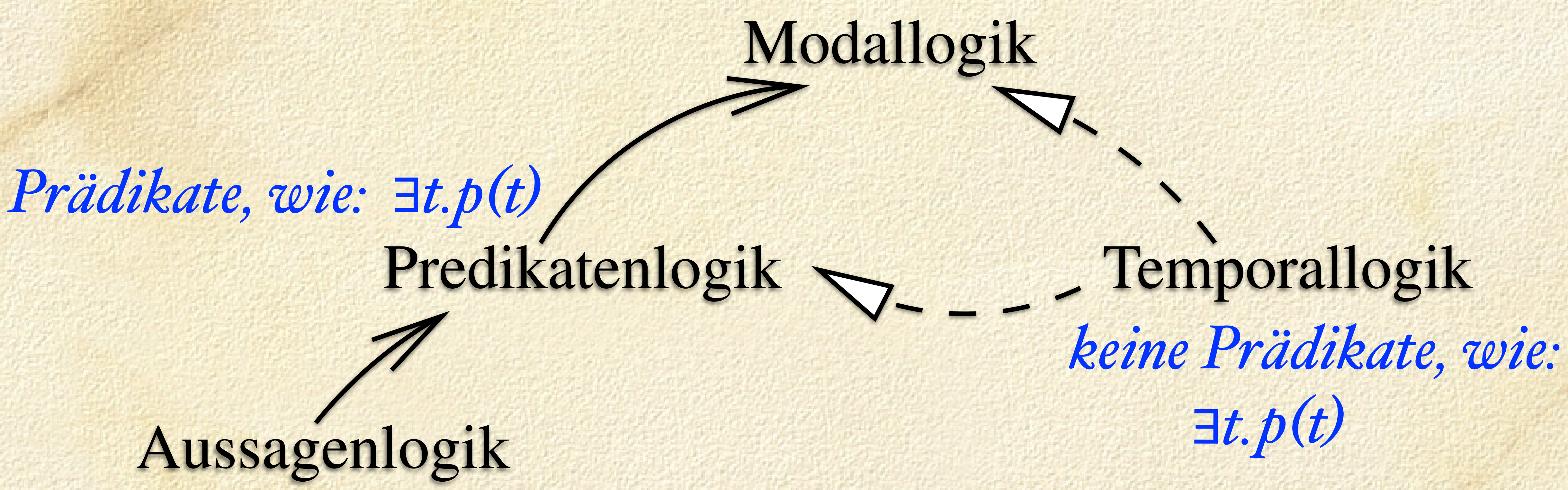
1.5 Temporale Logik



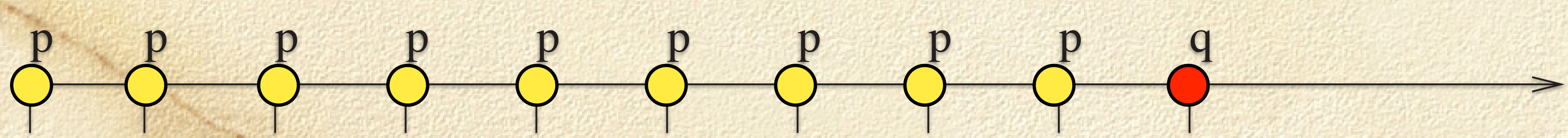
Semantik von LTL-Formeln

Xp „next time“: p gilt im zweitem Element A_1 der Folge (auch $\bigcirc p$ geschrieben),

Fp „eventually, in the future“: p gilt in einem Element einer gegebenen Folge (auch $\Diamond p$),

Gp „always, globally“: p gilt in allen Elementen einer gegebenen Folge (auch $\Box p$),

$p\,U q$ „until“: es gibt einen Element der gegebenen Folge, in dem q gilt und vor diesem Element gilt immer p.

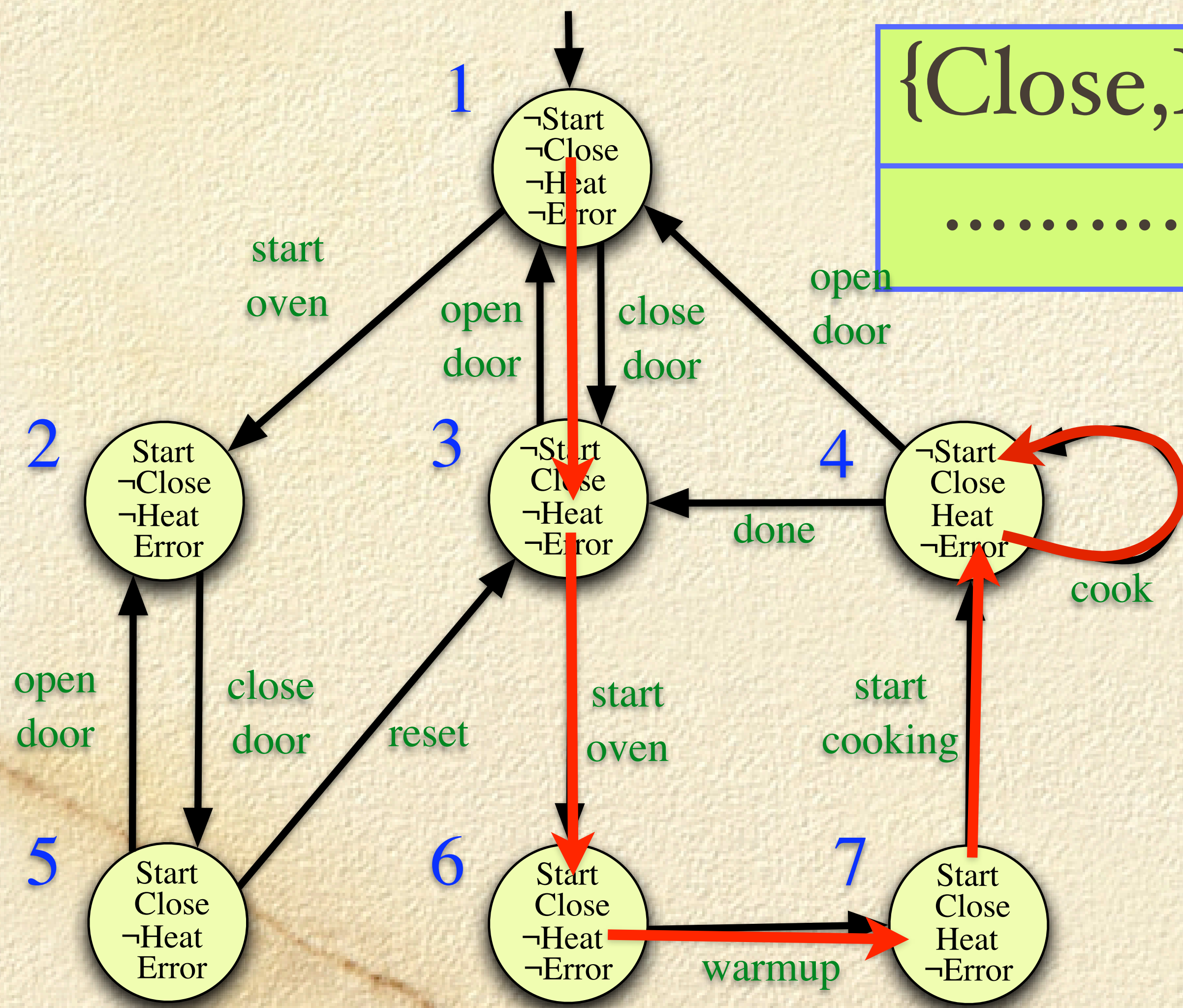


Definition 1.31 *Sei $\alpha = A_0 A_1 A_2 \cdots \in \mathcal{P}(\mathcal{AP})^\omega$ eine unendliche Folge von Aussagenmengen. Dann sei für $i \in \mathbb{N}$ die Folge $\alpha^i = A_i A_{i+1} \cdots$ der i-Suffix von α. In der folgenden induktiven Definition seien $p \in AP$ und f, f_1, f_2 LTL-Formeln. $\boxed{\alpha \models f}$ ist zu lesen als „für die Folge α gilt f“ oder „$\boxed{\alpha \text{ erfüllt } f}$“.*

$\alpha = \emptyset \; \{Close\} \; \{Start,Close\} \; \{Start,Close,Heat\}$

$\{Close,Heat\} \; \{Close,Heat\} \; \{Close,Heat\}$

....



$\alpha^3 = \{Start,Close,Heat\}$
$\{Close,Heat\}$
$\{Close,Heat\}$
$\{Close,Heat\}$
....

Definition 1.31 *Sei $\alpha = A_0A_1A_2 \cdots \in \mathcal{P}(\mathcal{AP})^\omega$ eine unendliche Folge von Aussagenmengen. Dann sei für $i \in \mathbb{N}$ die Folge $\alpha^i = A_iA_{i+1}\cdots$ der i-Suffix von α. In der folgenden induktiven Definition seien $p \in AP$ und f, f_1, f_2 LTL-Formeln. $\boxed{\alpha \models f}$ ist zu lesen als „für die Folge α gilt f“ oder „$\boxed{\alpha \text{ erfüllt } f}$“.*

1. $\alpha \models \textbf{true}$.
2. $\alpha \not\models \textbf{false}$.
3. $\alpha \models p \quad \Leftrightarrow \quad p \in A_0$.

Konstante & elementare Aussagen

4. $\alpha \models \neg f \quad \Leftrightarrow \quad \alpha \not\models f$.
5. $\alpha \models f_1 \vee f_2 \quad \Leftrightarrow \quad \alpha \models f_1$ *oder* $\alpha \models f_2$.
6. $\alpha \models f_1 \wedge f_2 \quad \Leftrightarrow \quad \alpha \models f_1$ *und* $\alpha \models f_2$.

aussagenlogische Junktoren

7. $\alpha \models Xf \quad \Leftrightarrow \quad \alpha^1 \models f$.

nächster Schritt

8. $\alpha \models Ff \quad \Leftrightarrow \quad \exists k \geq 0 : \alpha^k \models f$.
9. $\alpha \models Gf \quad \Leftrightarrow \quad \forall k \geq 0 : \alpha^k \models f$.

irgendwann & immer

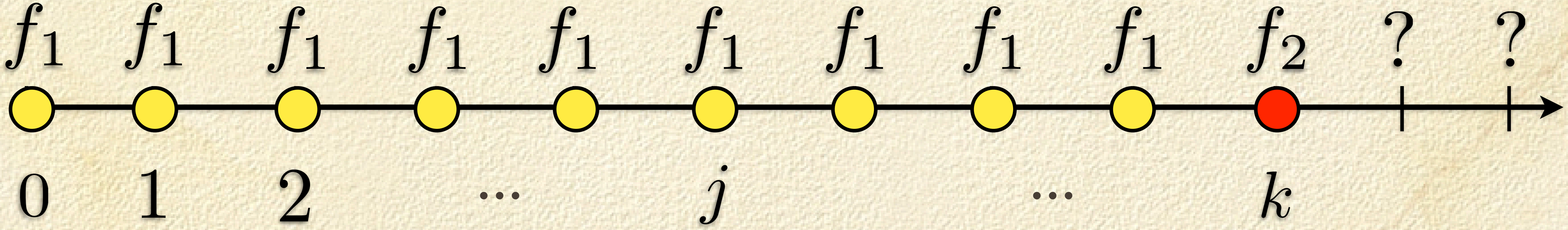
10. $\alpha \models f_1 U f_2 \quad \Leftrightarrow \quad \exists k \geq 0 \,.\, \alpha^k \models f_2$ *und für alle* $0 \leq j < k$ *gilt* $\alpha^j \models f_1$.

until

Definition 1.31 *Sei $\alpha = A_0 A_1 A_2 \cdots \in \mathcal{P}(\mathcal{AP})^\omega$ eine unendliche Folge von Aussagenmengen. Dann sei für $i \in \mathbb{N}$ die Folge $\alpha^i = A_i A_{i+1} \cdots$ der i-Suffix von α. In der folgenden induktiven Definition seien $p \in AP$ und f, f_1, f_2 LTL-Formeln.* $\boxed{\alpha \models f}$ *ist zu lesen als „für die Folge α gilt f“ oder „$\boxed{\alpha \text{ erfüllt } f}$“.*

10. $\alpha \models f_1 U f_2 \quad \Leftrightarrow \quad \exists k \geq 0 \,.\, \alpha^k \models f_2$ *und für alle* $0 \leq j < k$ *gilt* $\alpha^j \models f_1$.

until



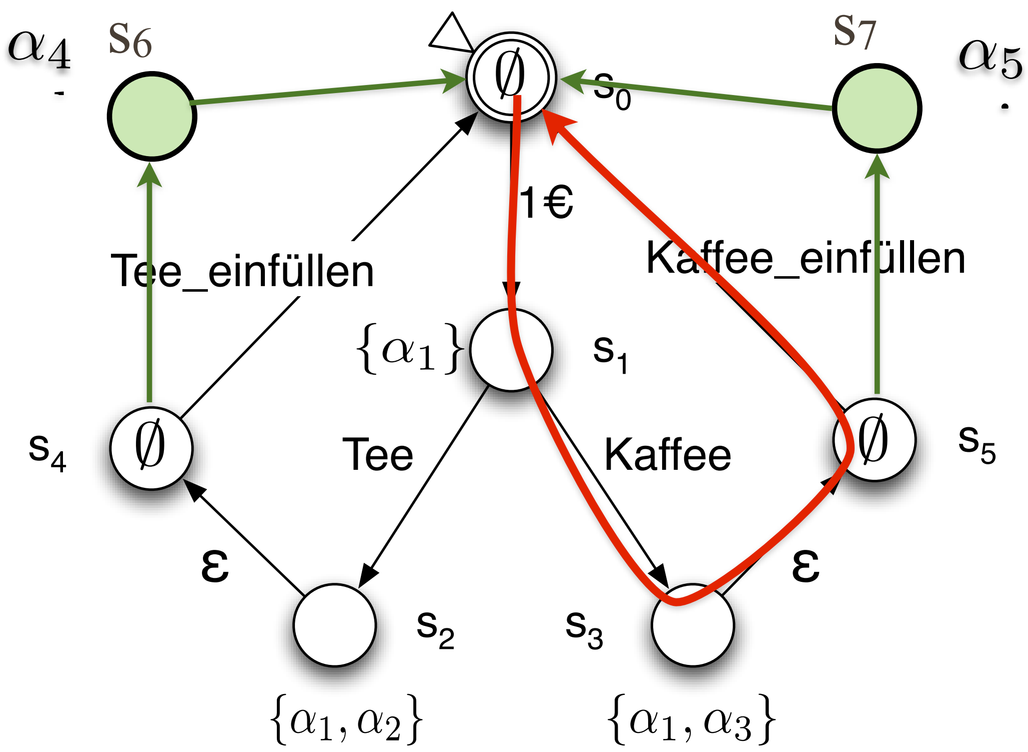
- $F\, f \Leftrightarrow \mathbf{true}\, U\, f$

$L^\omega(f) := \{\alpha \in \mathcal{P}(AP)^\omega \mid \alpha \models f\}$ *heißt Menge der α erfüllenden Folgen über AP oder die* Sprache von α.

Präsenzaufgabe 4.2:

$(\varnothing\ \{\alpha_1\}\ \{\alpha_1, \alpha_2\}\ \varnothing\ )^\omega$

1. Betrachte das TS aus Abb. 1.12. Betrachte die ω-Sprache $L = y^\omega$ mit $y = (s_0 s_1 s_3 s_5)$. Gib die Ettikettensprache $E_S(L)$ an!

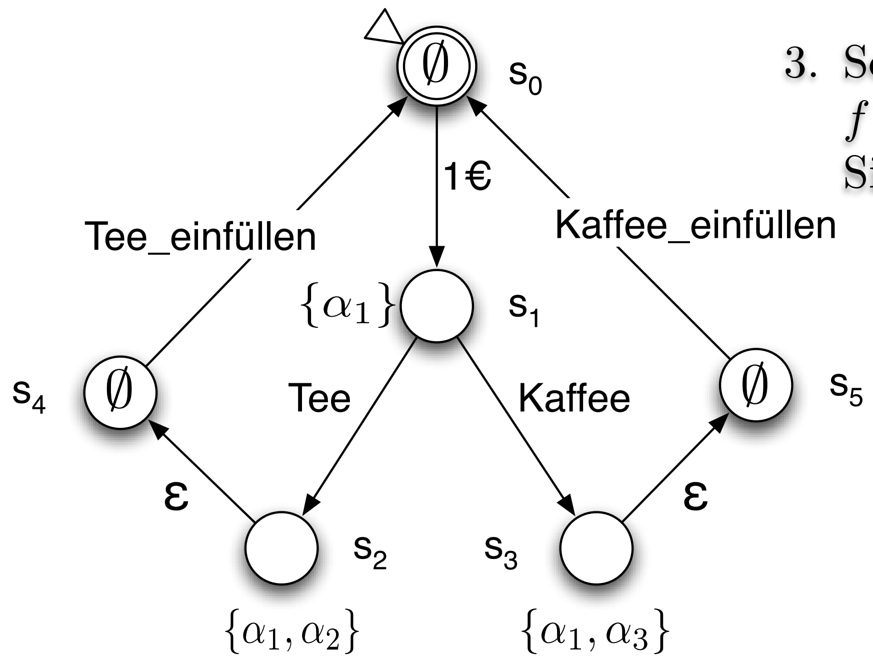


$\alpha_4 =$ „In der Tasse ist Tee.“

$\alpha_5 =$ „In der Tasse ist Kaffee.“

$$\Box(\alpha_2 \implies (\neg\alpha_5\ \mathbf{U}\ \alpha_4))$$

$$\Box(\alpha_2 \implies \bigcirc\bigcirc\alpha_4)$$



3. Sei $AP = \{\alpha_1, \alpha_2, \alpha_3\}$. Gib die Menge $L^\omega(f)$ (vgl. Def. 1.31) für (a) $f = \Box\alpha_2$ und (b) $f = \Diamond\alpha_1$ an! (Beachte, dass die Sprache $L^\omega(f)$ völlig unabhängig vom TS aus Abb. 1.12 ist.) Sie können dabei die folgenden Mengen verwenden ($\alpha \in AP$ und $A \subseteq AP$):

$$\begin{aligned} M(A) &:= \{X \subseteq AP \mid A \subseteq X\} \\ M(\alpha) &:= M(\{\alpha\}) \end{aligned}$$

Lösung: $L^\omega(f)$ ist hier eine unendliche Folgensprache über den logischen Atomen AP, d.h. eine Teilmenge von $\mathcal{P}(AP)^\omega$.

(a) Es ist $L^\omega(\Box\alpha_2) = M(\alpha_2)^\omega = (\{\{\alpha_2\}, \{\alpha_2, \alpha_1\}, \{\alpha_2, \alpha_3\}, \{\alpha_2, \alpha_1, \alpha_3\}\})^\omega$.

(b) Es ist $L^\omega(\Diamond\alpha_1) = \mathcal{P}(AP)^* \cdot M(\alpha_1) \cdot \mathcal{P}(AP)^*$.

$$
\begin{array}{llcl}
8. & \alpha \models F f & \Leftrightarrow & \exists k \geq 0 : \alpha^k \models f. \\
9. & \alpha \models G f & \Leftrightarrow & \forall k \geq 0 : \alpha^k \models f.
\end{array}
$$

irgend wann & immer

- $G\,f \Leftrightarrow \neg F \neg f.$
- $f_1 \wedge f_2 \Leftrightarrow \neg(\neg f_1 \vee \neg f_2),$
- $F\,f \Leftrightarrow \mathbf{true}\,U\,f,$

Eine LTL-Formel f gilt für eine unendliche Folgen $\pi = s_0 s_1 s_2 \cdots$ von Zuständen einer Kripke-Struktur $M := (S, S_0, R, E_S)$, wenn sie für die zugehörige Zustandsetikettenfolge $E_S(\pi) = E_S(s_0)E_S(s_1)E_S(s_2)\cdots \in \mathcal{P}(AP)^\omega$ gilt. Man schreibt $M, \pi \models f$.

Definition 1.33 *Sei $M := (S, S_0, R, E_S)$ eine Kripke-Struktur und $\pi = s_0 s_1 s_2 \cdots \in S^\omega$ eine unendliche Folge von Zuständen von M. Dann sei $M, \pi \models f :\Leftrightarrow E_S(\pi) \models f$.*

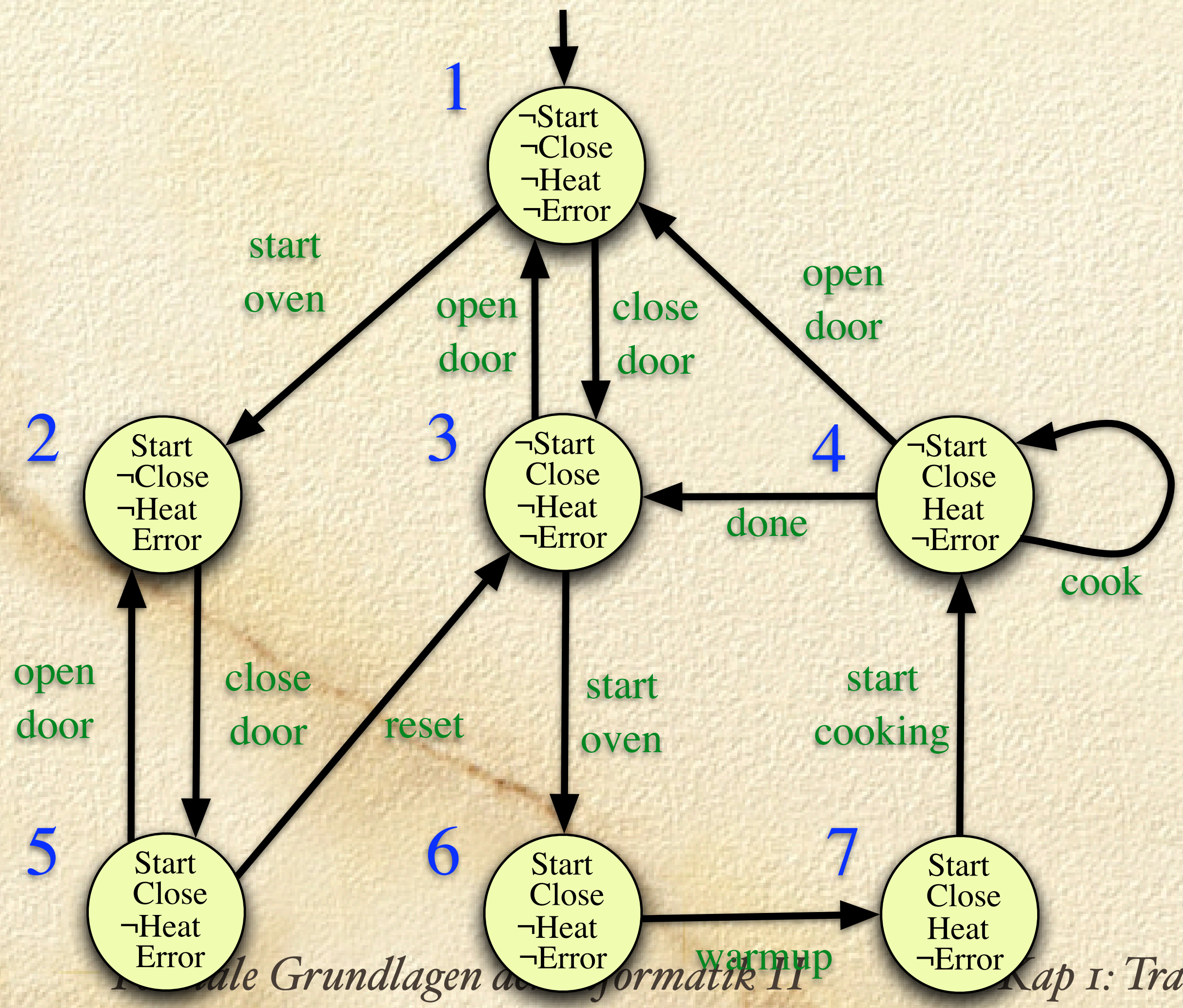


Abb. 1.16
Seite 36

Definition 1.35 *Eine LTL-Formel f gilt (ist gültig) im Zustand $s \in S$ einer Kripke-Struktur $M := (S, S_0, R, E_S)$ falls $M, \pi \models f$ für alle Pfade $\pi = s_0 s_1 s_2 \cdots \in S^\omega$ gilt, die in s beginnen, d.h. $s_0 = s$ (in Zeichen $M, s \models f$). Sie gilt in M, falls sie in allen Anfangszuständen gilt, also: $M \models f :\Leftrightarrow \forall s \in S_0 : M, s \models f$.*

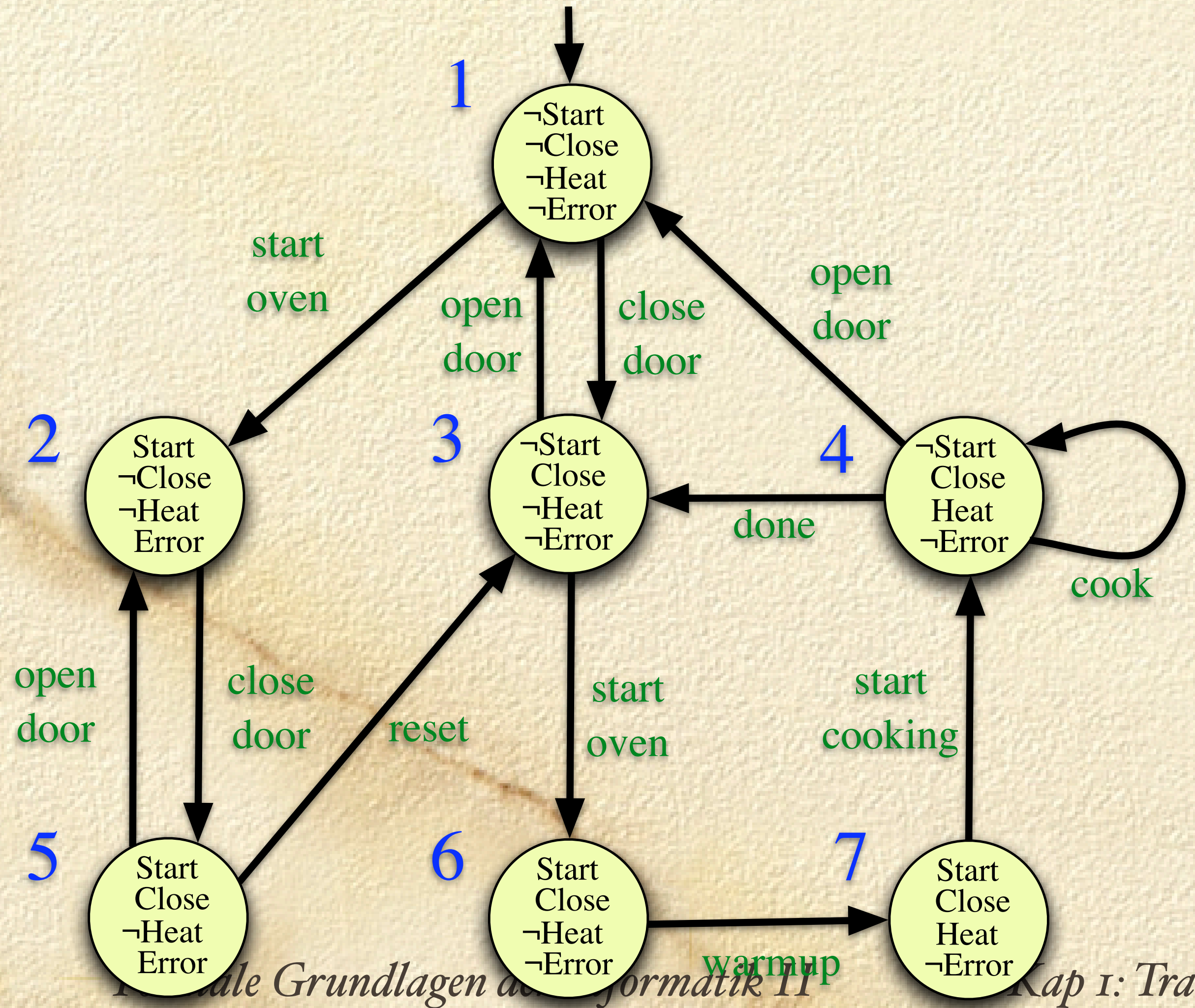
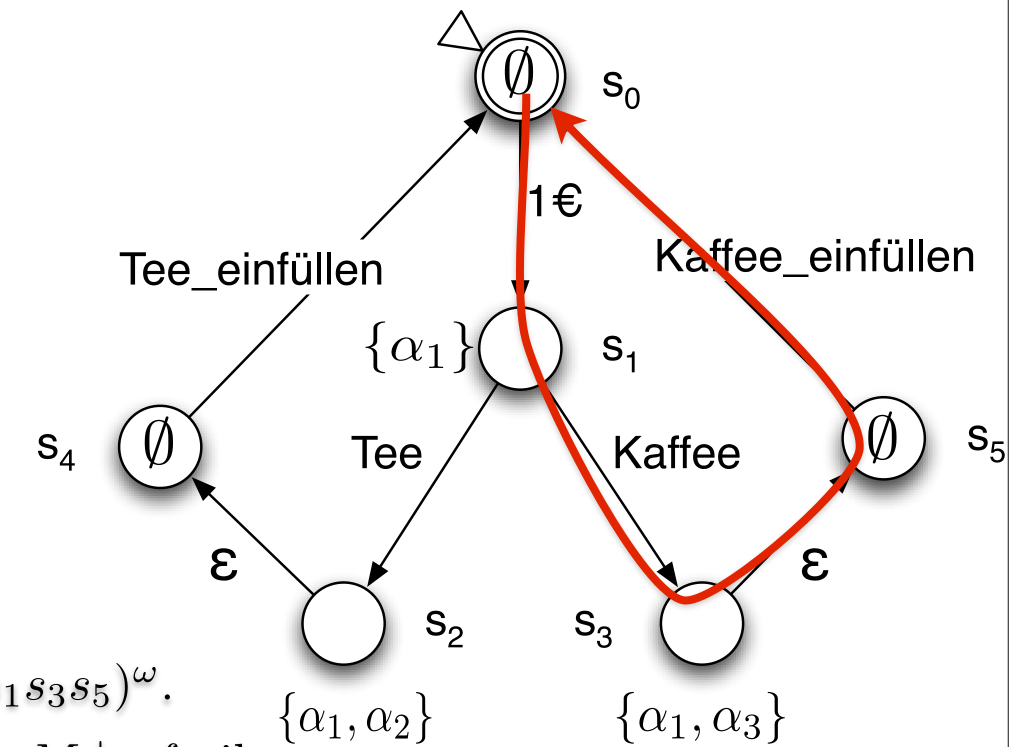


Abb. 1.16
Seite 36



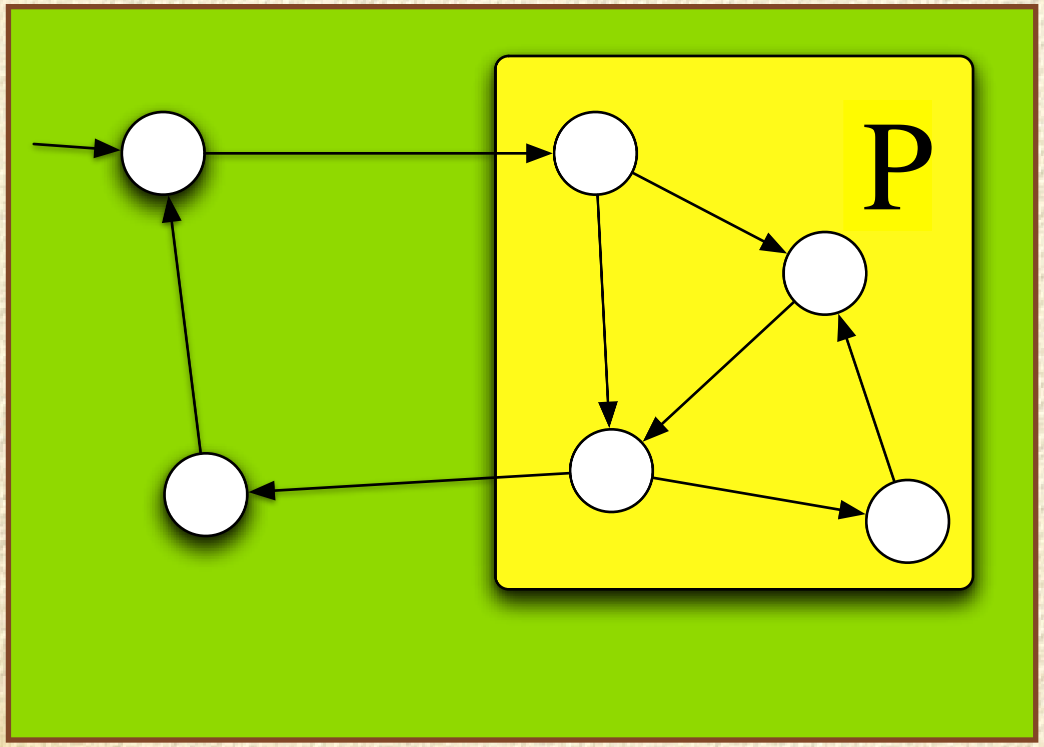
ungsaufgabe 4.4: Betrachten Sie die Kripkestruktur M aus Abb. 1.12.

1. Betrachte einen unendlichen Zustandspfad $\pi = s_0 s_{i_1} s_{i_2} \ldots$ aus der Menge $(s_0 s_1 s_3 s_5)^\omega$.
   Gib an, ob für die folgenden LTL-Formeln f jeweils $M, \pi \models f$ und allgemeiner: $M \models f$ gilt.

f	$M, \pi \models f$	$M \models f$ $\mathrm{A}f$
$\bigcirc(\alpha_3 \vee \alpha_1)$	**Lösung:** Ja	**Lösung:** Ja
$\Box \alpha_2$	**Lösung:** Nein	**Lösung:** Nein
$\Box(\alpha_1 \Longrightarrow \bigcirc \alpha_2)$	**Lösung:** Nein	**Lösung:** Nein
$\Box((\alpha_1 \wedge \neg\alpha_2 \wedge \neg\alpha_3) \Longrightarrow \bigcirc(\alpha_2 \vee \alpha_3))$	**Lösung:** Ja	**Lösung:** Ja
$\Box\Diamond\alpha_3$	**Lösung:** Ja	**Lösung:** Nein
$\Box((\neg\alpha_1)\ \mathbf{U}\ \alpha_1)$	**Lösung:** Ja	**Lösung:** Ja

Satz 1.36 *Zu jeder LTL-Formel f kann eine Kripke-Struktur (ein Büchi-Automat) M konstruiert werden, die genau die für f gültigen Folgen akzeptiert: $L^\omega(M) = L^\omega(f)$.*

Die Zeit- und Platz-Komplexität dieses Algorithmus ist $2^{\mathcal{O}(|f|)}$

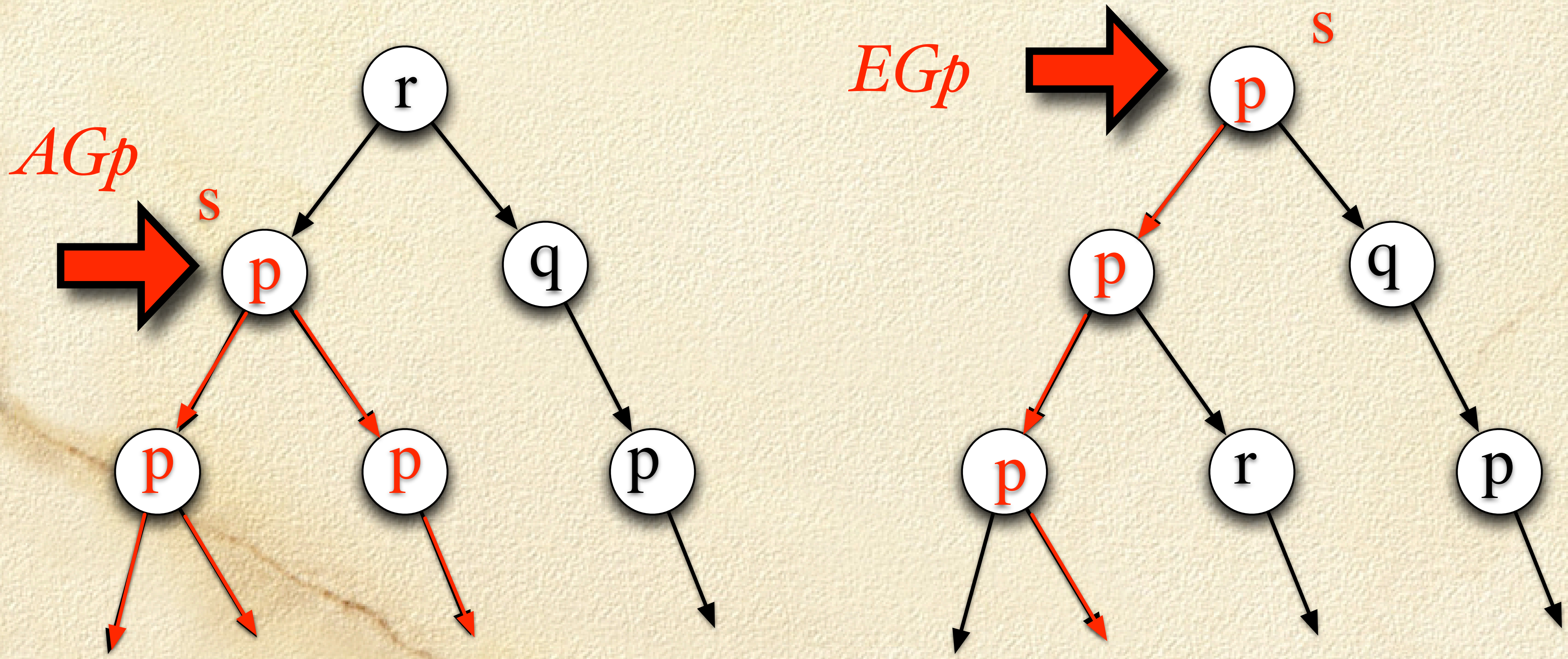


1.5.2 Syntax und Semantik von CTL- und CTL*-Formeln

CTL besitzt zusätzlich *Zustandsformeln* mit Quantoren, die sich auf alle von einem Zustand ausgehenden Pfade beziehen.

Ap „für alle Pfade, die von einem gegeben Zustand s ausgehen, gilt die Formel p“,

Ep „es gibt einen Pfad, der von einem gegeben Zustand s ausgeht und für den die Formel p gilt“.



Diese können alle mittels EXg, EGg, $E[g_1 U g_2]$ ausgedrückt werden:

Satz 1.40 *Es gelten die folgenden Äquivalenzen:*

- $AXg \Leftrightarrow \neg EX(\neg g)$,
- $EFg \Leftrightarrow E(\mathbf{true}\, U g)$,
- $AGg \Leftrightarrow \neg EF(\neg g)$,
- $AFg \Leftrightarrow \neg EG(\neg g)$,
- $A[g_1 U g_2] \Leftrightarrow \neg E[\neg g_2 U (\neg g_1 \wedge \neg g_2)] \wedge \neg EG \neg g_2$

Präsenzaufgabe 5.1:

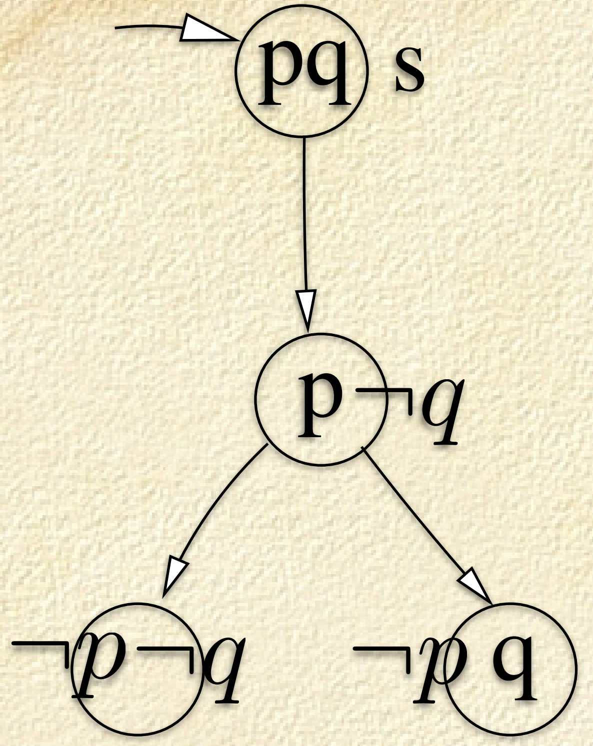
Aufgabe 1.41 Gegeben sind die folgenden Kripke-Strukturen M_1 und M_2:

LTL:

A (pq ∧ X(p ∧ (X¬q ∨ Xq))

pq, p¬q, ¬p¬q, ...

pq, p¬q, ¬p q, ...

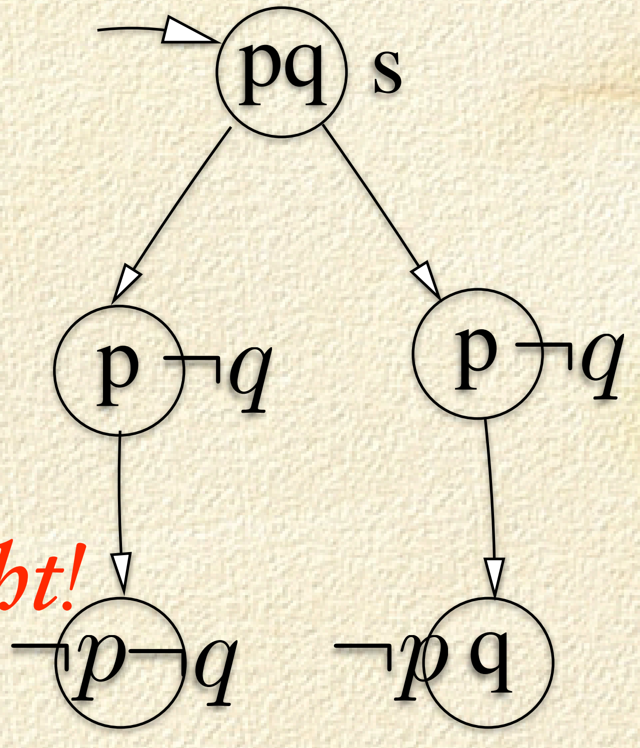


LTL:

A (pq ∧ X(p ∧ (X¬q ∨ Xq))

pq, p¬q, ¬p¬q, ...

pq, p¬q, ¬p q, ...



gilt hier nicht!

CTL:

pq ∧ AX(p ∧ EX¬q ∧ EXq)

CTL:

pq ∧ AX(p ∧ EX¬q ∧ EXq)

$\mathbf{AXEX}q$

(a) M_1

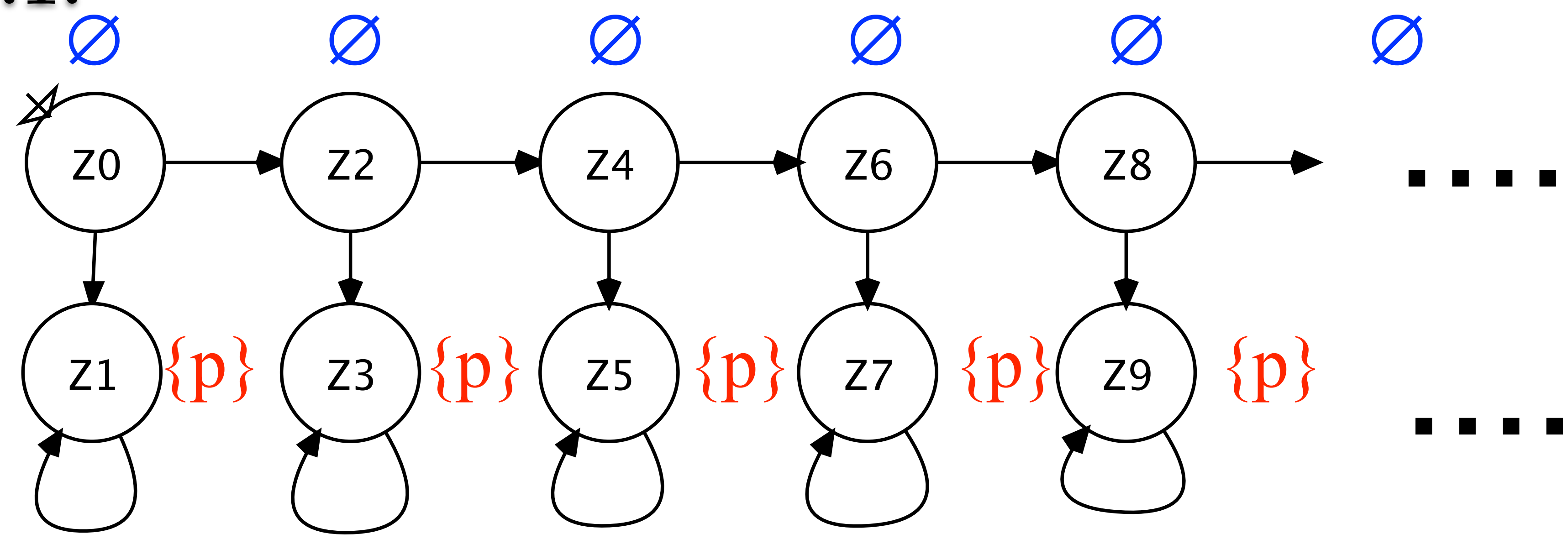
(b) M_2

schon mal gesehen?

LTL: *folgenäquivalent*

CTL: *bisimilar*

Präsenzaufgabe 5.1:



(a) Gilt $f_1 = \mathbf{EF}p$?

(a) $f_1 = \mathbf{EF}p$ gilt, denn $\pi = z_0 z_1 z_1 \cdots$ ist eine Abwicklung, die $\mathbf{F}p$ erfüllt.

(b) Gilt $f_2 = \mathbf{AGEF}p$?

$M, z \models \mathbf{EF}p$ für alle Zustände z.

(c) Gilt $f_3 = \mathbf{AF}p$?

(c) $f_3 = \mathbf{AF}p$ gilt nicht, denn p gilt nirgends auf dem Pfad $z_0 z_2 z_4 z_6 \cdots$

LTL: F p   A F p

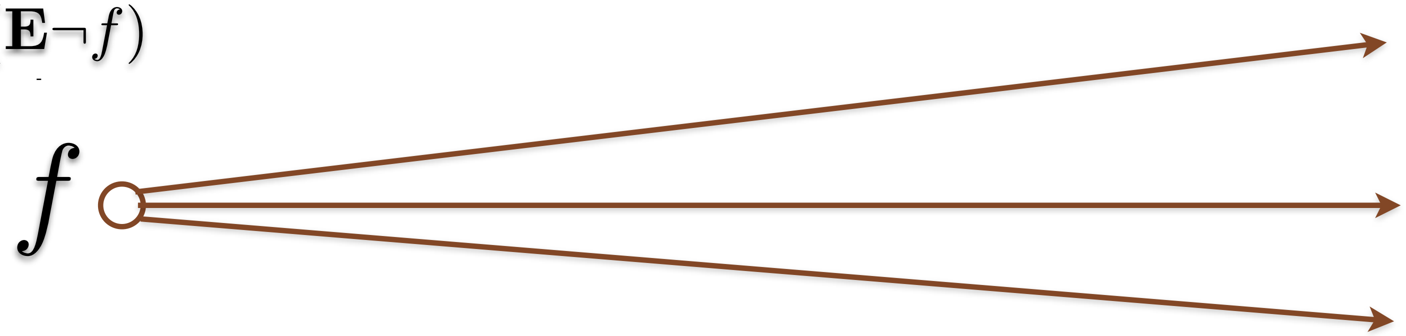
LTL: p ⇒ Fp   A (p ⇒ Fp)

Präsenzaufgabe 5.2: Äquivalenzen in CTL.

1. Formulieren Sie die folgenden Äquivalenzen in natürlicher Sprache und begründen Sie deren Gültigkeit: (i) $\neg\mathbf{G}f \equiv \mathbf{F}(\neg f)$,

(ii) $\mathbf{F}f \equiv \mathit{True}\,\mathbf{U}f$

(iii) $\mathbf{A}f \equiv \neg(\mathbf{E}\neg f)$



(iv) $\neg\mathbf{X}f \equiv \mathbf{X}\neg f$

2. Beweisen Sie die Äquivalenzen:

$$
\begin{aligned}
\mathbf{AX}f &\equiv \neg\mathbf{EX}(\neg f) \\
\mathbf{EF}f &\equiv \mathbf{E}[True\,\mathbf{U}f] \\
\mathbf{AG}f &\equiv \neg\mathbf{EF}(\neg f) \\
\mathbf{AF}f &\equiv \neg\mathbf{EG}(\neg f)
\end{aligned}
$$

(a) Mit (iii) und (iv) gilt $\mathbf{AX}f \equiv \neg\mathbf{E}(\neg\mathbf{X}f) \equiv \neg\mathbf{EX}(\neg f)$.

(b) Mit (ii) gilt $\mathbf{EF}f \equiv \mathbf{E}[True\,\mathbf{U}f]$.

(c) Mit (iii) und (i) gilt $\mathbf{AG}f \equiv \neg\mathbf{E}(\neg\mathbf{G}f) \equiv \neg\mathbf{EF}(\neg f)$.

(d) Mit (iii) $\neg\mathbf{G}f \equiv \mathbf{F}(\neg f)$ gilt $\mathbf{AF}f \equiv \neg\mathbf{E}(\neg\mathbf{G}f) \equiv \neg\mathbf{EG}(\neg f)$.

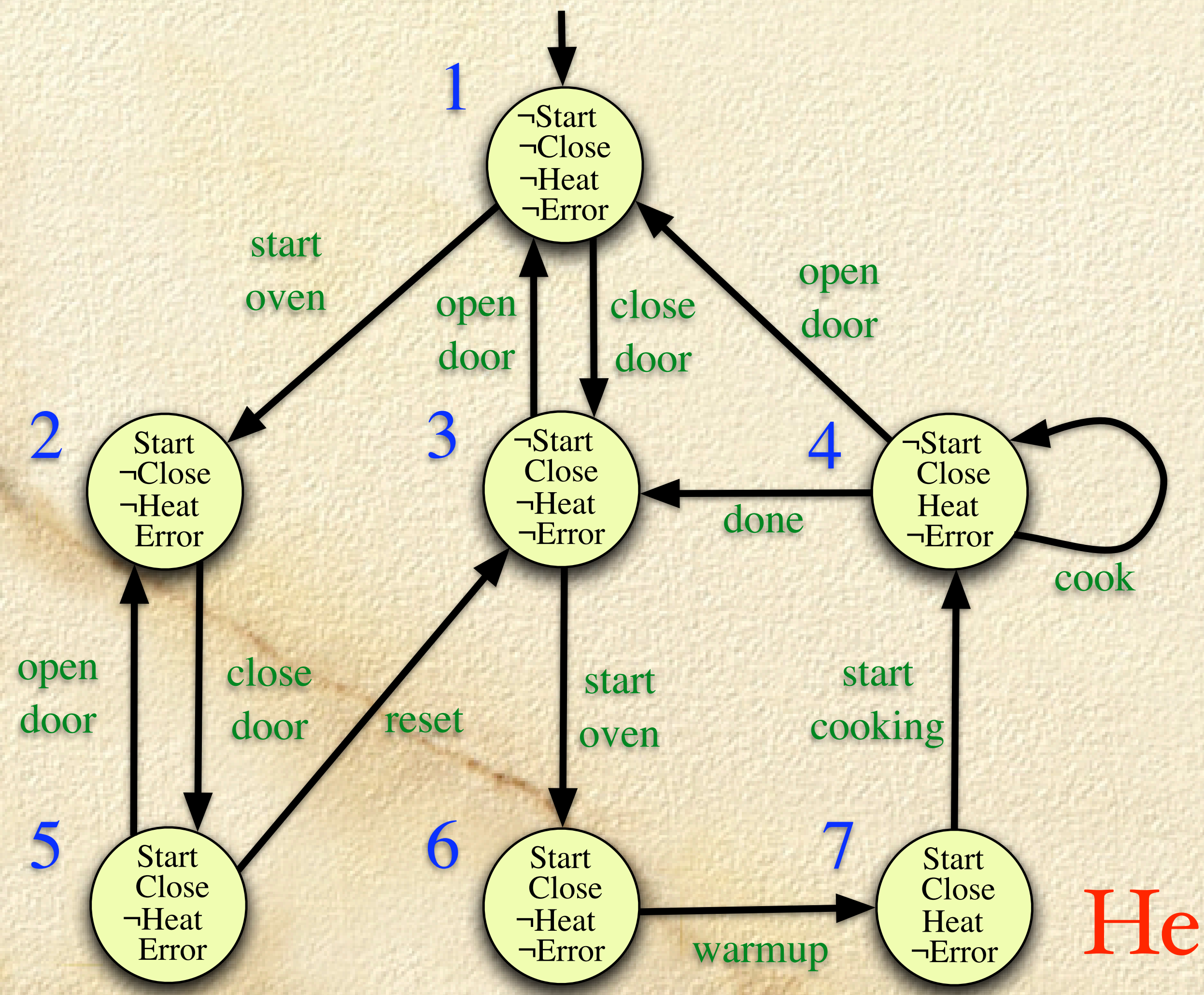
Algorithmus für CTL-Formeln f.

Erweitere $E_S(s)$ für alle $s \in S$ schrittweise zu $label(s)$. Das wird dann die Menge der Teilformeln von f, die in s wahr sind.

Rekursion über die Schachtelungstiefe von f

6 zu entwickelnde Prozeduren

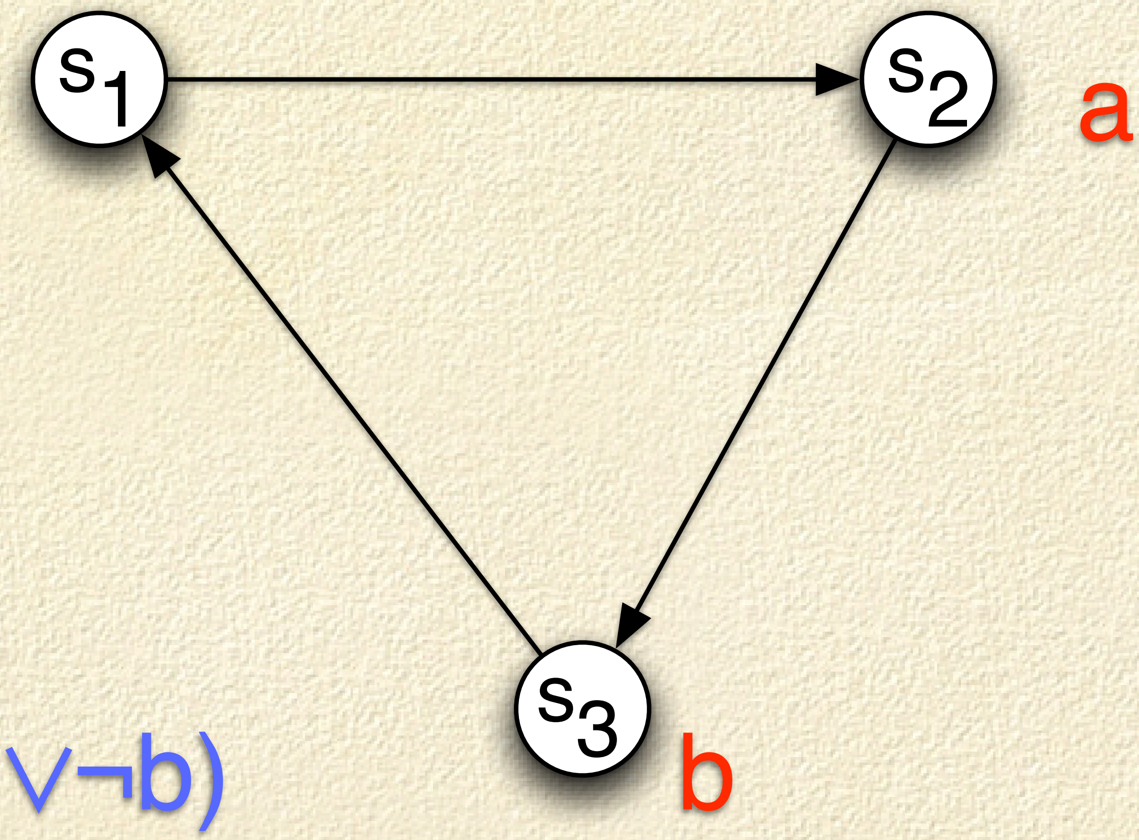
1. $f \in AP$ ist atomar
2. $f = \neg f_1$
3. $f = f_1 \vee f_2$
4. $f = EX\,f_1$
5. $f = E[f_1 U f_2]$
6. $f = EG\,f_1$



$f =$
$\neg\text{Start} \Rightarrow F(\boxed{\text{Heat} \wedge \neg\text{Error}})$

Heat ∧ ¬Error

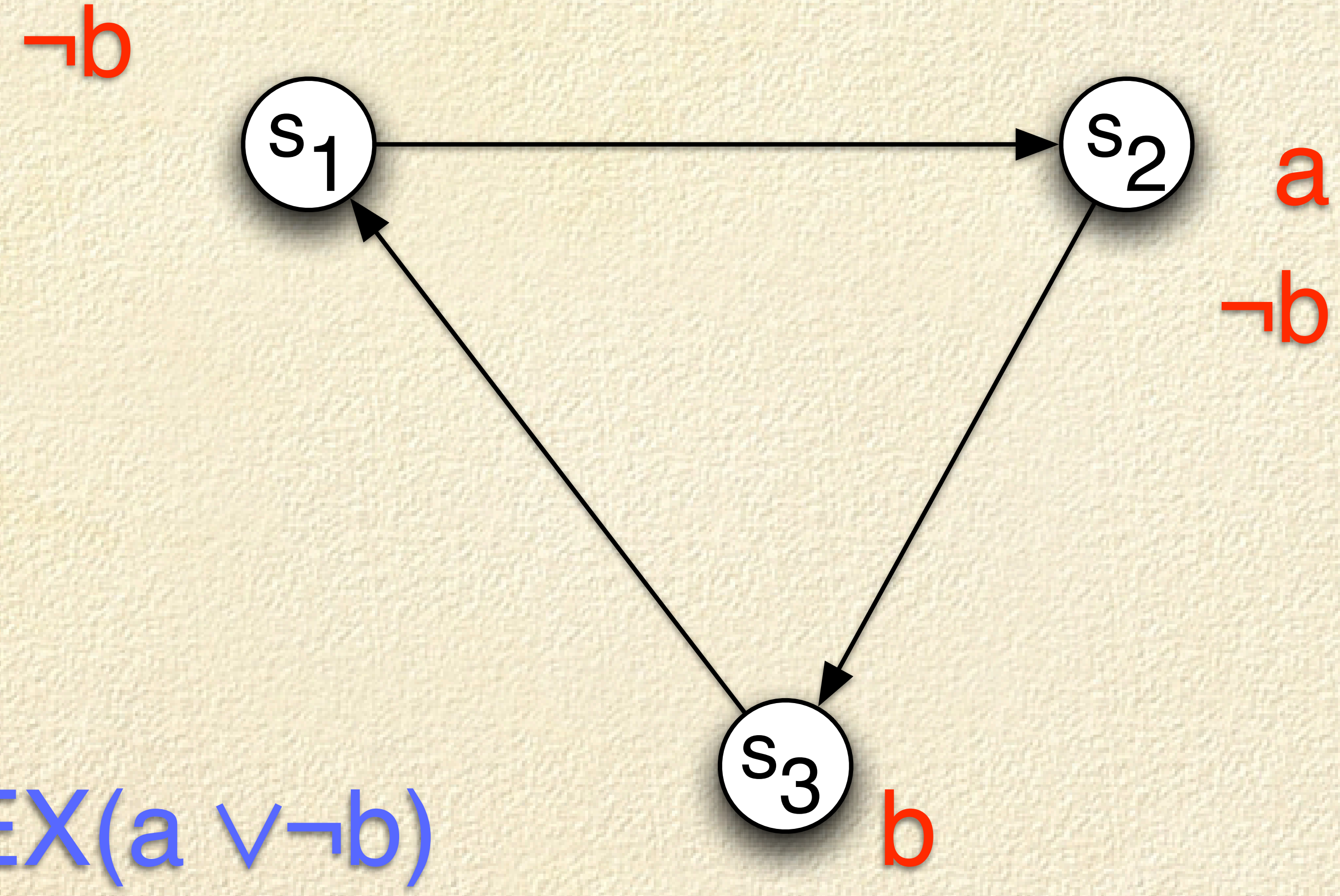
1. f atomar:
für Zustände s mit $f \in L(s)$ setzte $f \in label(s)$.



f = EX(a ∨¬b)

1. f atomar:
   für Zustände s mit $f \in L(s)$ setzte $f \in label(s)$.

2. $f = \neg f_1$:
   für Zustände s mit $f_1 \notin label(s)$ setzte $\neg f_1 \in label(s)$.

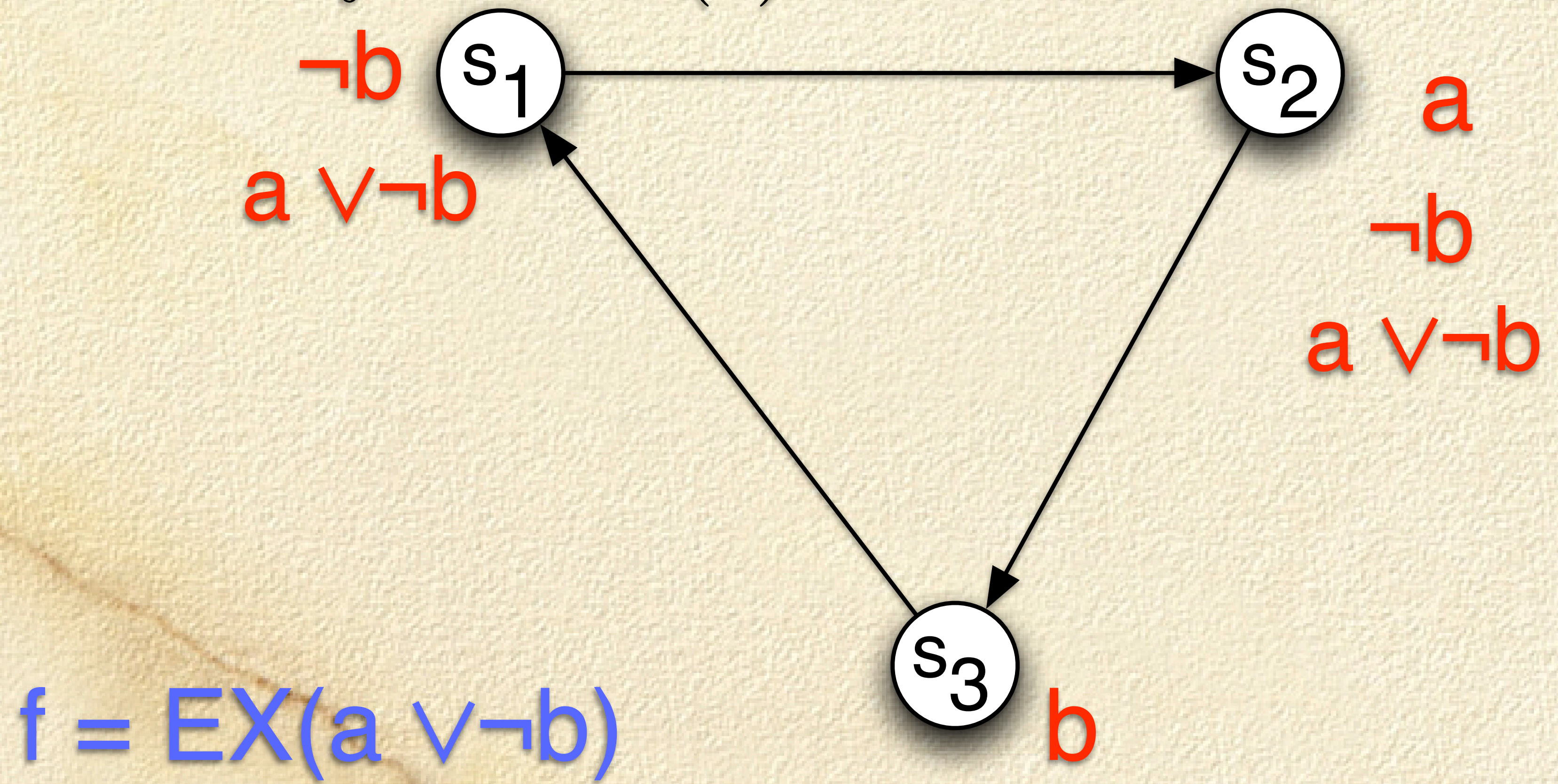


f = EX(a ∨¬b)

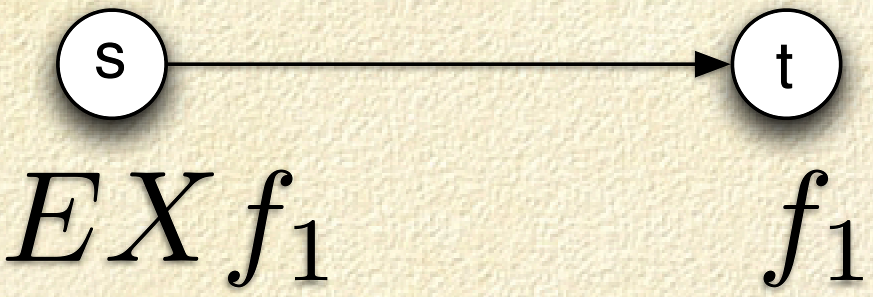
1. f atomar:
   für Zustände s mit $f \in L(s)$ setzte $f \in label(s)$.

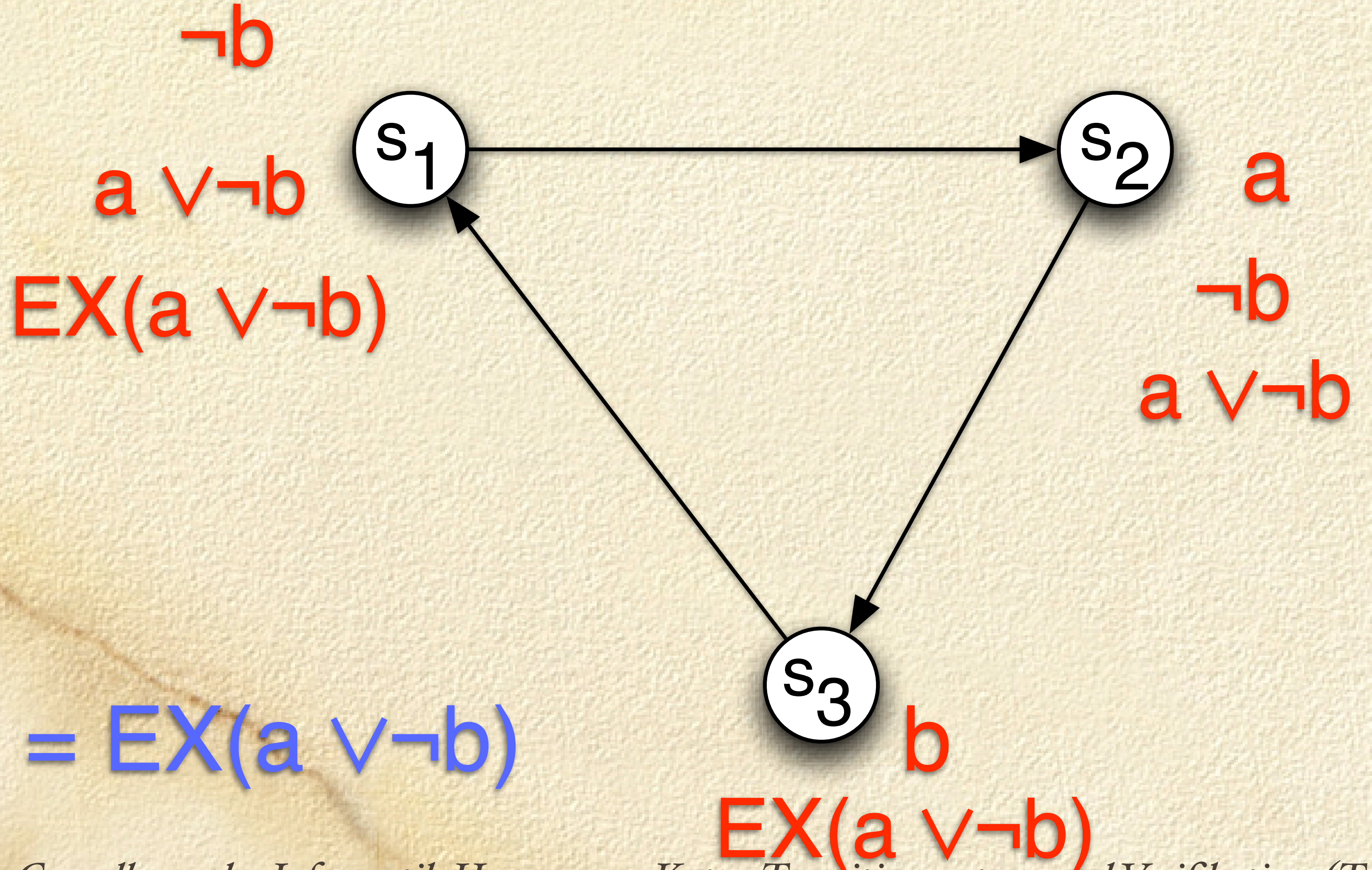
2. $f = \neg f_1$:
   für Zustände s mit $f_1 \notin label(s)$ setzte $\neg f_1 \in label(s)$.

3. $f = f_1 \vee f_2$:
   für Zustände s mit $f_1 \in label(s)$ oder $f_2 \in label(s)$ setzte $f \in label(s)$.



4. $f = EX f_1$:
für Zustände s mit $R(s,t)$ und $f_1 \in label(t)$ setzte $f \in label(s)$.

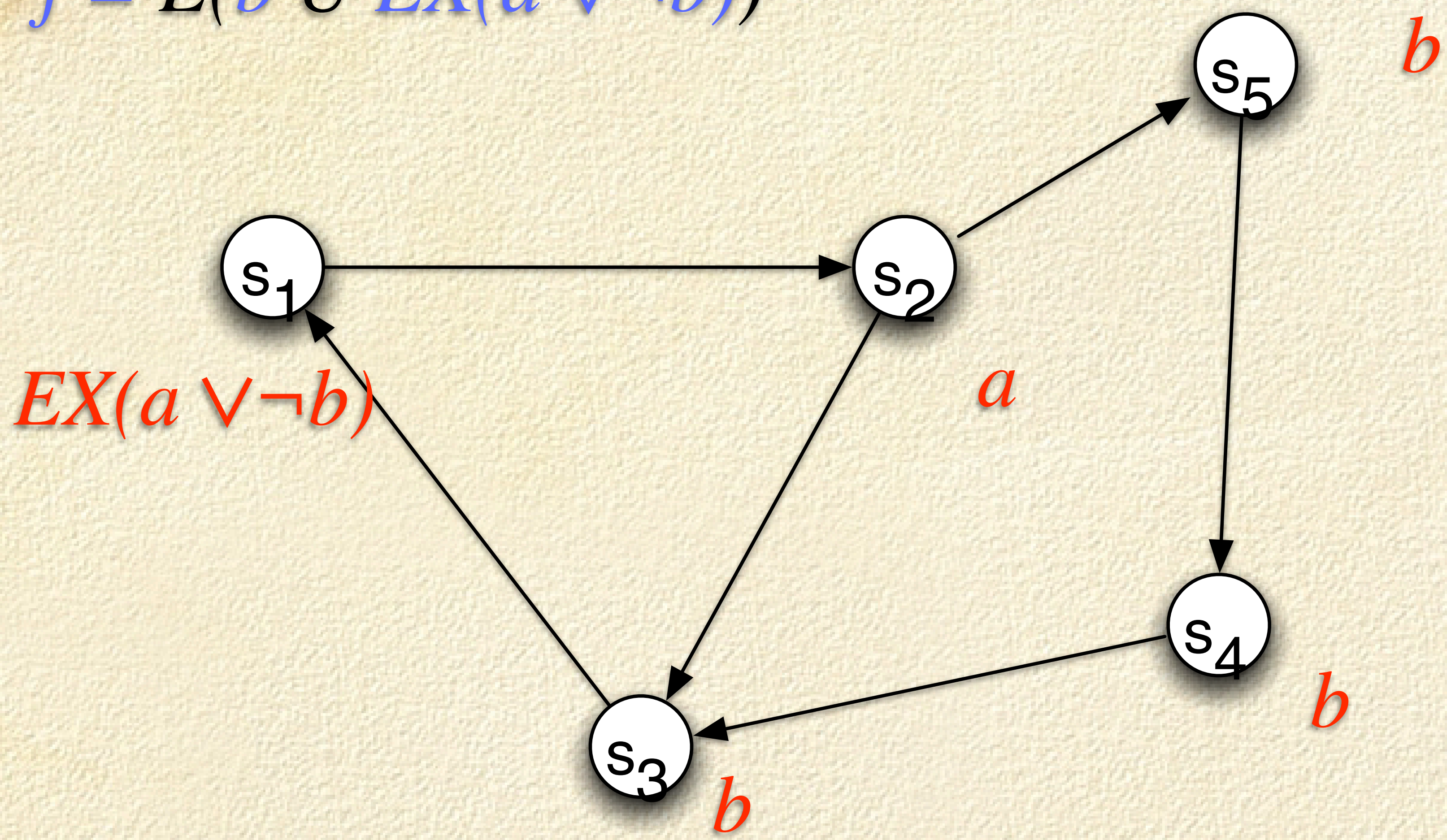




f = EX(a ∨¬b)

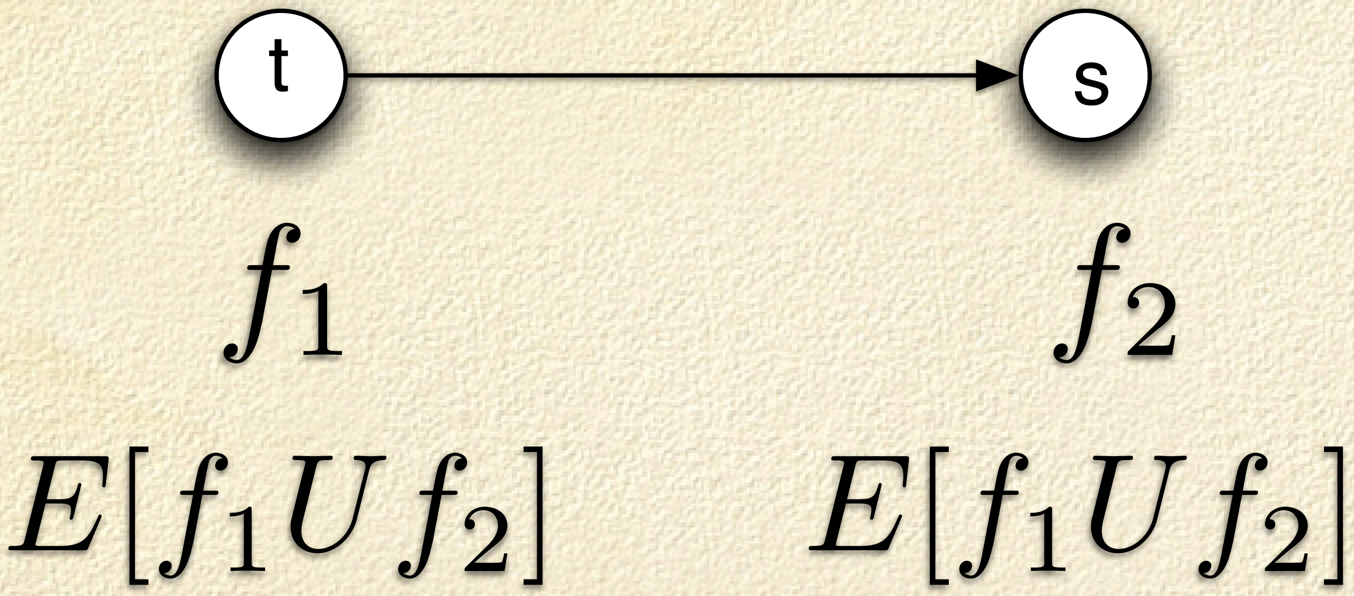
5. $f = E[f_1 U f_2]$:

$f = E(b\ U\ EX(a \vee \neg b))$



5. $f = E[f_1 U f_2]$:

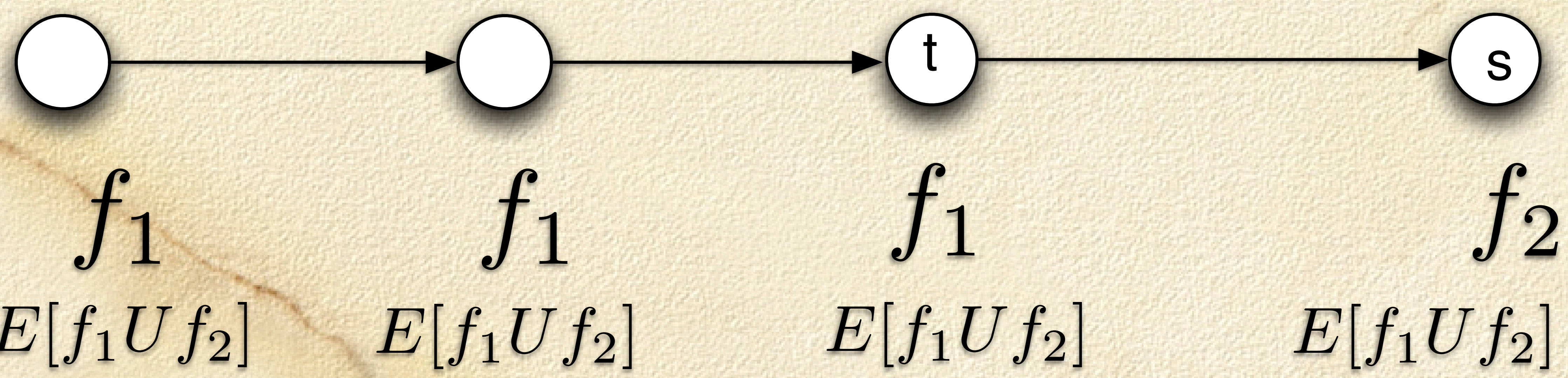
für Zustände s mit $f_2 \in label(s)$ setzte $f \in label(s)$;
für Zustände t mit $R(t, s)$ und $f_1 \in label(s)$ setze $f \in label(t)$;



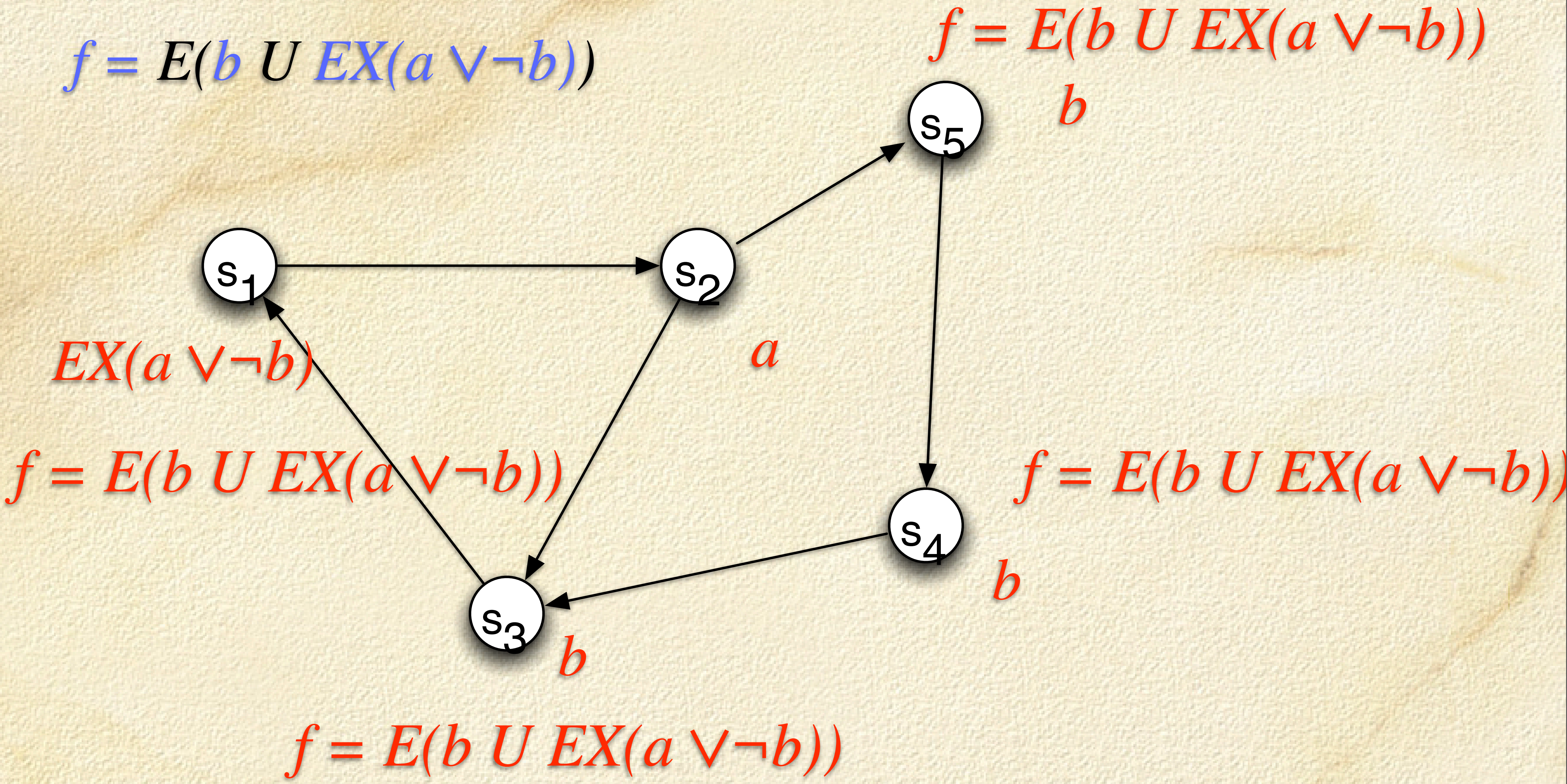
5. $f = E[f_1 U f_2]$:

für Zustände s mit $f_2 \in label(s)$ setzte $f \in label(s)$;
für Zustände t mit $R(t,s)$ und $f_1 \in label(s)$ setze $f \in label(t)$;

fahre schrittweise in Gegenrichtung der Transitionen fort und setze $f \in label(s)$, falls es einen Pfad von s zu einem s' mit $f_2 \in label(s')$ gibt, auf dem für alle Zustände t davor $f_1 \in label(t)$ gilt. Siehe Algorithmus 5.5.

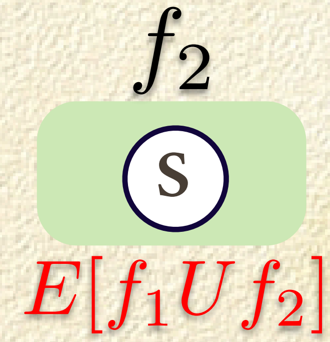


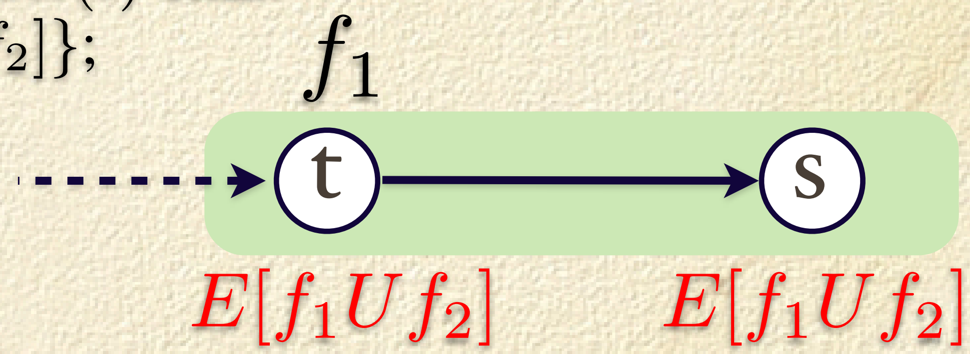
$f = E(b\ U\ EX(a \vee \neg b))$



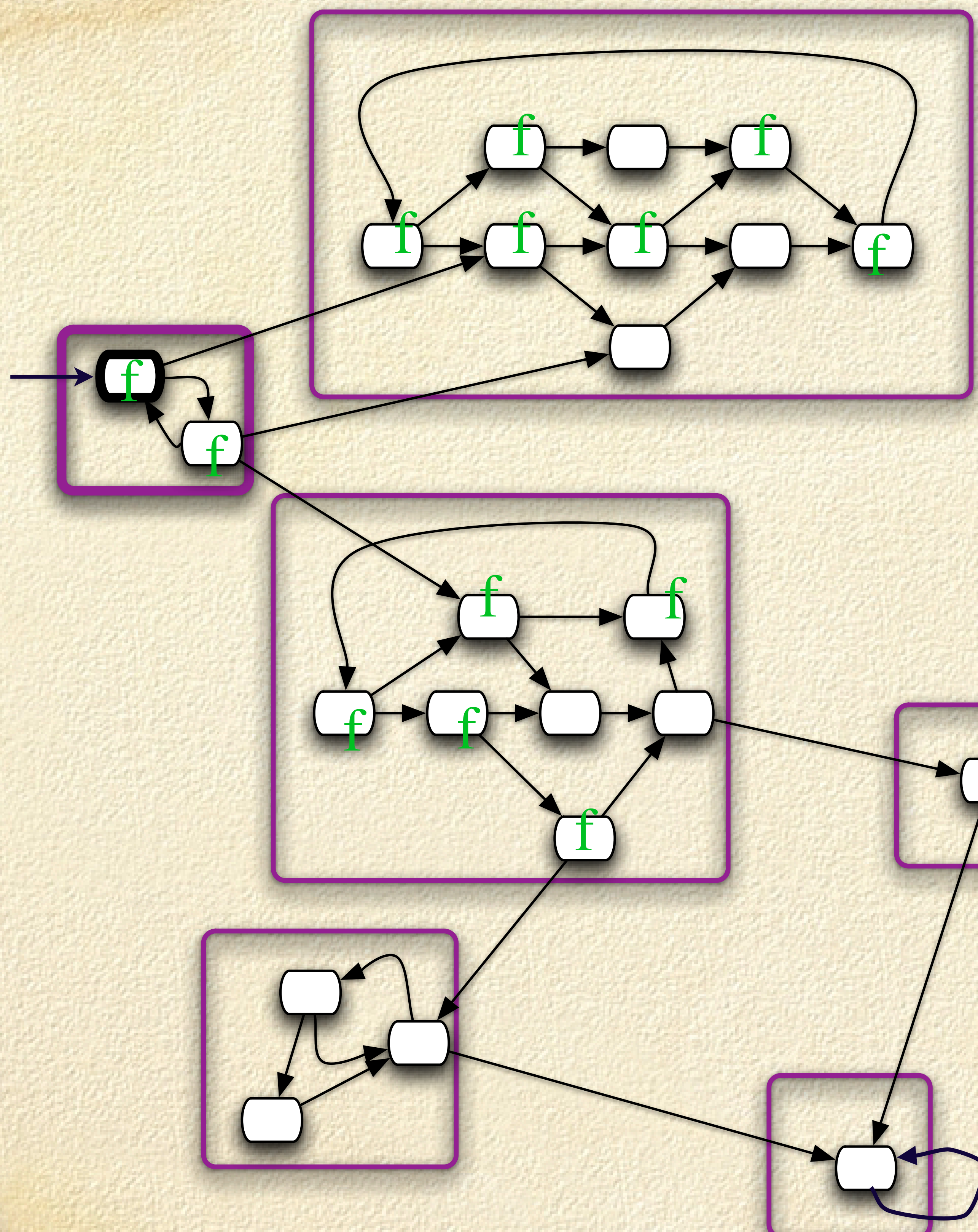
Algorithmus 1.6 Auszeichnen mit $E(f_1 U f_2)$

PROCEDURE $CheckEU(f_1, f_2)$
    $T := \{s \mid f_2 \in label(s)\};$
    FOR ALL $s \in T$ DO $label(s) := label(s) \cup \{E[f_1 U f_2]\};$
    WHILE $T \neq \emptyset$ DO
        CHOOSE $s \in T;$
        $T := T \setminus \{s\};$
        FOR ALL t SUCH THAT $R(t, s)$ DO
            IF $E[f_1 U f_2] \notin label(t)$ AND $f_1 \in label(t)$ THEN
                $label(t) := label(t) \cup \{E[f_1 U f_2]\};$
                $T := T \cup \{t\};$
            END IF ;
        END FOR ALL ;
    END WHILE ;
END PROCEDURE ;

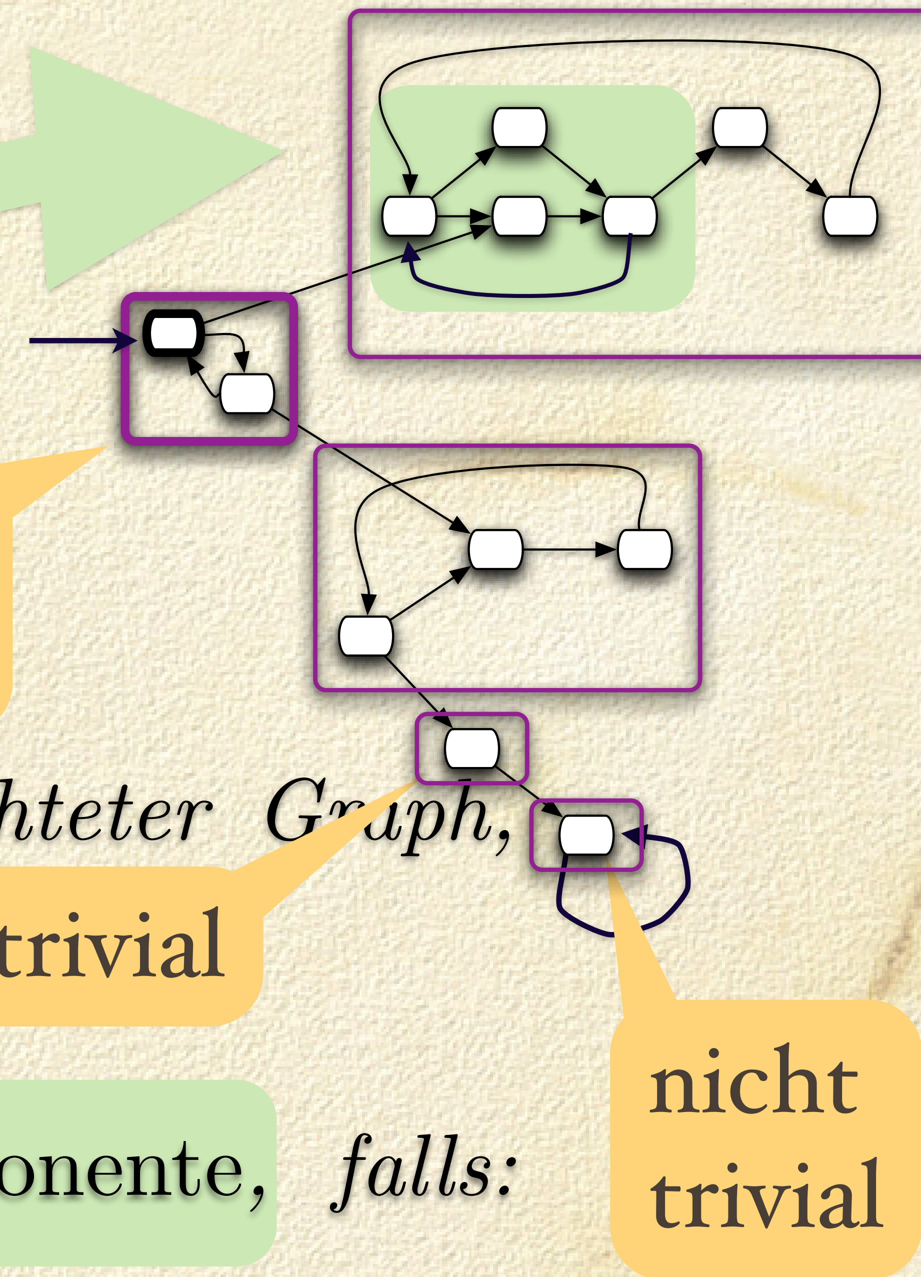




EG f ??



EG f ??



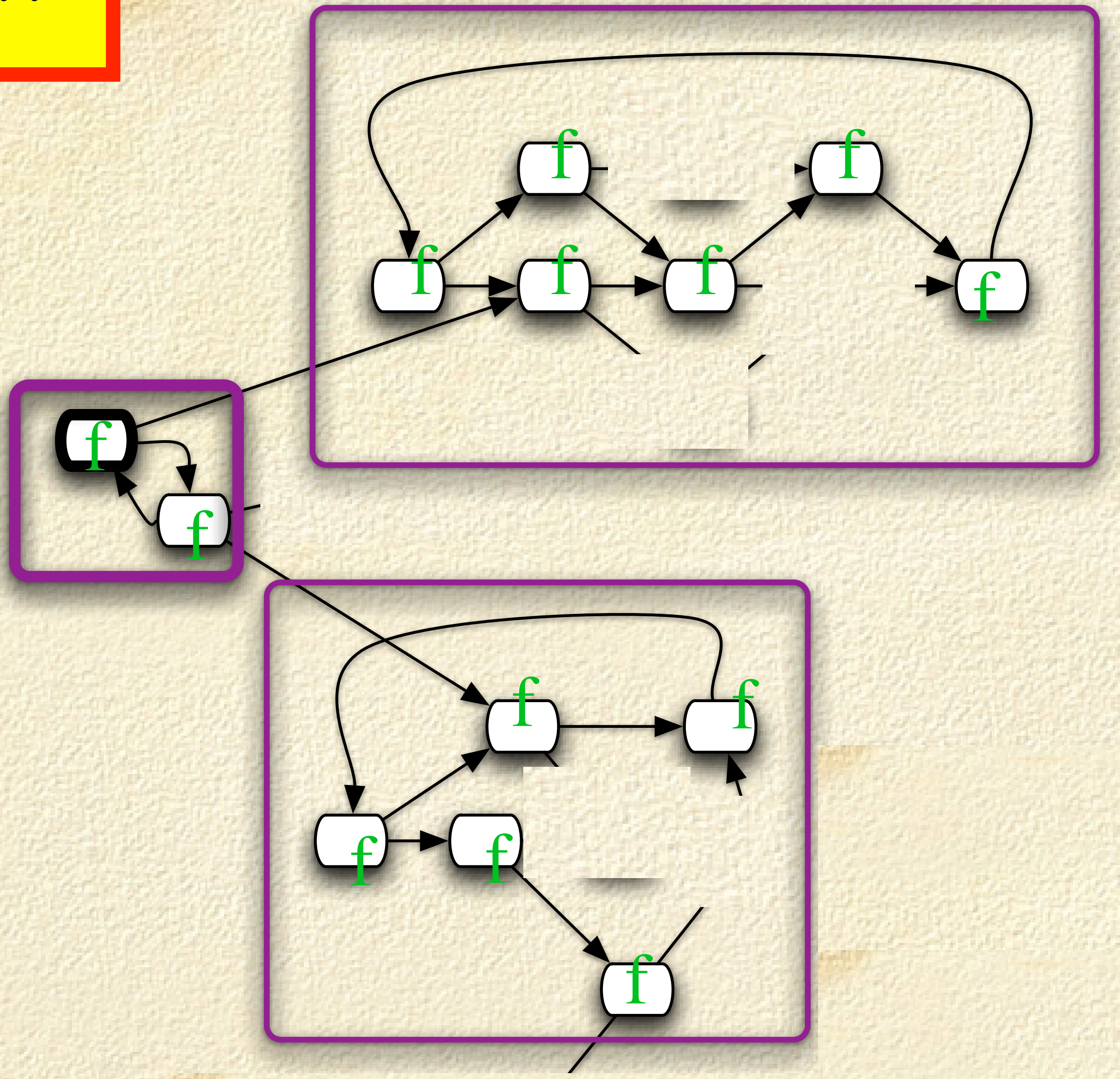
Definition 1.44 *Sei* $G = (K, R)$ *ein gerichteter Graph, d.h.:* $R \subseteq K \times K$:

a) $A \subseteq K$ *heißt* Zusammenhangskomponente, *falls:* $\forall a, a' \in A : aR^*a'$.

b) *Sie heißt* strenge Zusammenhangskomponente (SZK) *(strongly connected component: SZK), falls sie maximal ist, d.h.:* $\neg\exists k \in K \backslash A \,.\, \forall a \in A : kR^*a \wedge aR^*k$.

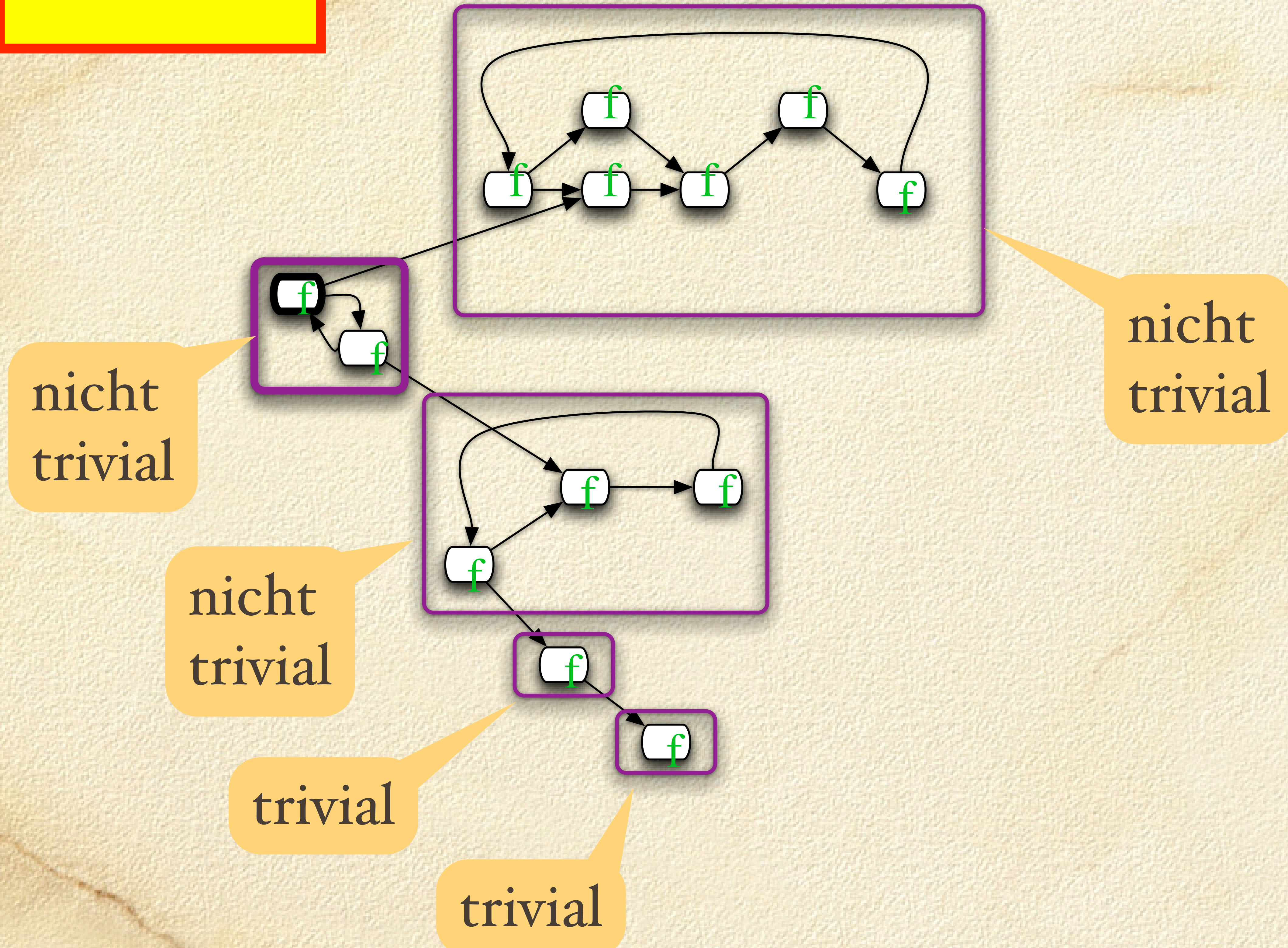
c) *Sie heißt* nichttriviale Zusammenhangskomponente, *falls:* $|A| > 1$ *oder* $\exists a \in A \,.\, aR^+a$.

EG f ??



SZK

EG f ??



f
f
f
f
f
f
f
f
f
f
f
f
f
f
nicht trivial
nicht trivial
nicht trivial
trivial
trivial

Nun betrachten wir wieder die Formel: $f = EG f_1$:
Aus $M = (S, R, L)$ konstruiere $M' = (S', R', L')$ mit:

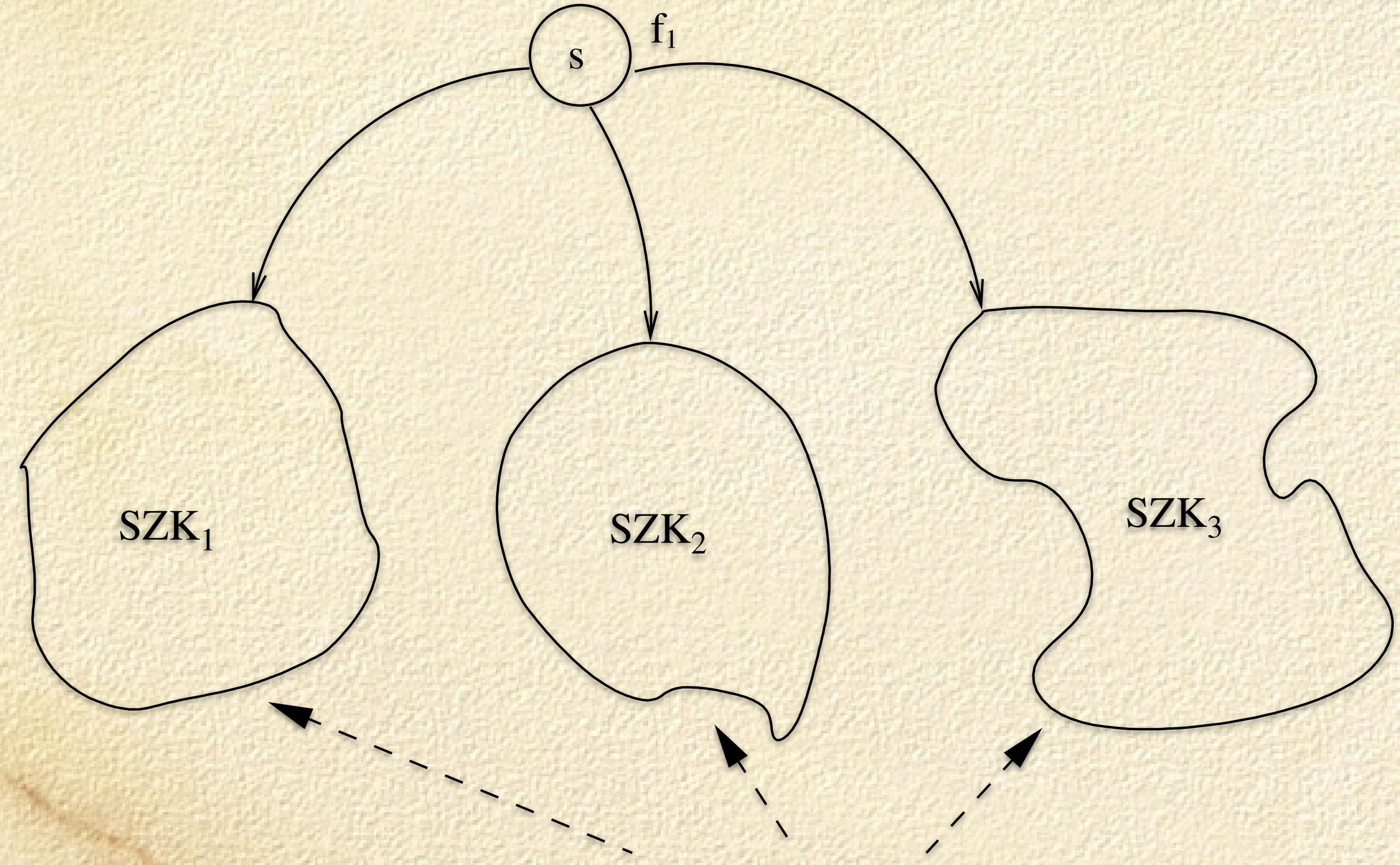
$$\begin{aligned} S' &= \{s \in S \mid M, s \models f_1\} \\ R' &= R_{|S' \times S'} \\ L' &= L_{|S'} \end{aligned}$$

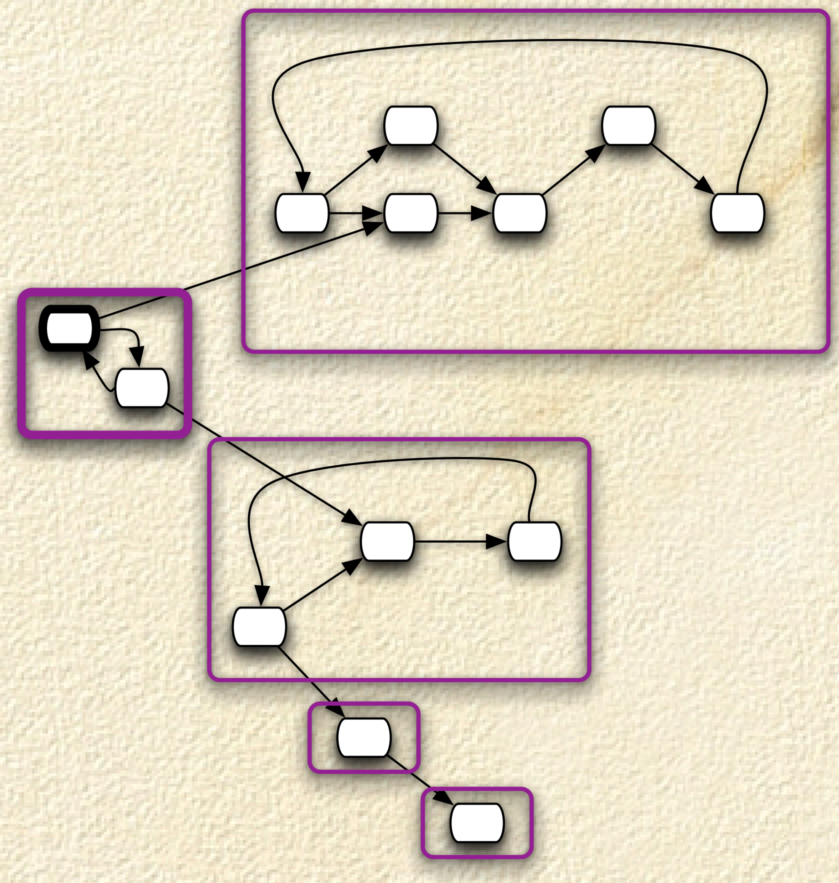
d.h. die „Einschränkung" von M, in der f_1 gilt.

Lemma 1.45 $M, s \models EG f_1$ *gdw.*

1. $s \in S'$

2. *Es gibt einen Pfad in* M', *der von* s *zu einer nicht-trivialen starken Zusamenhangskomponente in* (S', R') *führt.*





Daraus Algorithmus zur Entscheidung von EGf_1:

1. Konstruiere $M' = (S', R', L')$
2. Konstruiere alle SZK von M'. (Algorithmus von Tarjan mit $O(|S'| + |R'|)$ Zeitkomplexität).
3. Finde Zustände in nichttrivialen SZK.
4. Suche von diesen rückwärts alle Zustände die dorthin führen.

   insgesamt: $O(|S| + |R|)$. Siehe Algorithmus 5.6.

Algorithmus 1.7 Auszeichnen mit EGf_1

Seite 178

PROCEDURE $CheckEGf_1$

$S' := \{s \mid f_1 \in label(s)\};$

$SCC := \{C \mid C \text{ a nontrivial } SCC \text{ of } S'\};$

$T := \bigcup_{C \in SCC} \{s \mid s \in C\};$

FOR ALL $s \in T$ **DO** $label(s) := label(s) \cup \{EGf_1\};$

WHILE $T \neq \emptyset$ **DO**

  CHOOSE $s \in T;$

  $T := T \setminus \{s\};$

  FOR ALL t **SUCH THAT** $t \in S'$ **AND** $R(t, s)$ **DO**

    IF $EGf_1 \notin label(t)$ **THEN**

      $label(t) := label(t) \cup \{EGf_1\};$

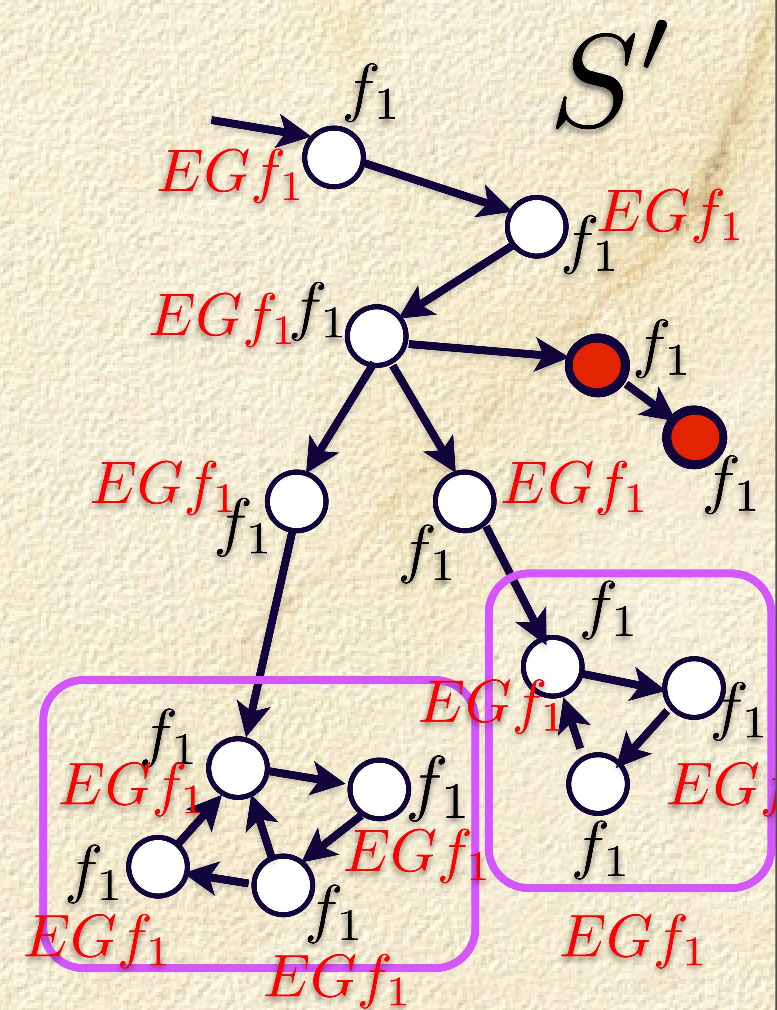
      $T := T \cup \{t\};$

    END IF ;

  END FOR ALL ;

END WHILE ;

END PROCEDURE ;

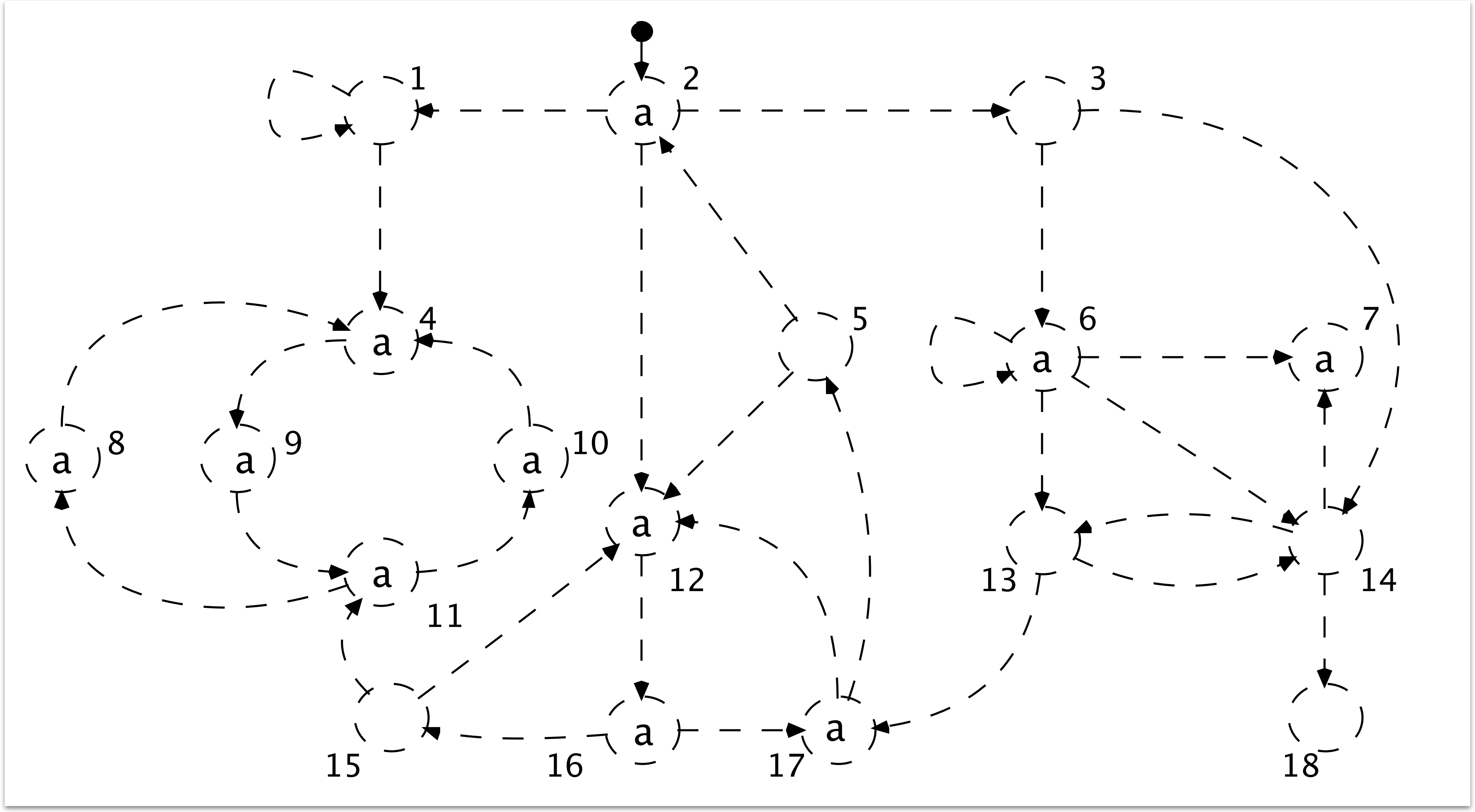


Satz 1.46 *Es gibt einen Algorithmus, der für eine Kripke-Struktur* $M := (S, S_0, R, E_S)$ *und eine CTL-Formel* f *in* $\mathcal{O}(|f| \cdot (|S| + |R|))$ *Zeitkomplexität entscheidet, ob* f *für* M *gilt, d.h. ob* $M \models f$.

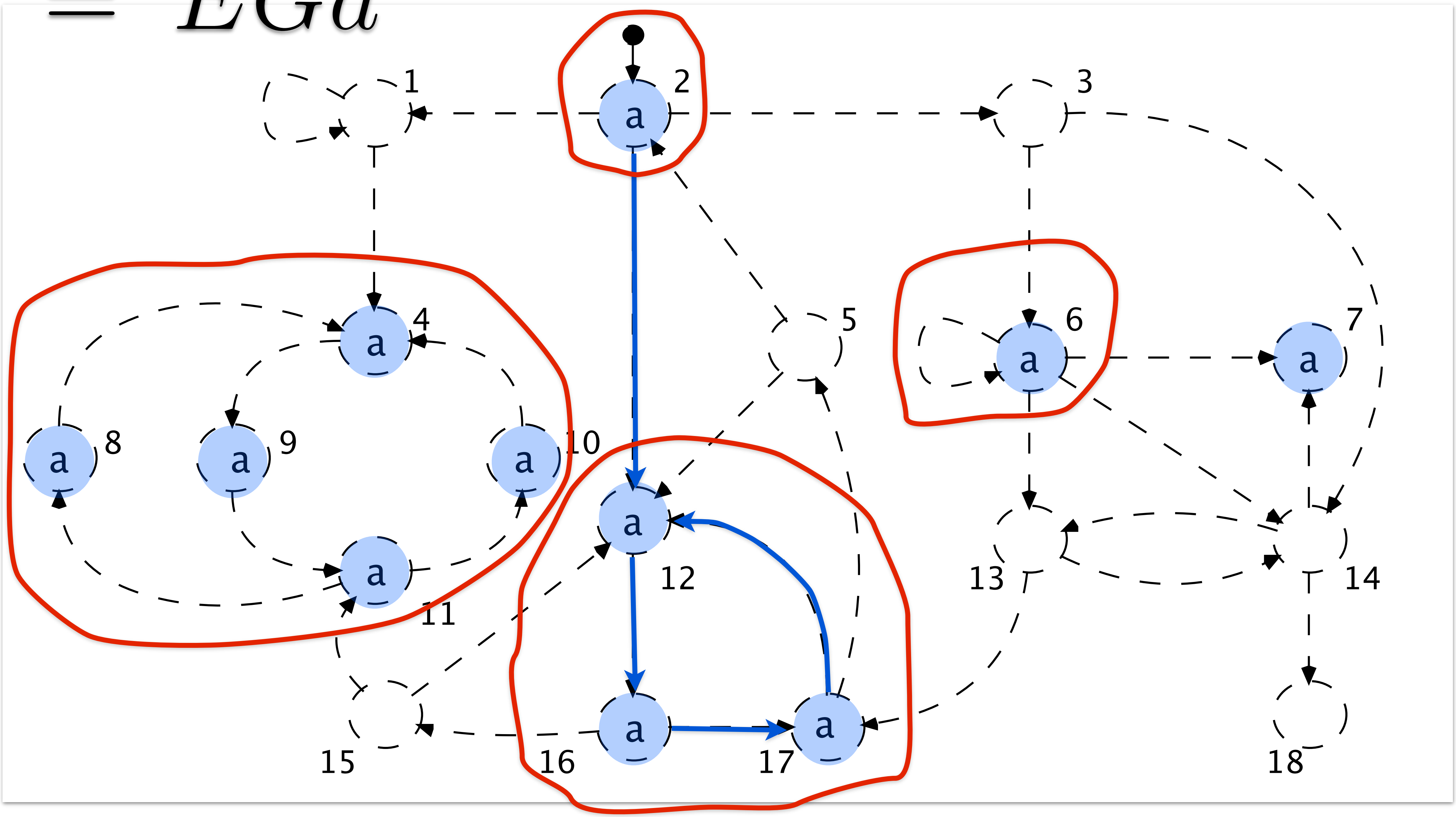
Beweis:
Wende obiges Verfahren auf die Atome von f an und fahre induktiv fort mit den Teilformeln von f, aufsteigend mit deren Schachtelung. Die Schachtelungstiefe ist durch $\mathcal{O}(|f|)$ begrenzt. Auf jeder Ebene gibt es maximal $\mathcal{O}(|f| \cdot (|S| + |R|))$ Operationen. $\square$

1. Wir wollen mit dem in der Vorlesung behandelten CTL-Algorithmus entscheiden, ob die Formel $f = EGa$ gilt. Dazu benötigen wir die Einschränkung von M auf die a erfüllenden Zustände. Heben Sie diese in der folgenden Abbildung hervor, bspw. indem Sie die übrigbleibenden Kanten und Knoten nachzeichnen. ....(4 Pkt.)

   Heben Sie auf dem resultierenden Ergebnis alle strengen Zusammenhangskomponenten hervor und kennzeichnen Sie die nicht-trivialen. ....(4 Pkt.)



$$f = EGa$$



2. Wenn f gilt, geben Sie einen Pfad an, der dies nachweist (als ω-regulären Ausdruck). Erläutern Sie, wie der CTL-Algorithmus diesen Pfad ermittelt. . . . . (4 Pkt.)

$\pi =$ $2\,(12,16,17)^{\omega}$

Wie viele nicht triviale, strenge Zusammenhangskomponenten (SZK)?