4. Aufgabenblatt zur Vorlesung

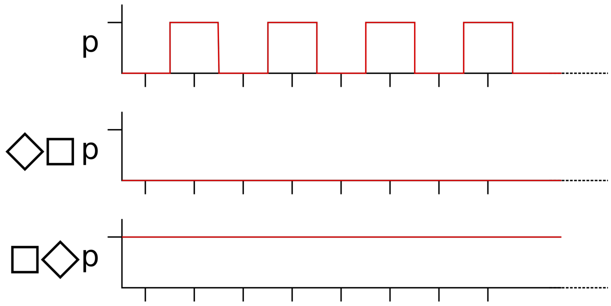
Nichtsequentielle und Verteilte Programmierung SoSe 2018

Anton Oehler, Mark Niehues

Aufgabe 1 Lineare Temporale Logik I *10 Punkte*

(a)

$$\Diamond\Box\, p \Leftrightarrow \Box\Diamond\, p$$



Anhand der Grafik lässt sich erkennen, dass die Aussage nicht für alle Belegungen gilt.

(b)

$$((\Box\, p \Rightarrow \Diamond\, q) \wedge \Diamond\Box\, q) \quad \Rightarrow \quad \Diamond\, q \tag{1}$$

Die Aussage gilt für alle Belegungen, da die schwächere Aussage $\Diamond\Box\, q \Rightarrow \Diamond\, q$ für alle Belegungen gilt: Wenn ab einem Zeitpunkt q immer *true* ist, dann ist q auch irgendwann in der Zukunft *true*.

(c) (i) zu Zeigen: $(\Box\, p \wedge \Box\, q) \Leftrightarrow \Box(p \wedge q)$
$\Box\, p_i \wedge \Box\, q_i := \forall j \geq i : p_j \wedge \forall j \geq i : q_j => \forall j \geq i : p_j \wedge q_j$
$\Box(p_i \wedge q_i) := \forall j \geq i : p_j \wedge q_j$
q.e.d.

(ii) zu Zeigen: $(\Diamond\, p \vee \Diamond\, q) \Leftrightarrow \Diamond(p \vee q)$
$\Diamond\, p_i \vee \Diamond\, q_i := \exists j \geq i : p_j \vee \exists j \geq i : q_j => \exists j \geq i : p \vee q$
$\Diamond(p \vee q) := \exists j \geq i : p \vee q$
q.e.d.

Aufgabe 2 Lineare Temporale Logik II *10 Punkte*

(a) (i) $I_x(\alpha U \beta) = I_x(\alpha)\dot{U} I_x(\beta)$ mit $\dot{U} : \{t, f\}^{NxN} \rightarrow \{t, f\}^N$
wobei $\dot{U}((a_i)_{i\in N}, (b_i)_{i\in N}) = (w_i)_{i\in N}$
und $w_i = \begin{cases} t & falls\, \exists j \geq i : b_j = t \wedge \forall k \in [i, ..., j) : a_k = t \\ f & sonst \end{cases}$

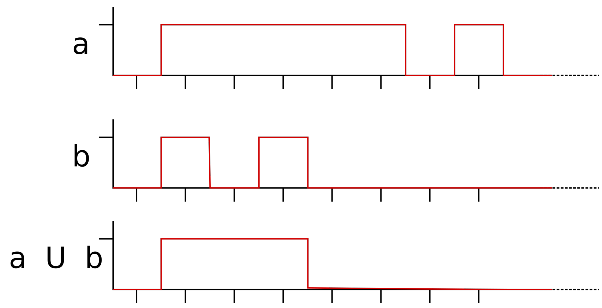


Abbildung 1: Beispiel für den *bis* Operator

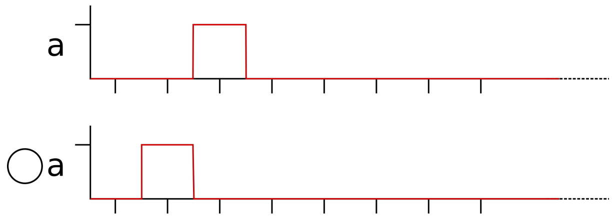


Abbildung 2: Beispiel für den *nächster* Operator

(ii) $I_x(\bigcirc\alpha) = \dot{\bigcirc} I_x(\alpha)$ mit $\dot{\bigcirc} : \{t, f\}^N \to \{t, f\}^N$
wobei $\dot{\bigcirc}((a_i)_{i\in N}) = (w_i)_{i\in N}$
und $w_i = \begin{cases} t, & falls\, a_{i+1} = t \\ f, & sonst \end{cases}$

(b) Falls mehrere Threads im Spiel sind, ist der nächste Schritt unter Annahme schwacher Fairness nicht genau definiert.

(c) $\Diamond\, a \equiv true\, U a$, da letzteres true ist, wenn $\exists j \geq i : a_j = true \wedge true$

$\Box\, a \equiv \neg(\Diamond\,(\neg a)) \equiv \neg(true\, U \neg a)$, die Herleitung folgt dementsprechend aus dem 1. Fall.

Aufgabe 3 Der Algorthmus von Peterson II *10 Punkte*

```
======================================================
  global adrin = false, bdrin = false, letzter = a
a1: U                         | b1: U
a2: adrin  <- true            | b2: bdrin  <- true
a3: letzter <- a              | b3: letzter <- b
a4: while bdrin &&            | b4: while adrin &&
          letzter = a do      |           letzter = b do
a5:     NOP                   | b5:     NOP
a6: K                         | b6: K
a7: adrin <- false            | b7: bdrin <- false
```

Für alle möglichen Zustandsfolgen gilt:

- $\Box(a_1 \Rightarrow \Diamond(a_2 \vee a_\bot))$
- $\Box(a_2 \Rightarrow \Diamond a_3)$
- $\Box(a_3 \Rightarrow \Diamond a_4)$
- $\Box(a_4 \Rightarrow \Diamond(a_5 \vee a_6)$
- $\Box(a_5 \Rightarrow \Diamond a_4)$
- $\Box(a_6 \Rightarrow \Diamond a_7)$
- $\Box(a_7 \Rightarrow \Diamond a_1)$
- $\Box(a_\bot \Rightarrow \Box a_\bot)$
- Analog für die Zustände von b

Außerdem legen wir folgende Invarianten fest:

1) $\Box(letzter = a \vee letzter = b)$
2) $\Box(adrin \equiv a_3 \vee a_4 \vee a_5 \vee a_6 \vee a_7)$
3) $\Box(bdrin \equiv b_3 \vee b_4 \vee b_5 \vee b_6 \vee b_7)$

(a) Zu beweisen:

- $(a_6 \wedge (b_4 \vee b_5)) \Rightarrow (adrin \wedge letzter = b)$
  Beweis durch Widerspruch:

$$\neg((a_6 \wedge (b_4 \vee b_5)) \Rightarrow (adrin \wedge letzter = b))$$

$$(a_6 \wedge (b_4 \vee b_5)) \wedge \neg(adrin \wedge letzter = b)$$

$$(a_6 \wedge (b_4 \vee b_5)) \wedge (\neg adrin \vee letzter \neq b)$$

  Aus Invariante 1) folgt:

$$(a_6 \wedge (b_4 \vee b_5)) \wedge (\neg adrin \vee letzter = a)$$

  Nach Invariante 2) und a_6 ist $adrin = true$:

$$(a_6 \wedge (b_4 \vee b_5)) \wedge (false \vee letzter = a)$$

$$a_6 \wedge (b_4 \vee b_5) \wedge letzter = a$$

  Aus $(b_4 \vee b_5)$ und Invariante 3) folgt $bdrin = true$
  b befindet sich in der Schleife, wartet also darauf dass a $adrin$ auf $false$ setzt. Ist aber $letzter = a$ und $bdrin = true$, so hätte a niemals von a_4 nach a_6 gehen dürfen.
  Folglich kann nicht gleichzeitig a_6, $(b_4 \vee b_5)$ und $letzter = a$ gelten: *Widerspruch!*
- $((a_4 \vee a_5) \wedge b_6) \Rightarrow (bdrin \wedge letzter = a)$
  Beweis analog zu vorherigem, nur mit a und b vertauscht.

(b) Zu zeigen:

- $\Box\big[(a_4 \land \Box\neg a_6) \Rightarrow \Box\Diamond(bdrin \land letzter \neq b)\big]$
  Invariante 1): $letzter \neq b \equiv letzter = a$
  $$\Box\big[(a_4 \land \Box\neg a_6) \Rightarrow \Box\Diamond(bdrin \land letzter = a)\big]$$
  Es gilt immer, dass, wenn a einerseits bei a_4, also vor bei der Überprüfung der Schleife ist, andererseits *niemals*(immer nicht) a_6 eintreten wird, daraus folgt, dass immer irgendwann $bdrin = true$ und $letzter = a$ sein muss.
  Beweis durch Widerspruch: Nehmen wir an, a ist bei a_4 und es wird niemals a_6 eintreten, irgendwann wird aber $bdrin = false$ oder $letzter \neq a$. Dann könnte im Folgenden a einen Schritt machen, die Schleifenbedingungen sind nicht erfüllt und a würde von a_4 in a_6 übergehen. Dass niemals a_6 eintreten kann wurde aber vorausgesetzt: *Widerspruch!*
  Formal:
  $$\neg\left(\Box\left[(a_4 \land \Box\neg a_6) \Rightarrow \Box\Diamond(bdrin \land letzter = a)\right]\right)$$
  $$\Diamond\left(\neg\left[(a_4 \land \Box\neg a_6) \Rightarrow \Box\Diamond(bdrin \land letzter = a)\right]\right)$$
  $$\Diamond\left[(a_4 \land \Box\neg a_6) \land \neg\Box\Diamond(bdrin \land letzter = a)\right]$$
  $$\Diamond\left[(a_4 \land \Box\neg a_6) \land \Diamond\neg\Diamond(bdrin \land letzter = a)\right]$$
  $$\Diamond\left[(a_4 \land \Box\neg a_6) \land \Diamond\Box\neg(bdrin \land letzter = a)\right]$$
  $$\Diamond\left[(a_4 \land \Box\neg a_6) \land \Diamond\Box(\neg bdrin \lor letzter \neq a)\right]$$
  Ist nun $bdrin = false$ oder $letzter \neq a$, wird der rechte Teilausdruck zu *true* ausgewertet, der linke Teilausdruck jedoch *irgendwann*(nämlich wenn a den nächsten Schritt macht) zu $false$, da a dann bei a_6 ist, was jedoch *niemals* eintreten sollte.
- $\Box\big[\Diamond\Box(\neg bdrin) \lor \Diamond(letzter = b)\big]$
  Es gilt immer, dass entweder irgendwann immer $bdrin = false$ ist oder dass irgendwann $letzter = b$ sein wird.
  Damit *bdrin* dauerhaft (*immer*) $false$ bleiben würde, müsste b in b_1 abstürzen und zu $b_\perp$ übergehen, da *bdrin* nur in b_1 und b_2 $false$ ist und wir eingangs gefordert haben, dass immer auf b_2 irgendwann b_3 folgt, wodurch dem *bdrin* wieder auf *true* gesetzt werden würde. Nur wenn b von a_1 zu $a_\perp$ übergeht, in dem es daraufhin für immer verweilen würde (siehe ebenfalls aufgestellte Bedingungen), kann *bdrin* für immer $false$ bleiben.
  Stürzt b jedoch nicht ab und geht folglich von b_1 zu b_2 über, wird es immer auch irgendwann zu b_3 und weiter zu b_4 übergehen, wonach $letzter = b$ gelten würde, womit $\Diamond(letzter = b)$ gelten würde.
  Also:
  $$\Diamond\Box(\neg bdrin) \equiv b_\perp$$
  Falls $b_\perp$, dann $\Diamond\Box(\neg bdrin)$, sonst $\Diamond(letzter = b)$
- $\Box\big[(a_4 \land \Box\neg a_6 \land \Diamond(letzter = b)) \Rightarrow \Diamond\Box(letzter = b))\big]$

```
======================================================
  global adrin = false, bdrin = false, letzter = a
a1: U                       | b1: U
a2: adrin  <- true          | b2: bdrin  <- true
a3: letzter <- a            | b3: letzter <- b
a4: while bdrin &&          | b4: while adrin &&
          letzter = a do    |           letzter = b do
a5:     NOP                 | b5:     NOP
a6: K                       | b6: K
a7: adrin <- false          | b7: bdrin <- false
```