Formale Spezifikation und Verifikation
Temporallogik

Wintersemester 2023/24
Übungsblatt 09

30. Januar 2024

1 Temporallogik a)–c)

Jetzt scheint die Sonne ($\texttt{sun}$) und irgendwann regnet es ($\texttt{rain}$).

$$\underbrace{\texttt{sun}}_{\text{gilt jetzt}} \wedge (\Diamond \texttt{rain})$$

1 Temporallogik a)–c)

Jetzt scheint die Sonne ($\texttt{sun}$) und irgendwann regnet es ($\texttt{rain}$).

$$\underbrace{\texttt{sun}}_{\text{gilt jetzt}} \wedge (\Diamond \texttt{rain}) \qquad \text{Sicherheit} + \text{Lebendigkeit}$$

1 Temporallogik a)–c)

Jetzt scheint die Sonne (`sun`) und irgendwann regnet es (`rain`).

$$\underbrace{\texttt{sun}}_{\text{gilt jetzt}} \wedge (\Diamond \texttt{rain}) \qquad \text{Sicherheit} + \text{Lebendigkeit}$$

Während die Webcam an ist (`webcam`) leuchtet das entsprechende Licht (`light`)

1 Temporallogik a)–c)

Jetzt scheint die Sonne ($\texttt{sun}$) und irgendwann regnet es ($\texttt{rain}$).

$$\underbrace{\texttt{sun}}_{\text{gilt jetzt}} \wedge (\Diamond\texttt{rain}) \qquad \text{Sicherheit + Lebendigkeit}$$

Während die Webcam an ist ($\texttt{webcam}$) leuchtet das entsprechende Licht ($\texttt{light}$)
"Immer: wenn die Webcam gerade an ist, dann muss jetzt auch das Licht leuchten"

$$\Box(\underbrace{\texttt{webcam} \Rightarrow \texttt{light}}_{\text{selber Zeitpunkt}})$$

1 Temporallogik a)–c)

Jetzt scheint die Sonne (`sun`) und irgendwann regnet es (`rain`).

$$\underbrace{\texttt{sun}}_{\text{gilt jetzt}} \wedge (\Diamond \texttt{rain}) \qquad \text{Sicherheit + Lebendigkeit}$$

Während die Webcam an ist (`webcam`) leuchtet das entsprechende Licht (`light`)
"Immer: wenn die Webcam gerade an ist, dann muss jetzt auch das Licht leuchten"

$$\Box\underbrace{(\texttt{webcam} \Rightarrow \texttt{light})}_{\text{selber Zeitpunkt}} \qquad \text{Sicherheit (\underline{Invariante})}$$

1 Temporallogik a)–c)

Jetzt scheint die Sonne (`sun`) und irgendwann regnet es (`rain`).

$\underbrace{\texttt{sun}}_{\text{gilt jetzt}} \wedge (\Diamond \texttt{rain})$ Sicherheit + Lebendigkeit

Während die Webcam an ist (`webcam`) leuchtet das entsprechende Licht (`light`)
"Immer: wenn die Webcam gerade an ist, dann muss jetzt auch das Licht leuchten"

$\Box(\underbrace{\texttt{webcam} \Rightarrow \texttt{light}}_{\text{selber Zeitpunkt}})$ Sicherheit (<u>Invariante</u>)

Nach jedem Drücken der Taste `shutdown` [...] später garantiert aus (`off`).

$\Box(\texttt{shutdown} \Rightarrow (\Diamond \texttt{off}))$

1 Temporallogik a)–c)

Jetzt scheint die Sonne (`sun`) und irgendwann regnet es (`rain`).

$\underbrace{\texttt{sun}}_{\text{gilt jetzt}} \wedge (\Diamond \texttt{rain})$ Sicherheit + Lebendigkeit

Während die Webcam an ist (`webcam`) leuchtet das entsprechende Licht (`light`)
"Immer: wenn die Webcam gerade an ist, dann muss jetzt auch das Licht leuchten"

$\Box(\underbrace{\texttt{webcam} \Rightarrow \texttt{light}}_{\text{selber Zeitpunkt}})$ Sicherheit (Invariante)

Nach jedem Drücken der Taste `shutdown` [...] später garantiert aus (`off`).

$\Box(\texttt{shutdown} \Rightarrow (\Diamond \texttt{off}))$ Lebendigkeit (Response)

1 Temporallogik d)

Man darf nicht Auto fahren ($\neg$`drive`) bevor man einen Führerschein hat (`license`). Achten Sie darauf, *nicht* zu erzwingen, dass irgendwann ein Führerschein vorhanden ist. Zwischen welchen beiden temporalen Operatoren müssen Sie den richtigen wählen?

1 Temporallogik d)

Man darf nicht Auto fahren ($\neg\texttt{drive}$) bevor man einen Führerschein hat ($\texttt{license}$).
Achten Sie darauf, *nicht* zu erzwingen, dass irgendwann ein Führerschein vorhanden ist.
Zwischen welchen beiden temporalen Operatoren müssen Sie den richtigen wählen?

Auswahl zwischen: "until" $\mathcal{U}$ (✗) und "weak until" $\mathcal{W}$ (✓)

$$(\neg\texttt{drive})\ \mathcal{W}\ \texttt{license}$$

1 Temporallogik d)

Man darf nicht Auto fahren ($\neg$`drive`) bevor man einen Führerschein hat (`license`). Achten Sie darauf, *nicht* zu erzwingen, dass irgendwann ein Führerschein vorhanden ist. Zwischen welchen beiden temporalen Operatoren müssen Sie den richtigen wählen?

Auswahl zwischen: “until” $\mathcal{U}$ (✗) und “weak until” $\mathcal{W}$ (✓)

$(\neg\texttt{drive})\ \mathcal{W}\ \texttt{license}$ Sicherheit

1 Temporallogik d)

Man darf nicht Auto fahren ($\neg\mathtt{drive}$) bevor man einen Führerschein hat ($\mathtt{license}$). Achten Sie darauf, *nicht* zu erzwingen, dass irgendwann ein Führerschein vorhanden ist. Zwischen welchen beiden temporalen Operatoren müssen Sie den richtigen wählen?

Auswahl zwischen: “until” $\mathcal{U}$ (✗) und “weak until” $\mathcal{W}$ (✓)

$(\neg\mathtt{drive})\ \mathcal{W}\ \mathtt{license}$ Sicherheit

Alternative Formulierung (nicht äquivalent!)

$\square(\mathtt{drive} \Rightarrow \mathtt{license})$ Sicherheit

1 Temporallogik d)

Man darf nicht Auto fahren ($\neg\texttt{drive}$) bevor man einen Führerschein hat ($\texttt{license}$). Achten Sie darauf, *nicht* zu erzwingen, dass irgendwann ein Führerschein vorhanden ist. Zwischen welchen beiden temporalen Operatoren müssen Sie den richtigen wählen?

Auswahl zwischen: “until” $\mathcal{U}$ (✗) und “weak until” $\mathcal{W}$ (✓)

$(\neg\texttt{drive})\ \mathcal{W}\ \texttt{license}$ Sicherheit

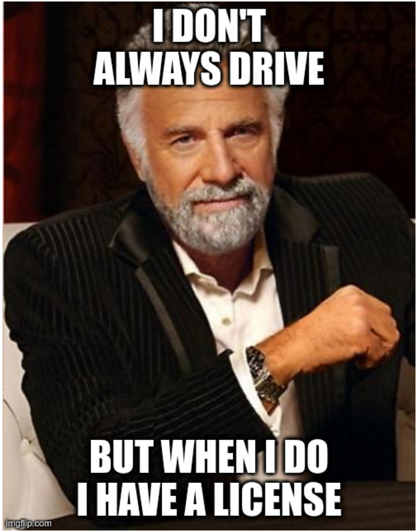
Alternative Formulierung (nicht äquivalent!)

$\Box(\texttt{drive} \Rightarrow \texttt{license})$ Sicherheit

Unterschied: Die erste Formulierung geht davon aus, dass man den Führerschein nicht wieder verliert

1 Temporallogik d)

$$\square(\texttt{drive} \Rightarrow \texttt{license})$$



1 Temporallogik e)

Ein Licht blinkt einmal

1 Temporallogik e)

Ein Licht blinkt einmal

1 Temporallogik e)

Ein Licht blinkt jetzt zweimal ($\texttt{light}$)
Möglichkeit 1:

$$\texttt{light} \wedge (\circ(\neg\texttt{light})) \wedge (\circ\circ\texttt{light}) \wedge (\circ\circ\circ(\neg\texttt{light}))$$

1 Temporallogik e)

Ein Licht blinkt jetzt zweimal ($\texttt{light}$)
Möglichkeit 1:

$$\texttt{light} \land (\circ(\neg\texttt{light})) \land (\circ \circ \texttt{light}) \land (\circ \circ \circ(\neg\texttt{light}))$$

Sicherheitseigenschaft (nur *endlich* viele Schritte beschrieben)

1 Temporallogik e)

Ein Licht blinkt jetzt zweimal ($\texttt{light}$)
Möglichkeit 1:

$$\texttt{light} \wedge (\circ(\neg\texttt{light})) \wedge (\circ \circ \texttt{light}) \wedge (\circ \circ \circ(\neg\texttt{light}))$$

Sicherheitseigenschaft (nur *endlich* viele Schritte beschrieben)

Möglichkeit 2: keine konkrete Angabe der "Blink-Zeitpunkte"

$$\texttt{light}\,\mathcal{U}\,(\neg\texttt{light})\,\mathcal{U}\,\texttt{light}\,\mathcal{U}\,(\neg\texttt{light})$$

Lebendigkeitseigenschaft (an/aus müssen sich in endlicher Zeit abwechseln)

1 Temporallogik f)

Nachdem die Temperatur t einmal 30° überschritten hat, sinkt t nie wieder darunter.

1 Temporallogik f)

Nachdem die Temperatur t einmal 30° überschritten hat, sinkt t nie wieder darunter.
Immer: wenn jetzt $t > 30$, dann ab da immer: $t > 30$.

$$\square\Big((t > 30) \Rightarrow (\square(t > 30))\Big)$$

1 Temporallogik f)

Nachdem die Temperatur t einmal 30° überschritten hat, sinkt t nie wieder darunter.
Immer: wenn jetzt $t > 30$, dann ab da immer: $t > 30$.

$$\Box\Big((t > 30) \Rightarrow (\Box(t > 30))\Big)$$ Sicherheit

1 Temporallogik f)

Nachdem die Temperatur t einmal 30° überschritten hat, sinkt t nie wieder darunter.
Immer: wenn jetzt $t > 30$, dann ab da immer: $t > 30$.

$$\Box\Big((t > 30) \Rightarrow (\Box(t > 30))\Big) \qquad \text{Sicherheit}$$

Äquivalent Alternative: Es gilt eine Zeit lang $t \leq 30$, und falls irgendwann der Fall eintritt, dass nicht mehr $t \leq 30$, dann gilt ab da immer: $t > 30$

$$(t \leq 30)\ \mathcal{W}\ (\Box(t > 30))$$

1 Temporallogik g)

Eine Ampel soll in der Zukunft garantiert irgendwann installiert werden, sobald diese in den Betrieb geht, soll Sie immer mal wieder auf grün stehen (`green`).

1 Temporallogik g)

Eine Ampel soll in der Zukunft garantiert irgendwann installiert werden, sobald diese in den Betrieb geht, soll Sie immer mal wieder auf grün stehen ($\texttt{green}$).

Zunächst: Eine Ampel soll immer mal wieder auf grün stehen ($\texttt{green}$).

$$\Box(\Diamond\texttt{green})$$

1 Temporallogik g)

Eine Ampel soll in der Zukunft garantiert irgendwann installiert werden, sobald diese in den Betrieb geht, soll Sie immer mal wieder auf grün stehen ($\texttt{green}$).

Zunächst: Eine Ampel soll immer mal wieder auf grün stehen ($\texttt{green}$).

$\Box(\Diamond \texttt{green})$ Lebendigkeit (Rekurrenz)

1 Temporallogik g)

Eine Ampel soll in der Zukunft garantiert irgendwann installiert werden, sobald diese in den Betrieb geht, soll Sie immer mal wieder auf grün stehen ($\mathtt{green}$).

Zunächst: Eine Ampel soll immer mal wieder auf grün stehen ($\mathtt{green}$).

$\Box(\Diamond \mathtt{green})$ Lebendigkeit (Rekurrenz)

Diese Formel gilt dann irgendwann

$\Diamond(\Box(\Diamond \mathtt{green}))$

1 Temporallogik g)

Eine Ampel soll in der Zukunft garantiert irgendwann installiert werden, sobald diese in den Betrieb geht, soll Sie immer mal wieder auf grün stehen ($\mathtt{green}$).

Zunächst: Eine Ampel soll immer mal wieder auf grün stehen ($\mathtt{green}$).

$\Box(\Diamond\mathtt{green})$ Lebendigkeit (Rekurrenz)

Diese Formel gilt dann irgendwann

$\Diamond(\Box(\Diamond\mathtt{green}))$ Lebendigkeit

1 Temporallogik h)

Programm zählt i von 0 bis n hoch und beendet sich genau dann wenn $i = n$.

1 Temporallogik h)

Programm zählt i von 0 bis n hoch und beendet sich genau dann wenn $i = n$.

$$(i = 0) \land \Big((\underbrace{0 \leq i < n}_{\approx \text{ Invariante}}) \, \mathcal{U} \, (i = n) \Big)$$

Außerdem formuliert: Das Programm beendet sich garantiert

1 Temporallogik h)

Programm zählt i von 0 bis n hoch und beendet sich genau dann wenn $i = n$.

$$(i = 0) \wedge \Big((\underbrace{0 \leq i < n}_{\approx \text{ Invariante}}) \, \mathcal{U} \, (i = n) \Big)$$

Außerdem formuliert: Das Programm beendet sich garantiert
$\Rightarrow$ Sicherheit + Lebendigkeit

1 Temporallogik h)

Programm zählt i von 0 bis n hoch und beendet sich genau dann wenn $i = n$.

$$(i = 0) \wedge \Big((\underbrace{0 \leq i < n}_{\approx \text{ Invariante}}) \,\mathcal{U}\, (i = n) \Big)$$

Außerdem formuliert: Das Programm beendet sich garantiert
$\Rightarrow$ Sicherheit + Lebendigkeit

Schwierig in LTL: Über zwei aufeinanderfolgende Zustände reden, z.B. $i' = i + 1$.
($\circ\ i$ ist *keine* Formel, da i keine Formel ist)

1 Temporallogik h)

Programm zählt i von 0 bis n hoch und beendet sich genau dann wenn $i = n$.

$$(i = 0) \land \Big((\underbrace{0 \leq i < n}_{\approx \text{ Invariante}})\, \mathcal{U}\, (i = n)\Big)$$

Außerdem formuliert: Das Programm beendet sich garantiert
$\Rightarrow$ Sicherheit $+$ Lebendigkeit

Schwierig in LTL: Über zwei aufeinanderfolgende Zustände reden, z.B. $i' = i + 1$.
($\circ\, i$ ist *keine* Formel, da i keine Formel ist)

Präzisere Formulierungen (benötigt Erweiterung der Logik)

- mit gestrichenen Variablen: jetzt i und im Nachfolger i'
- mit Hilfsvariablen und Quantoren: $(\exists k.\ (i = k) \land \circ(i = k + 1))\, \mathcal{U}\, (i = n)$

1 Temporallogik h) — Variante

Programm zählt i hoch und irgendwann bleibt $i = n$

1 Temporallogik h) — Variante

Programm zählt i hoch und irgendwann bleibt $i = n$

$$\Diamond(\Box(i = n))$$

1 Temporallogik h) — Variante

Programm zählt i hoch und irgendwann bleibt $i = n$

$\Diamond(\Box(i = n))$ Lebendigkeit (Stabilität)

Eigenschaft kann nur auf unendlich langen Abläufen wiederlegt werden

2 Ampelsteuerung

a) Die Ampeln sind nie gleichzeitig grün:

$$\Box\neg(G_1 \wedge G_2)$$

2 Ampelsteuerung

a) Die Ampeln sind nie gleichzeitig grün:

$$\Box\neg(G_1 \wedge G_2)$$

b) Jede Ampel wechselt unbegrenzt oft zwischen rot und grün:

Ampel 1 $\quad \Box\Big(\Diamond(G_1 \wedge \neg R_1) \wedge \Diamond(R_1 \wedge \neg G_1)\Big)$

Ampel 2 $\quad \Box\Big(\Diamond(G_2 \wedge \neg R_2) \wedge \Diamond(R_2 \wedge \neg G_2)\Big)$

Wir vermeiden, dass beide Ampeln dauerhaft rot und grün zeigen.

2 Ampelsteuerung

a) Die Ampeln sind nie gleichzeitig grün:

$$\Box\neg(G_1 \wedge G_2)$$

b) Jede Ampel wechselt unbegrenzt oft zwischen rot und grün:

Ampel 1 $\Box\Big(\Diamond(G_1 \wedge \neg R_1) \wedge \Diamond(R_1 \wedge \neg G_1)\Big)$

Ampel 2 $\Box\Big(\Diamond(G_2 \wedge \neg R_2) \wedge \Diamond(R_2 \wedge \neg G_2)\Big)$

Wir vermeiden, dass beide Ampeln dauerhaft rot und grün zeigen.

c) Ampel 1 bleibt rot bis garantiert Ampel 2 auf rot steht:

$$R_1 \,\mathcal{U}\, R_2$$

2 Ampelsteuerung

a) Die Ampeln sind nie gleichzeitig grün:

$$\Box\neg(G_1 \wedge G_2)$$

b) Jede Ampel wechselt unbegrenzt oft zwischen rot und grün:

$$\text{Ampel 1} \quad \Box\Big(\Diamond(G_1 \wedge \neg R_1) \wedge \Diamond(R_1 \wedge \neg G_1)\Big)$$

$$\text{Ampel 2} \quad \Box\Big(\Diamond(G_2 \wedge \neg R_2) \wedge \Diamond(R_2 \wedge \neg G_2)\Big)$$

Wir vermeiden, dass beide Ampeln dauerhaft rot und grün zeigen.

c) Ampel 1 bleibt rot bis garantiert Ampel 2 auf rot steht:

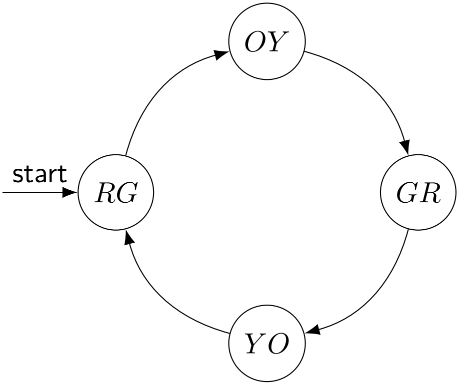
$$R_1 \,\mathcal{U}\, R_2$$

- Möglich: Ampel 1 schaltet 2× grün ohne dass Ampel 2 grün wird

3 Ampelsteuerung als Transitionssystem

Modellierung I

Ampeln schalten gleichzeitig um



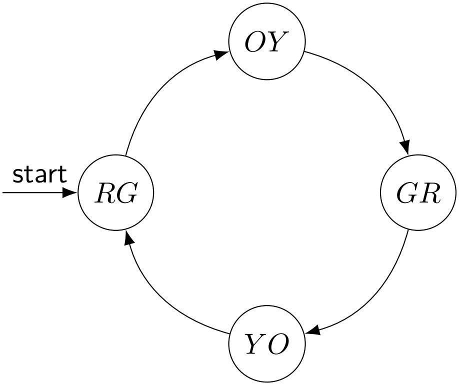
Zustände pro Ampel

G (green), Y (yellow), R (red),
O (orange $=$ red $+$ yellow)
(vgl. Beamer/Licht im Hörsaal)

3 Ampelsteuerung als Transitionssystem

Modellierung I

Ampeln schalten gleichzeitig um



Zustände pro Ampel

G (green), Y (yellow), R (red),
O (orange $=$ red $+$ yellow)
(vgl. Beamer/Licht im Hörsaal)

Was gilt wann? Beispielsweise:

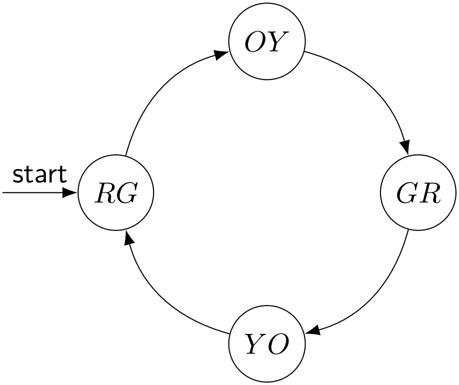
$RG \models R_1 \wedge G_2$

$OY \models \neg R_1 \wedge \neg R_2$ (siehe Angabe)

3 Ampelsteuerung als Transitionssystem

Modellierung I

Ampeln schalten gleichzeitig um



Zustände pro Ampel

G (green), Y (yellow), R (red),
O (orange = red + yellow)
(vgl. Beamer/Licht im Hörsaal)

Was gilt wann? Beispielsweise:

$$RG \models R_1 \wedge G_2$$

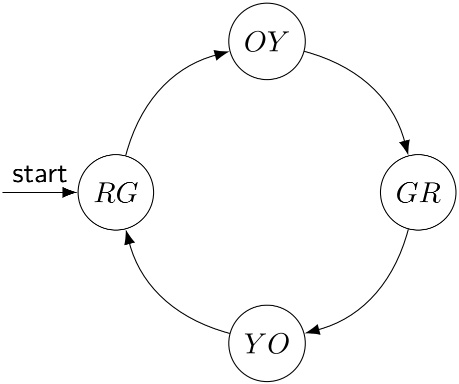
$$OY \models \neg R_1 \wedge \neg R_2$$ (siehe Angabe)

Eigenschaften

a) $\Box(\neg(G_1 \wedge G_2))$

3 Ampelsteuerung als Transitionssystem

Modellierung I
Ampeln schalten gleichzeitig um



Zustände pro Ampel
G (green), Y (yellow), R (red),
O (orange = red + yellow)
(vgl. Beamer/Licht im Hörsaal)

Was gilt wann? Beispielsweise:

$$RG \models R_1 \wedge G_2$$
$$OY \models \neg R_1 \wedge \neg R_2 \qquad \text{(siehe Angabe)}$$

Eigenschaften

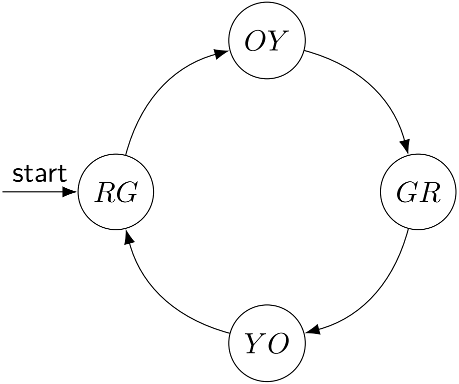
a) $\Box(\neg(G_1 \wedge G_2))$ ✓

b) $\Box\Big(\Diamond(G_1 \wedge \neg R_1) \wedge \Diamond(R_1 \wedge \neg G_1)\Big)$

3 Ampelsteuerung als Transitionssystem

Modellierung I

Ampeln schalten gleichzeitig um



Zustände pro Ampel

G (green), Y (yellow), R (red),
O (orange $=$ red $+$ yellow)
(vgl. Beamer/Licht im Hörsaal)

Was gilt wann? Beispielsweise:

$RG \models R_1 \wedge G_2$

$OY \models \neg R_1 \wedge \neg R_2$ (siehe Angabe)

Eigenschaften

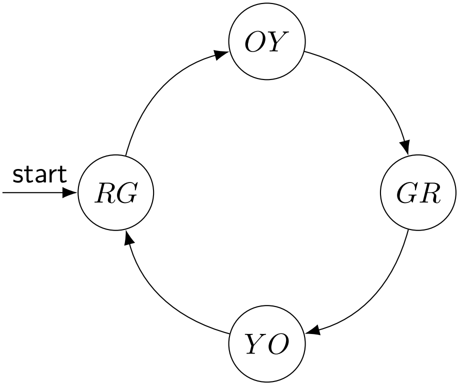
a) $\Box(\neg(G_1 \wedge G_2))$ ✓

b) $\Box\big(\Diamond(G_1 \wedge \neg R_1) \wedge \Diamond(R_1 \wedge \neg G_1)\big)$ ✓
(analog für Ampel 2)

c) $R_1 \,\mathcal{U}\, R_2$

3 Ampelsteuerung als Transitionssystem

Modellierung I
Ampeln schalten gleichzeitig um



Zustände pro Ampel
G (green), Y (yellow), R (red),
O (orange = red + yellow)
(vgl. Beamer/Licht im Hörsaal)

Was gilt wann? Beispielsweise:

$RG \models R_1 \wedge G_2$
$OY \models \neg R_1 \wedge \neg R_2$ (siehe Angabe)

Eigenschaften

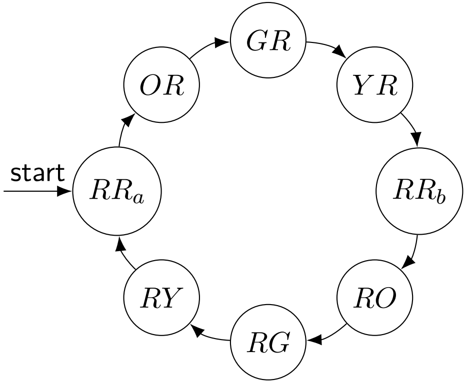
a) $\Box(\neg(G_1 \wedge G_2))$ ✓

b) $\Box\big(\Diamond(G_1 \wedge \neg R_1) \wedge \Diamond(R_1 \wedge \neg G_1)\big)$ ✓
(analog für Ampel 2)

c) $R_1 \,\mathcal{U}\, R_2$ ✗

3 Ampelsteuerung als Transitionssystem (II)

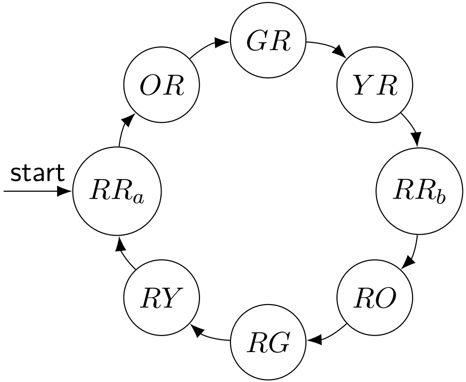
Modellierung II
Mit ordentlicher Rotphase



Beachte: zwei unterschiedliche rot/rot Zustände, RR_a und RR_b, um unterscheiden zu können, welche Ampel als nächstes grün werden soll.

Modellierung II

Mit ordentlicher Rotphase



Beachte: zwei unterschiedliche rot/rot Zustände, RR_a und RR_b, um unterscheiden zu können, welche Ampel als nächstes grün werden soll.

Eigenschaften

a) $\Box(\neg(G_1 \wedge G_2))$ ✓

b) $\Box\Big(\Diamond(G_1 \wedge \neg R_1) \wedge \Diamond(R_1 \wedge \neg G_1)\Big)$ ✓
(analog für Ampel 2)

c) $R_1 \,\mathcal{U}\, R_2$ und auch
$\Box(R_1 \Rightarrow (R_1 \,\mathcal{U}\, R_2))$ ✓