{AG} \bigwedge_j [(\mathtt{at}_j \wedge \mathtt{call}_j) \rightarrow \mathsf{AX}\, \neg\mathtt{closed}]$. Note that the premise $\mathtt{at}_j \wedge \mathtt{call}_j$ implies $\mathtt{closed}$ by ψ_2 and the conclusion $\neg\mathtt{closed}$ together with φ_1 implies $\mathtt{at}_j$ and then $\neg\mathtt{call}_j$ by ψ_2 so that the call is answered for all next state. But this requirement is too strong since it would prevent the lift system to answer calls for other floors if whenever the door closes a call for the current floor is simultaneously placed. So we give a weaker specification stating that a call pending for floor j must be answered immediately *the first time we reach* floor j:

$$\varphi_4^1 = \mathsf{AG} \bigwedge_j [\neg\mathtt{at}_j \rightarrow \mathsf{AX}((\mathtt{at}_j \wedge \mathtt{call}_j) \rightarrow \mathsf{AX}\, \neg\mathtt{closed})]$$

Next we state that the cabin must eventually leave the current floor if there is a call pending for some other floor:

$$\varphi_4^2 = \mathsf{AG} \bigwedge_i \neg\, \mathsf{EG}(\mathtt{at}_i \wedge \bigvee_{j\neq i} \mathtt{call}_j)$$

Finally, we state that the cabin does not change direction as long as there are calls pending for the current direction:

$$\varphi_4^3 = \mathsf{AG} \bigwedge_i [(\mathtt{at}_i \wedge \mathtt{up} \wedge \bigvee_{j>i} \mathtt{call}_j) \rightarrow \mathsf{AX}(\mathtt{at}_i \vee \mathtt{at}_{i+1})] \wedge [(\mathtt{at}_i \wedge \neg\mathtt{up} \wedge \bigvee_{j<i} \mathtt{call}_j) \rightarrow \mathsf{AX}(\mathtt{at}_i \vee \mathtt{at}_{i-1})]$$

[6] **b)** We have to show that the conjunction of all specifications above (including the assumptions $\psi_1 \wedge \psi_2 \wedge \psi_3$) implies $\mathsf{AG} \bigwedge_j [\mathtt{call}_j \rightarrow \mathsf{AF}(\mathtt{at}_j \wedge \neg\mathtt{closed})]$.
Remark: I have accepted more or less informal answers. Below, I demonstrate that it is also possible to give a formal (though not easy) proof.

We fix some $1 \leq j \leq n$ and for each state s of the model M we define a *rank* by:

$$r_j(s) = \begin{cases} 0 & \text{if } M, s \models \neg\mathtt{call}_j \\ j - i & \text{if } i < j \text{ and } M, s \models \mathtt{at}_i \wedge \mathtt{call}_j \wedge \mathtt{up} \\ i - j & \text{if } j < i \text{ and } M, s \models \mathtt{at}_i \wedge \mathtt{call}_j \wedge \neg\mathtt{up} \\ n - i + n - j & \text{if } j < i \text{ and } M, s \models \mathtt{at}_i \wedge \mathtt{call}_j \wedge \mathtt{up} \\ i - 1 + j - 1 & \text{if } i < j \text{ and } M, s \models \mathtt{at}_i \wedge \mathtt{call}_j \wedge \neg\mathtt{up} \\ 2\max(i-1, n-i) & \text{if } i = j \text{ and } M, s \models \mathtt{at}_i \wedge \mathtt{call}_j \end{cases}$$

and we will use an induction based on this rank to prove that for all infinite run σ of M and all $k \geq 0$, we have $\sigma, k \models \mathtt{call}_j \rightarrow \mathsf{F}(\mathtt{at}_j \wedge \neg\mathtt{closed})$.

The result is clear if $\sigma, k \models \neg\mathtt{call}_j$, i.e. when $r_j(\sigma(k)) = 0$. Assume now that $\sigma, k \models \mathtt{call}_j$ and let i be such that $\sigma, k \models \mathtt{at}_i$ (which exists and is unique by ψ_1).

Case 1: $i = j$. From ψ_2 we know that the door is closed and by φ_2 the lift system cannot stay idle, hence we have $\sigma, k+1 \models \neg(\texttt{closed} \wedge \texttt{at}_i)$. If $\sigma, k+1 \models \neg\texttt{closed}$ then by φ_1 we deduce that $\sigma, k+1 \models \texttt{at}_i$. Therefore, $\sigma, k \models \mathsf{F}(\texttt{at}_j \wedge \neg\texttt{closed})$ and we are done. Otherwise $\sigma, k+1 \models \neg\texttt{at}_i$. By $\psi_2 \wedge \psi_3$ we get $\sigma, k+1 \models \texttt{call}_j \wedge (\texttt{at}_{i+1} \wedge \texttt{up} \vee \texttt{at}_{i-1} \wedge \neg\texttt{up})$. We deduce with a direct computation that $r_j(\sigma(k+1)) < r_j(\sigma(k))$. We obtain by induction that $\sigma, k+1 \models \mathsf{F}(\texttt{at}_j \wedge \neg\texttt{closed})$ hence also $\sigma, k \models \mathsf{F}(\texttt{at}_j \wedge \neg\texttt{closed})$ and we are done.

Case 2: $j < i$ and $\sigma, k \models \texttt{up}$ (the case $i < j$ and $\sigma, k \models \neg\texttt{up}$ is symmetric).
First, using $\psi_2 \wedge \varphi_4^2$ we deduce that $\sigma, k \models \mathsf{F}\,\neg\texttt{at}_i$. Next, using $\psi_1 \wedge \psi_2 \wedge \psi_3$ we deduce that $\sigma, k \models (\texttt{at}_i \wedge \texttt{call}_j \wedge \texttt{up})\,\mathsf{U}\,(\texttt{at}_{i+1} \wedge \texttt{up} \vee \texttt{at}_{i-1} \wedge \neg\texttt{up})$. Hence, there exists $k' > k$ such that $\sigma, k' \models \texttt{at}_{i+1} \wedge \texttt{up} \vee \texttt{at}_{i-1} \wedge \neg\texttt{up}$ and $\sigma, k'-1 \models \texttt{at}_i \wedge \texttt{call}_j \wedge \texttt{up}$. From φ_1 we deduce that $\sigma, k' \models \texttt{closed}$ and using ψ_2 we get $\sigma, k' \models \texttt{call}_j \wedge (\texttt{at}_{i+1} \wedge \texttt{up} \vee \texttt{at}_{i-1} \wedge \neg\texttt{up})$.
If $\sigma, k' \models \texttt{call}_j \wedge \texttt{at}_{i+1} \wedge \texttt{up}$ or if $j < i-1$ and $\sigma, k' \models \texttt{call}_j \wedge \texttt{at}_{i-1} \wedge \neg\texttt{up}$ then $r_j(\sigma(k')) < r_j(\sigma(k))$ and we conclude by induction. Now, if $j = i-1$ and $\sigma, k' \models \texttt{at}_j \wedge \texttt{call}_j$ then using $\varphi_4^1 \wedge \varphi_1$ we deduce that $\sigma, k'+1 \models \neg\texttt{closed} \wedge \texttt{at}_j$, thus the call is also eventually answered.

Case 3: $i < j$ and $\sigma, k \models \texttt{up}$ (the case $j < i$ and $\sigma, k \models \neg\texttt{up}$ is symmetric).
First, using $\psi_2 \wedge \varphi_4^2$ we deduce that $\sigma, k \models \mathsf{F}\,\neg\texttt{at}_i$. Next, using $\varphi_4^3 \wedge \psi_2 \wedge \psi_3$ we deduce that $\sigma, k \models (\texttt{at}_i \wedge \texttt{call}_j \wedge \texttt{up})\,\mathsf{U}\,(\texttt{at}_{i+1} \wedge \texttt{up})$. Hence, there exists $k' > k$ such that $\sigma, k' \models \texttt{at}_{i+1} \wedge \texttt{up}$ and $\sigma, k'-1 \models \texttt{at}_i \wedge \texttt{call}_j \wedge \texttt{up}$. From φ_1 we deduce that $\sigma, k' \models \texttt{closed}$ and using ψ_2 we get $\sigma, k' \models \texttt{call}_j \wedge \texttt{at}_{i+1} \wedge \texttt{up}$.
If $i+1 < j$ then $r_j(\sigma(k')) < r_j(\sigma(k))$ and we can conclude by induction. Now, if $j = i+1$ then using $\varphi_4^1 \wedge \varphi_1$ we deduce that $\sigma, k'+1 \models \neg\texttt{closed} \wedge \texttt{at}_j$, thus the call is also eventually answered.

[3] **c)** (*i*) We should still answer cabin calls as before, but we do not answer landing calls if the cabin is $\frac{2}{3}$-full. The specifications ψ_1, ψ_2, ψ_3 and φ_1 are not affected. We define the macro $\texttt{npCall}_j = \texttt{cabinCall}_j \vee (\texttt{landingCall}_j \wedge \neg\tfrac{2}{3}\texttt{-full})$ which states that there is a call for floor j which should not be posponed. The new specification is obtained by replacing $\texttt{call}_j$ by $\texttt{npCall}_j$ in the specifications φ_2, φ_3, and $\varphi_4 = \varphi_4^1 \wedge \varphi_4^2 \wedge \varphi_4^3$. Note that landing calls are no more necessarily answered. For instance, if the cabin is $\frac{2}{3}$-full and there are only landing calls then by φ_2 the system is idle until some cabin call is placed.

(*ii*) We do as above but with another definition of the macro: $\texttt{npCall}_n = \texttt{call}_n$ and for $j < n$ we let $\texttt{npCall}_j = \texttt{cabinCall}_j \vee (\texttt{landingCall}_j \wedge \neg\texttt{call}_n)$.

(*iii*) The two policies are conflicting. We decide to give the priority to the first one. The new definition for the macro is $\texttt{npCall}_n = \texttt{cabinCall}_n \vee (\texttt{landingCall}_n \wedge \neg\tfrac{2}{3}\texttt{-full})$ and for $j < n$ we let $\texttt{npCall}_j = \texttt{cabinCall}_j \vee (\texttt{landingCall}_j \wedge \neg\tfrac{2}{3}\texttt{-full} \wedge \neg\texttt{call}_n)$.