A survey of complexity results for temporal logics

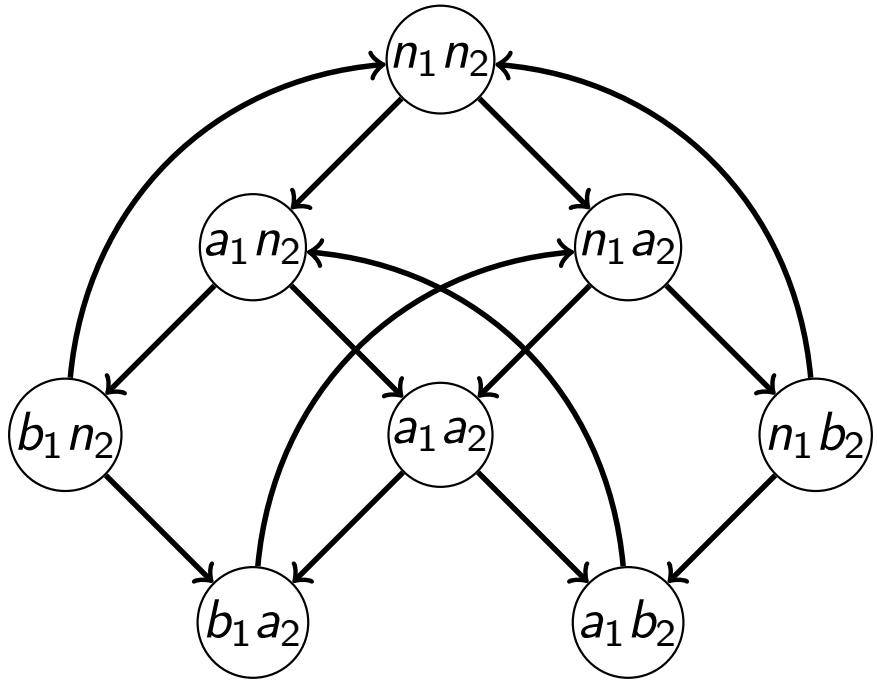
Martin Mundhenk

Institut für Informatik, Universität Jena

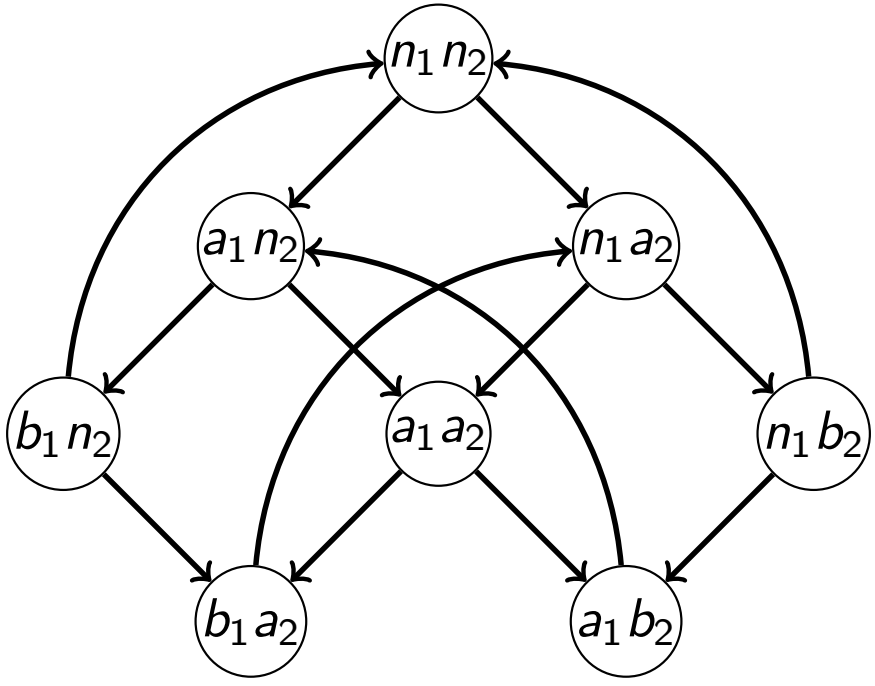
Standardbeispiel: mutual exclusion

Prozesse dürfen nicht gleichzeitig die selbe Ressource benutzen ...

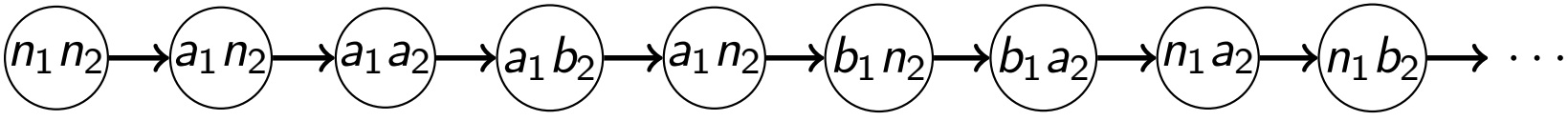
- n – Ressource wird weder angefordert noch benutzt
- a – Ressource wird angefordert (es wird auf Zugriff gewartet)
- b – Ressource wird benutzt



Temporale Operatoren



Ein Pfad durch das Modell:

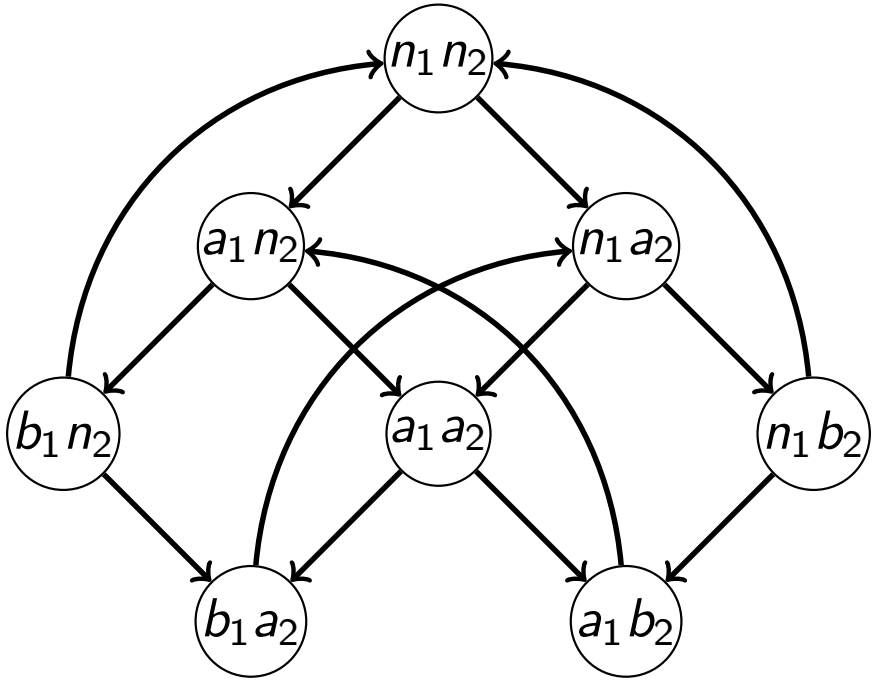


Eigenschaften des Pfades:

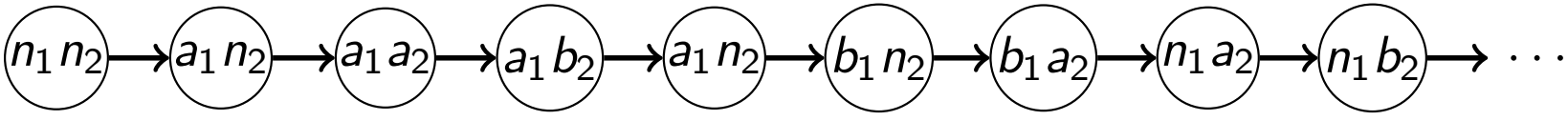
$$\mathsf{G}\ \neg(b_1 \wedge b_2)$$

G . . . globally

Temporale Operatoren



Ein Pfad durch das Modell:

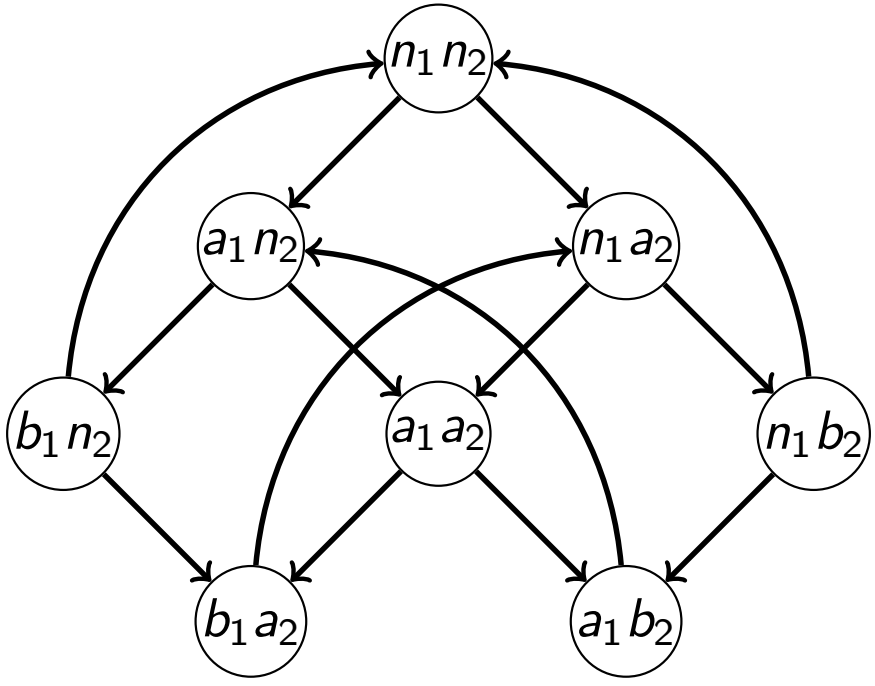


Eigenschaften des Pfades:

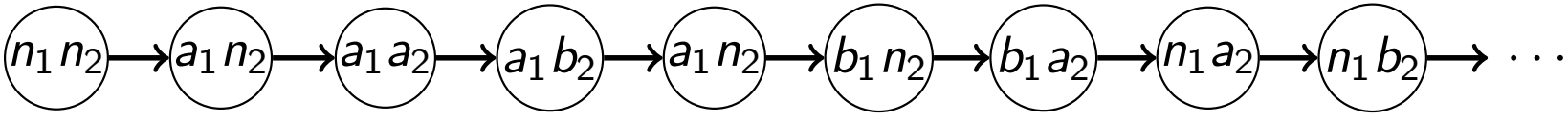
$$\mathrm{F}\,b_2$$

G . . . globally, F . . . future

Temporale Operatoren



Ein Pfad durch das Modell:

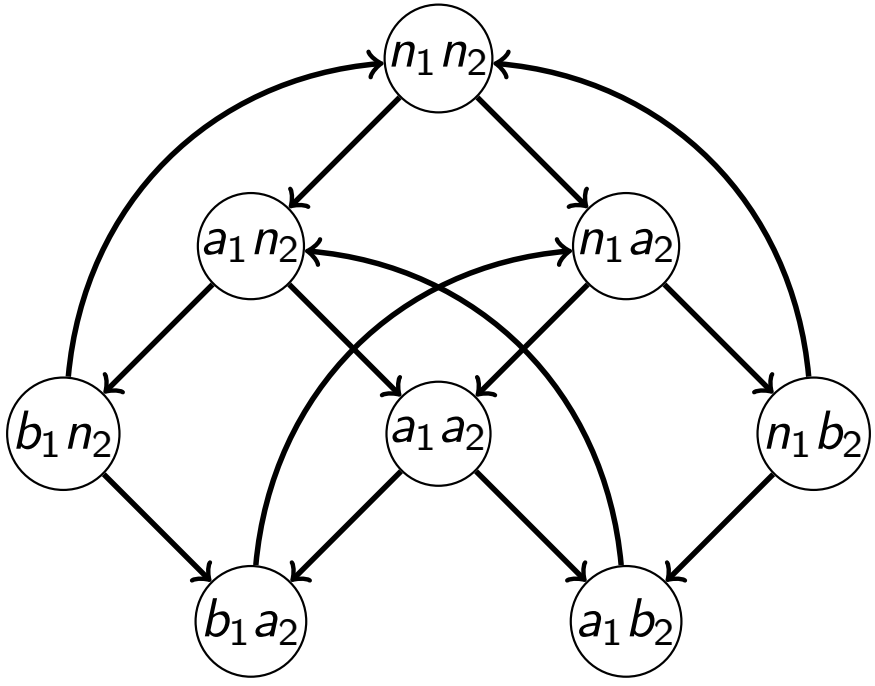


Eigenschaften des Pfades:

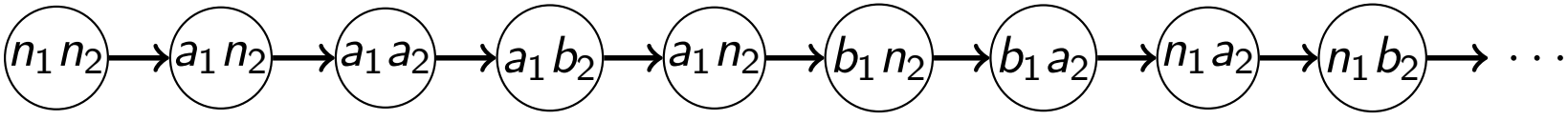
$$\mathrm{G}(a_2 \rightarrow \mathrm{F}\,b_2)$$

G . . . globally, F . . . future

Temporale Operatoren



Ein Pfad durch das Modell:

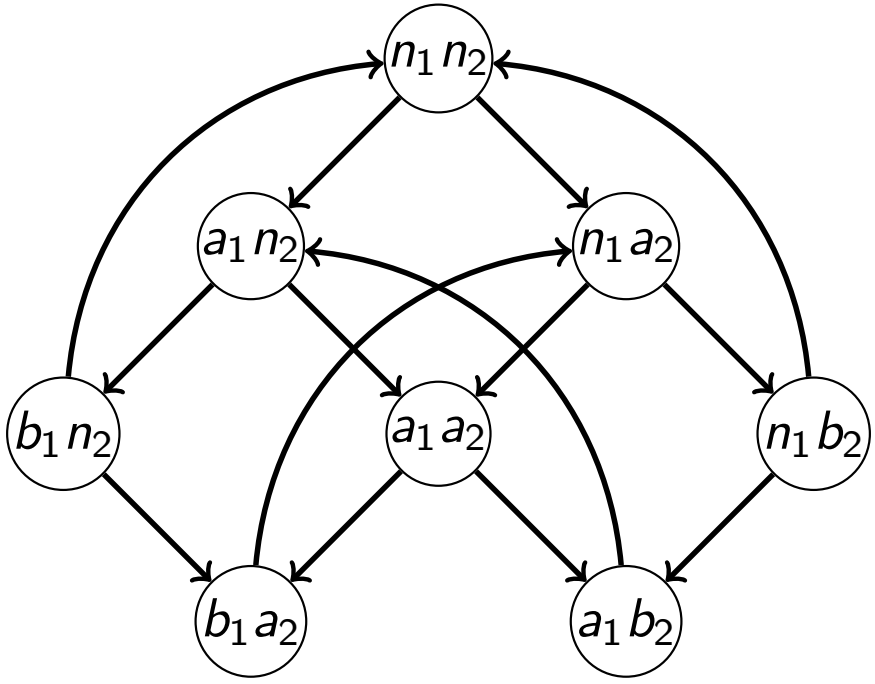


Eigenschaften des Pfades:

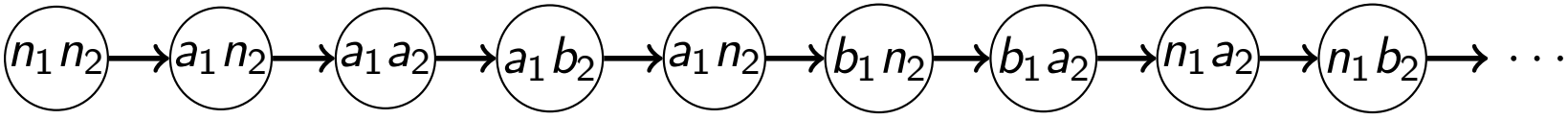
$$\mathsf{X}\,b_1$$

G . . . globally, F . . . future, X . . . next

Temporale Operatoren



Ein Pfad durch das Modell:

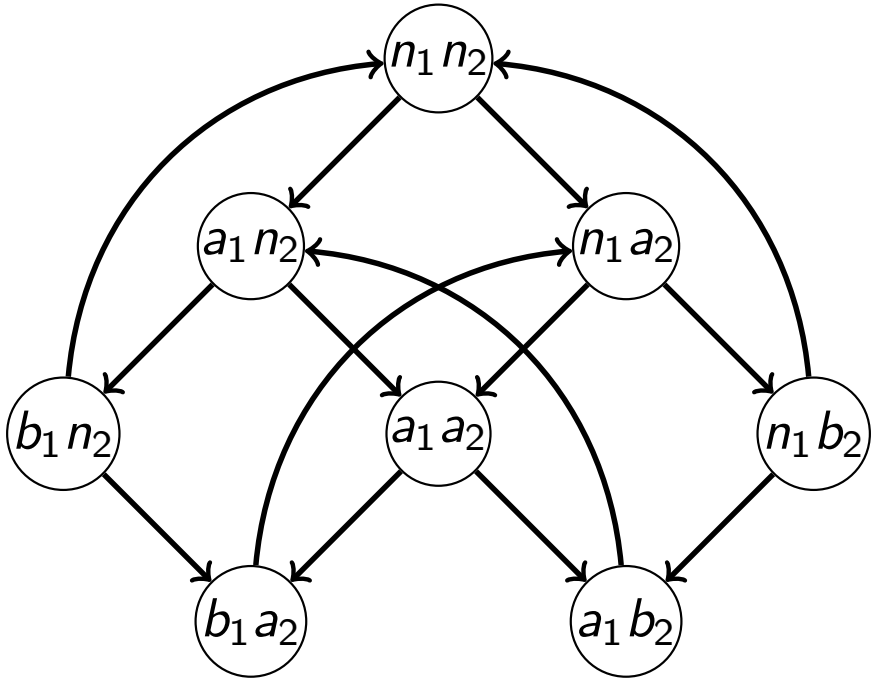


Eigenschaften des Pfades:

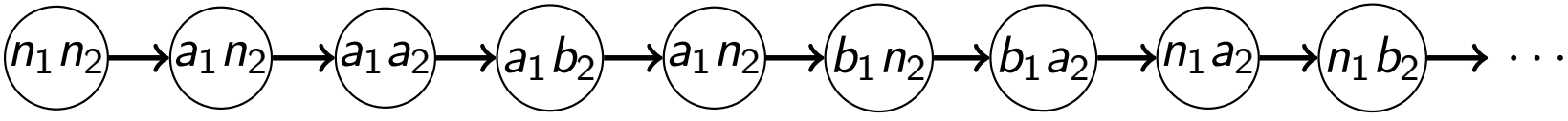
$$\mathsf{G}((a_1 \wedge \neg b_2) \rightarrow \mathsf{X}b_1)$$

G . . . globally, F . . . future, X . . . next

Temporale Operatoren



Ein Pfad durch das Modell:

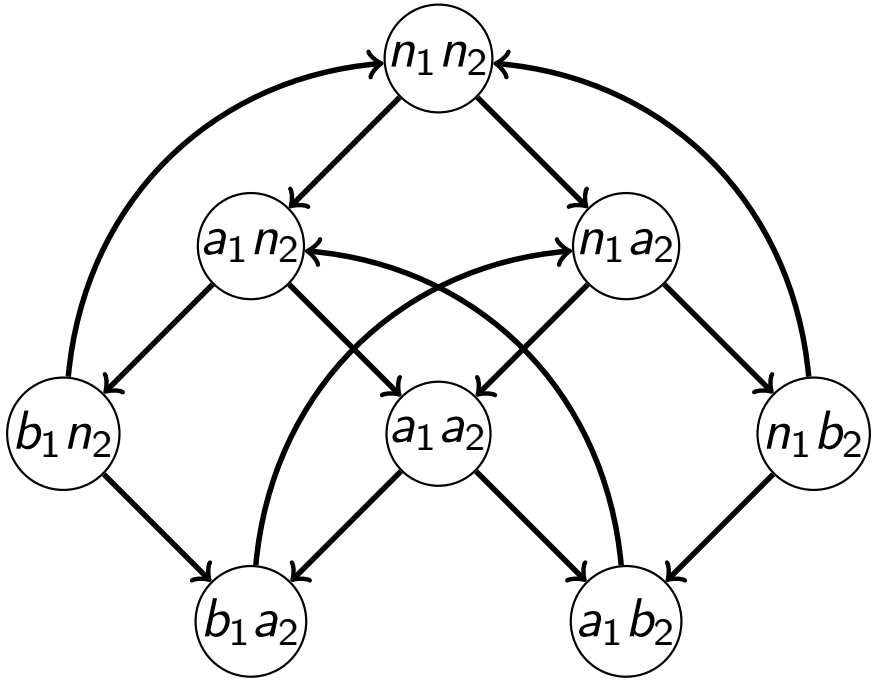


Eigenschaften des Pfades:

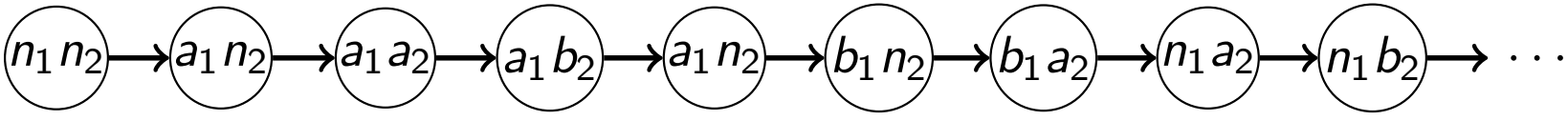
$$\neg \mathsf{G}((a_1 \wedge \neg b_2) \rightarrow \mathsf{X} b_1)$$

G . . . globally, F . . . future, X . . . next

Temporale Operatoren



Ein Pfad durch das Modell:

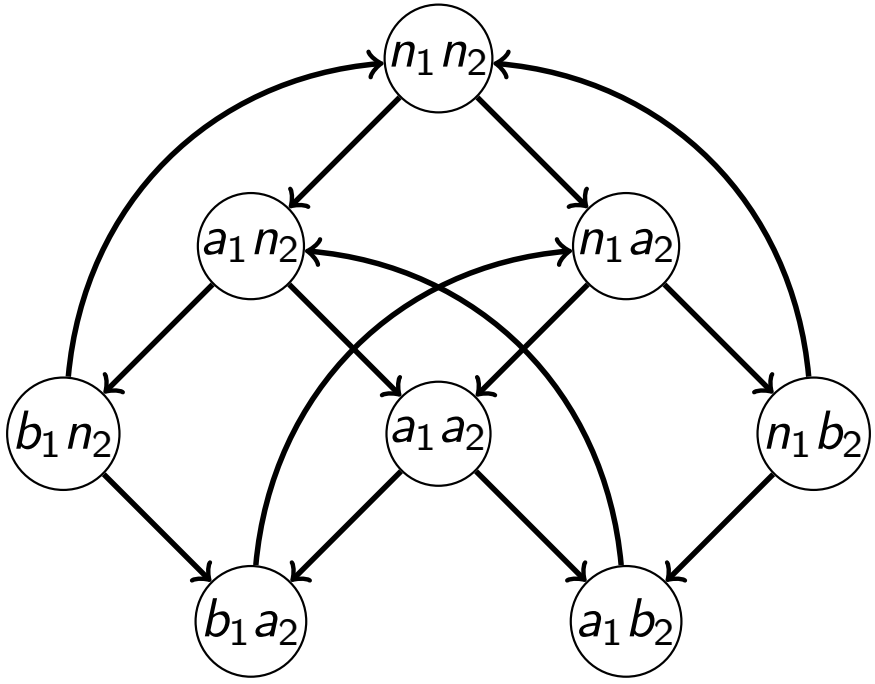


Eigenschaften des Pfades:

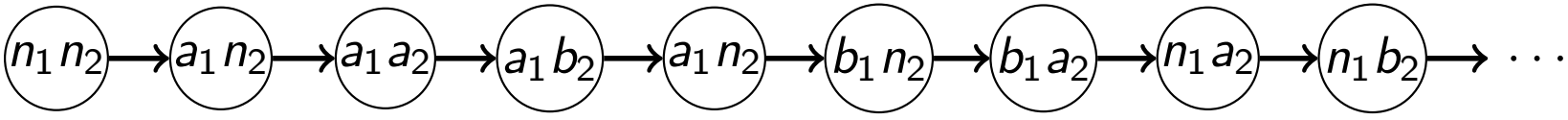
$$\neg \mathsf{G}((a_1 \wedge \neg b_2) \rightarrow \mathsf{X} b_1) \equiv \mathsf{F}\neg((a_1 \wedge \neg b_2) \rightarrow \mathsf{X} b_1)$$

G . . . globally, F . . . future, X . . . next

Temporale Operatoren



Ein Pfad durch das Modell:

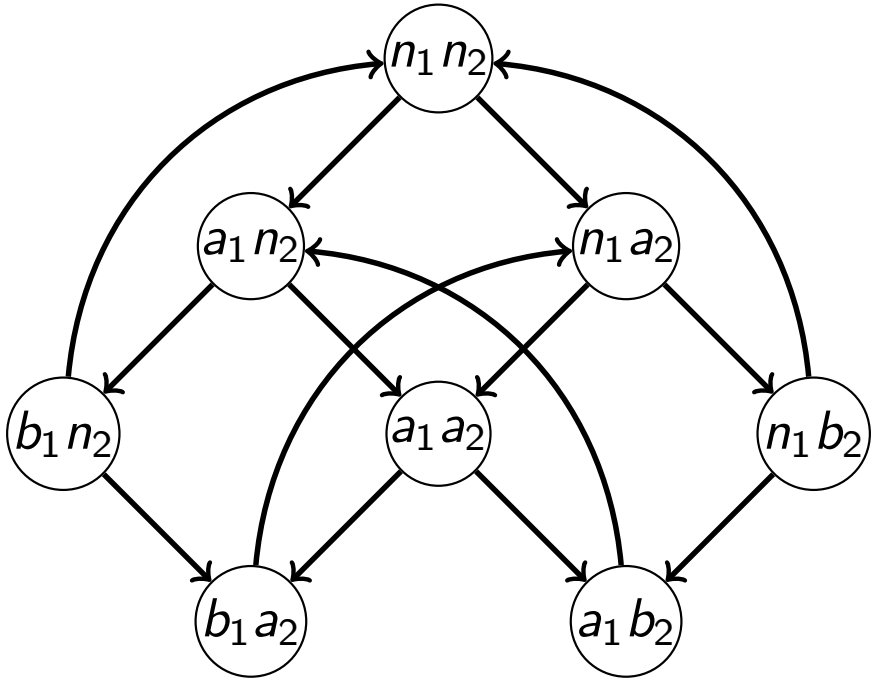


Eigenschaften des Pfades:

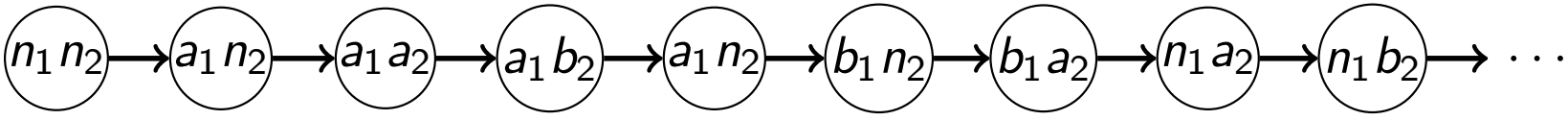
$$a_1 \mathrm{U} b_1$$

G . . . globally, F . . . future, X . . . next , U . . . until

Temporale Operatoren



Ein Pfad durch das Modell:

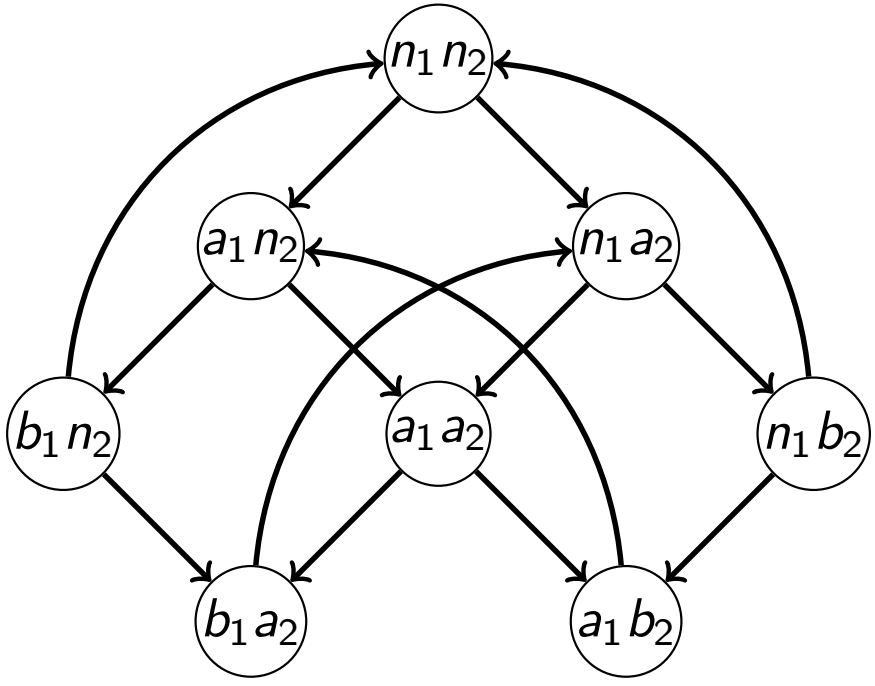


Eigenschaften des Pfades:

$$\mathrm{X}(a_1 \mathrm{U} b_1)$$

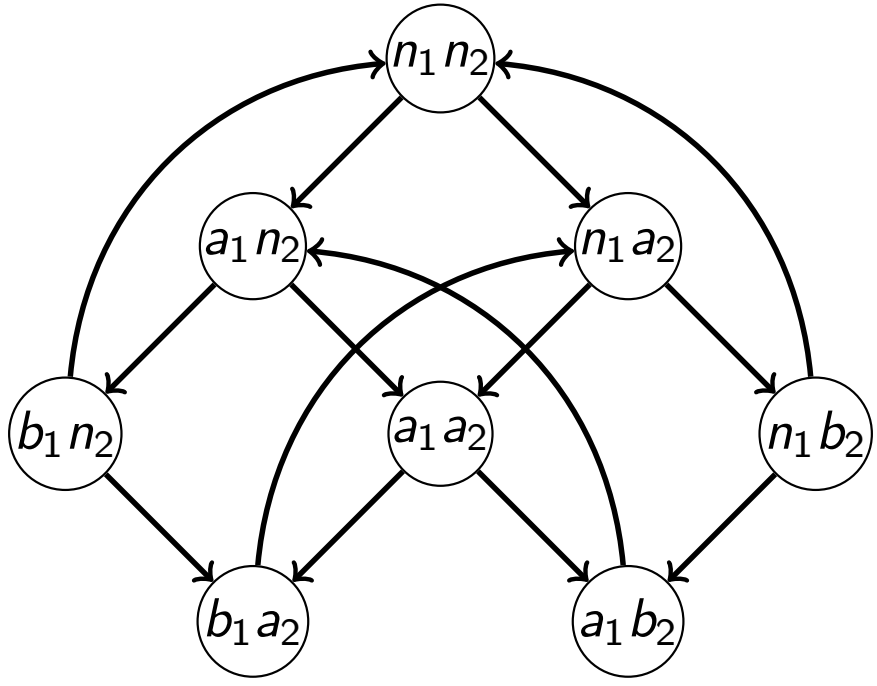
G . . . globally, F . . . future, X . . . next , U . . . until

Pfad-Quantoren



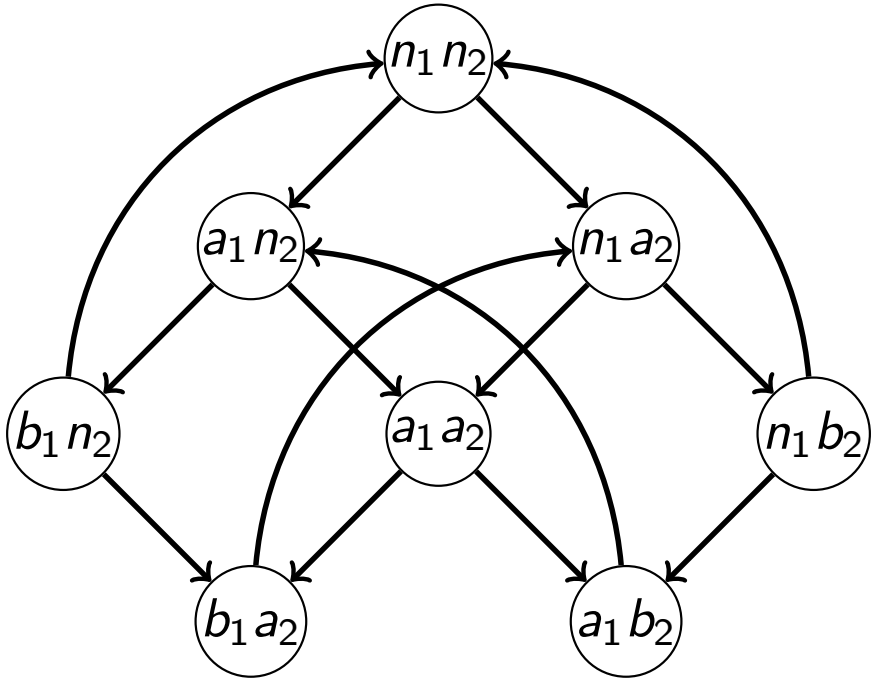
- Sicherheit: $\mathsf{AG}\,\neg(b_1 \wedge b_2)$
- Lebendigkeit: $\mathsf{AG}(a_1 \rightarrow \mathsf{AF}\,b_1)$
- kein Blockieren: $\mathsf{AG}(n_1 \rightarrow \mathsf{EX}\,a_1)$
- Flexibilität: $\mathsf{EF}(b_1 \wedge \mathsf{E}[b_1\,\mathsf{U}(\neg b_1 \wedge \mathsf{E}[\neg b_2\,\mathsf{U}\,b_1])])$

Pfad-Quantoren



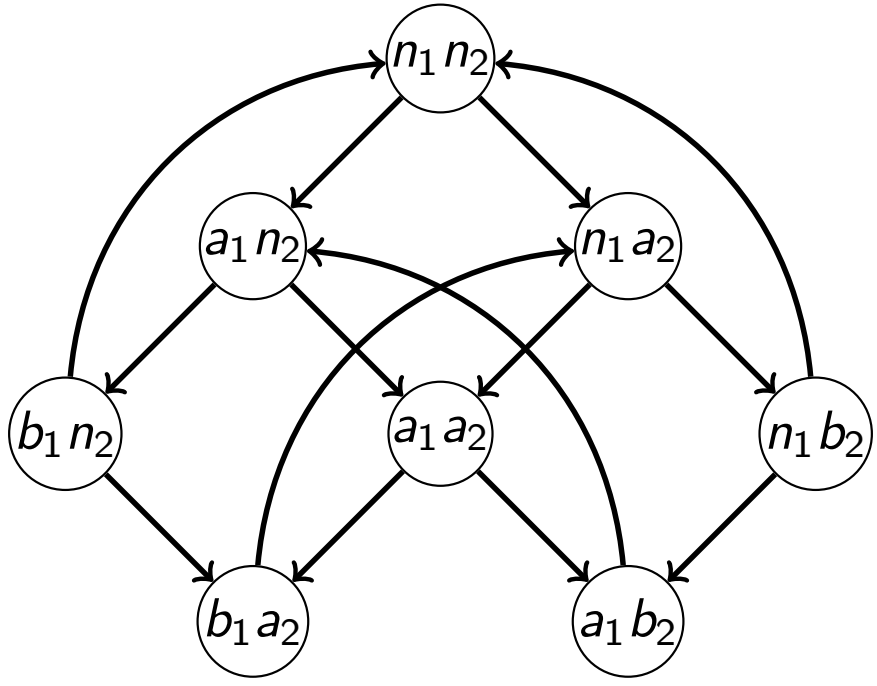
- Sicherheit: $\mathsf{AG}\,\neg(b_1 \wedge b_2)$
- Lebendigkeit: $\mathsf{AG}(a_1 \rightarrow \mathsf{AF}\,b_1)$
- kein Blockieren: $\mathsf{AG}(n_1 \rightarrow \mathsf{EX}\,a_1)$
- Flexibilität: $\mathsf{EF}(b_1 \wedge \mathsf{E}[b_1\,\mathsf{U}(\neg b_1 \wedge \mathsf{E}[\neg b_2\,\mathsf{U}\,b_1])])$

Pfad-Quantoren



- Sicherheit: $\mathsf{AG}\,\neg(b_1 \wedge b_2)$
- Lebendigkeit: $\mathsf{AG}(a_1 \rightarrow \mathsf{AF}\,b_1)$
- kein Blockieren: $\mathsf{AG}(n_1 \rightarrow \mathsf{EX}\,a_1)$
- Flexibilität: $\mathsf{EF}(b_1 \wedge \mathsf{E}[b_1\,\mathsf{U}(\neg b_1 \wedge \mathsf{E}[\neg b_2\,\mathsf{U}\,b_1])])$

Pfad-Quantoren



- Sicherheit: $\mathsf{AG}\,\neg(b_1 \wedge b_2)$
- Lebendigkeit: $\mathsf{AG}(a_1 \rightarrow \mathsf{AF}\,b_1)$
- kein Blockieren: $\mathsf{AG}(n_1 \rightarrow \mathsf{EX}\,a_1)$
- Flexibilität: $\mathsf{EF}(b_1 \wedge \mathsf{E}[b_1\,\mathsf{U}(\neg b_1 \wedge \mathsf{E}[\neg b_2\,\mathsf{U}\,b_1])])$

. . . mehr über temporale Operatoren

- $\neg \mathsf{G}\alpha \equiv \mathsf{F}\neg\alpha$
- $\mathsf{F}\alpha \equiv \top \mathsf{U}\alpha$
- $\neg \mathsf{X}\alpha \equiv \mathsf{X}\neg\alpha$
- $\neg \mathsf{A}\alpha \equiv \mathsf{E}\neg\alpha$

- mit $\mathsf{X}(\alpha \mathsf{U} \beta)$ lässt sich alles ausdrücken
- für $\mathsf{GF}\alpha$ benutzt man auch $\overset{\infty}{\mathsf{F}}\alpha$

Temporale Logiken

LTL: linear-time temporal logic

- Formeln mit temporalen Operatoren, ohne Pfad-Quantoren

$$\mathsf{F}(\mathsf{G}x \rightarrow y\mathsf{U}z)$$

- gültige Formel wird in jedem Modell auf *jedem* Pfad erfüllt
- Modell erfüllt Formel, wenn sie auf *einem* Pfad erfüllt wird

CTL: computation tree logic, branching-time temporal logic

- Formeln mit kombinierten Pfad-Quantoren und temporalen Operatoren

$$\mathsf{AF}(\mathsf{EG}x \rightarrow \mathsf{A}(y\mathsf{U}z))$$

CTL$^+$:

- Formeln mit alternierenden Pfad-Quantoren und temporalen Operatoren

$$\mathsf{E}(\mathsf{G}x \rightarrow (y\mathsf{U}(\mathsf{A}(x \wedge \mathsf{F}z))))$$

CTL*:

- Formeln mit Pfad-Quantoren und temporalen Operatoren

$$\mathsf{A}(\mathsf{F}x \wedge \mathsf{E}(\mathsf{GF}x \rightarrow (y\mathsf{U}z)))$$

Allgemeine Komplexitätsresultate

	LTL	CTL	CTL$^+$	CTL*
Erfüllbarkeit	PSPACE [SC85]	EXPTIME [FL79] [Pr80]	EEXPTIME [JL03]	EEXPTIME [VS85]
Model Checking	PSPACE [SC85]	P [CES86] [Sc02]	Δ_2^p [LMS01]	PSPACE [CES86]

Literatur:

[FL79] Fischer, Ladner 1979
[Pr80] Pratt 1980
[VS85] Vardi, Stockmeyer 1985
[SC85] Sistla, Clarke 1985
[CES86] Clarke, Emerson, Sistla 1986
[Sc02] Schnoebelen 2002
[LMS01] Laroussinie, Markey, Schnoebelen 2001
[JL03] Johannsen, Lange 2003

Komplexität des Erfüllbarkeitsproblems

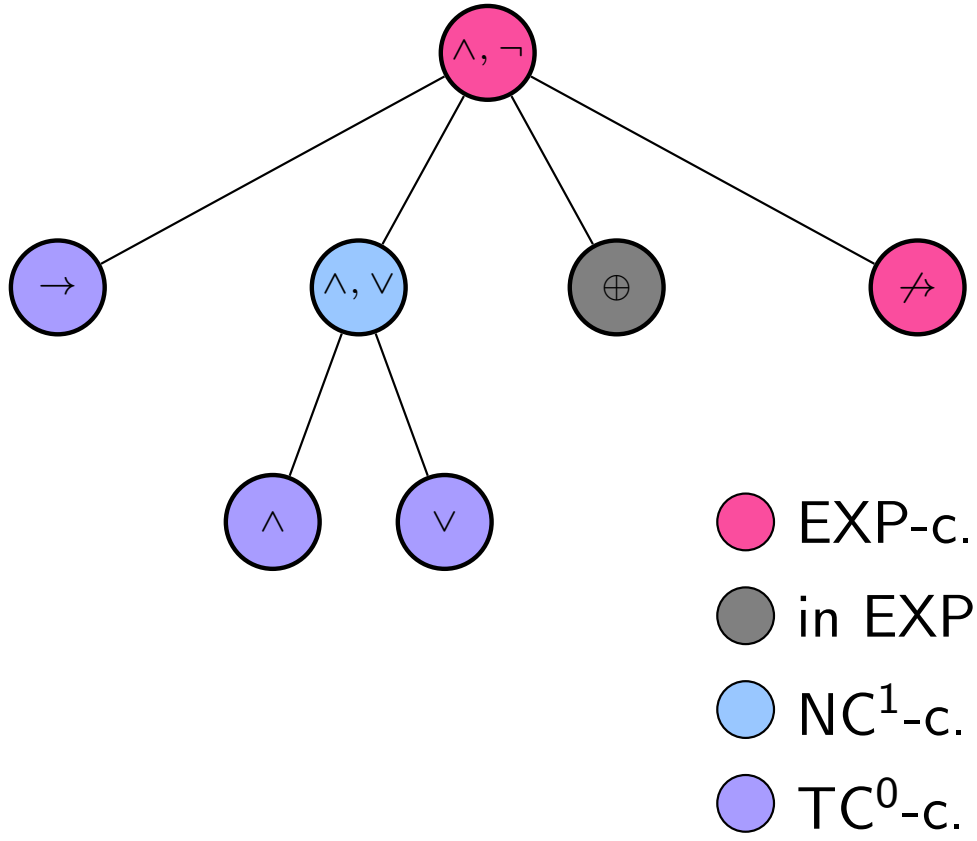
gegeben: Formel φ
gefragt: gibt es eine Modell M mit Zustand s, so dass $M, s \models \varphi$?

1. Fragmente bezgl. der Booleschen Operatoren

2. Fragmente bezgl. der temporalen Operatoren

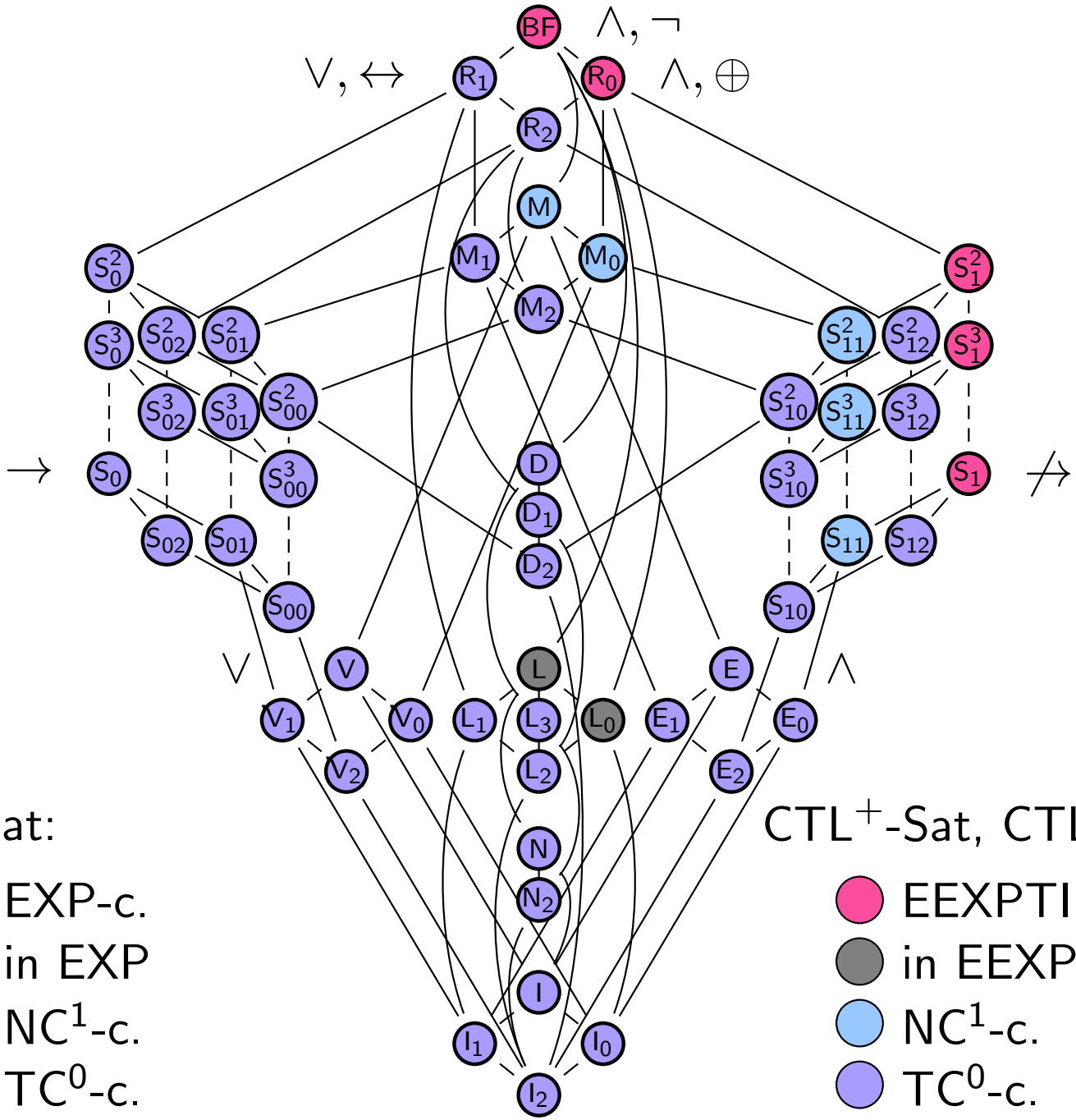
Fragmente bezgl. der Booleschen Operatoren

Komplexität von CTL-Sat:



Fischer, Ladner 1979
Pratt 1980
Meier, M., Thomas, Vollmer 2009

Fragmente à la Post's Verband



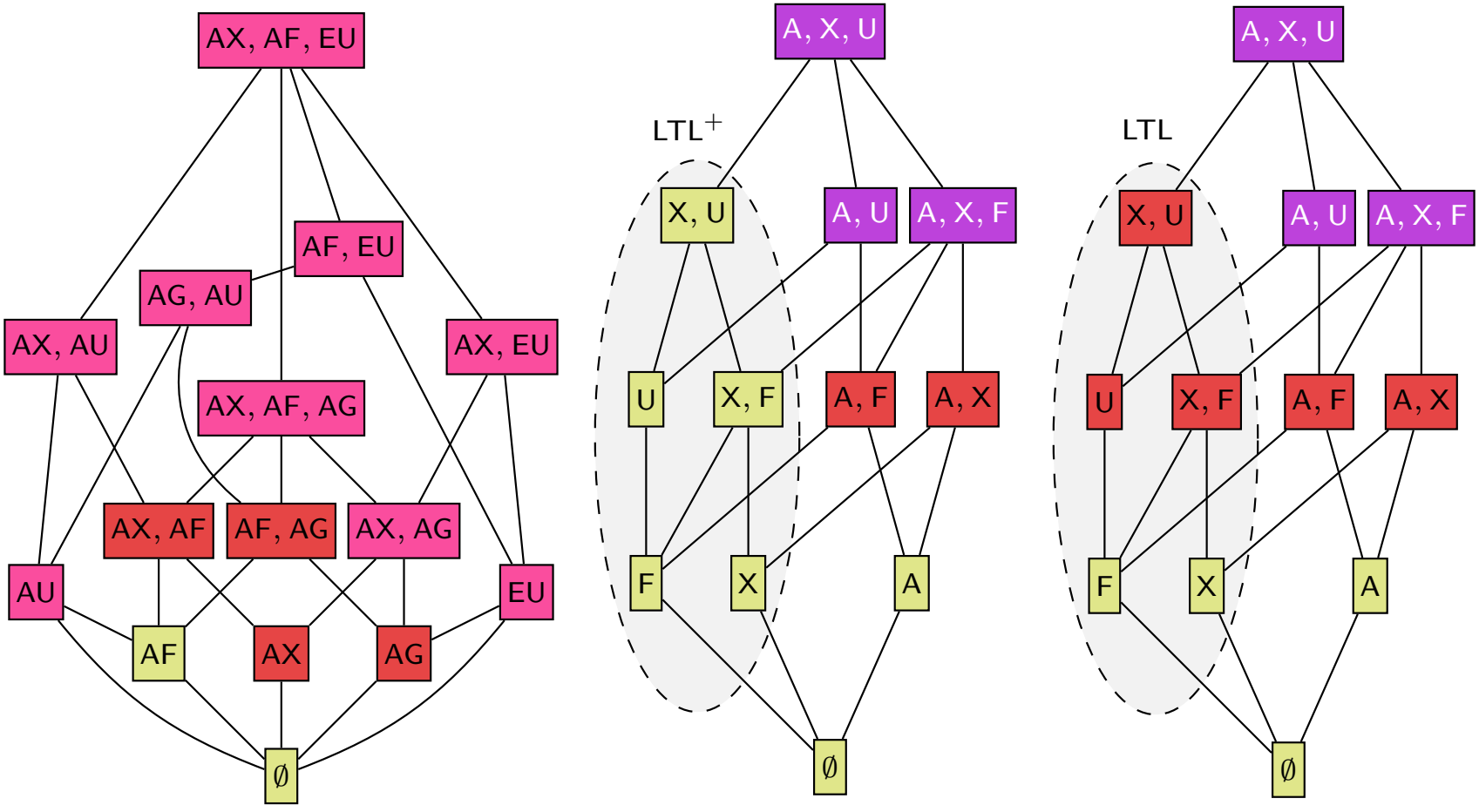
CTL-Sat:

- EXP-c.
- in EXP
- NC^1-c.
- TC^0-c.

CTL^+-Sat, $CTL^\star$-Sat:

- EEXPTIME-c.
- in EEXPTIME
- NC^1-c.
- TC^0-c.

Fragmente bezgl. der temporalen Operatoren



CTL-Sat CTL$^+$-Sat CTL*-Sat

NP-c. PSPACE-c. EXP-c. EEXPTIME-c.

Komplexität des Model Checking

gegeben: Formel φ, Modell M mit Zustand s
gefragt: $M, s \models \varphi$?

1. CTL Model Checking

2. CTL$^+$ Model Checking

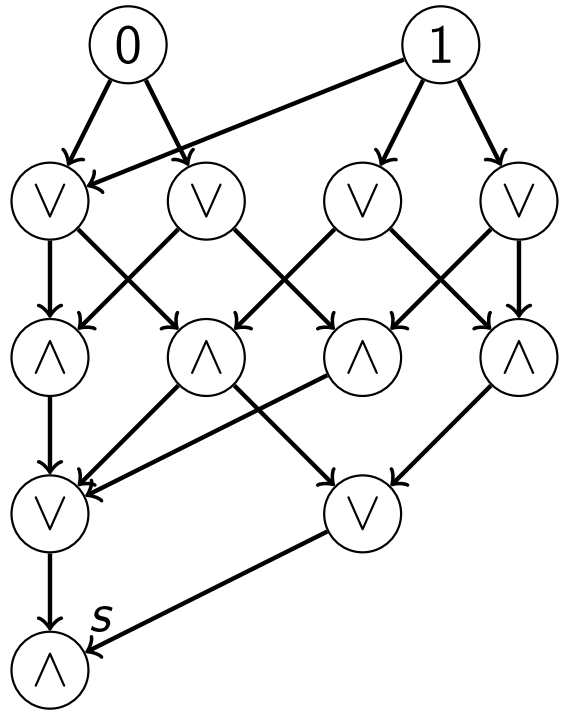
3. LTL Model Checking

4. CTL* Model Checking

P-Härte des Model Checking für CTL

Theorem (Schnoebelen 2003)

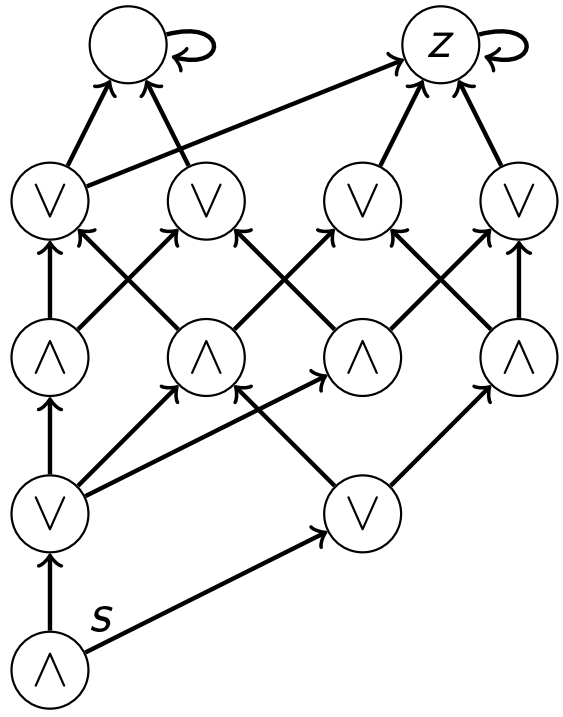
CTL *Model Checking ist* P*-vollständig.*



P-Härte des Model Checking für CTL

Theorem (Schnoebelen 2003)

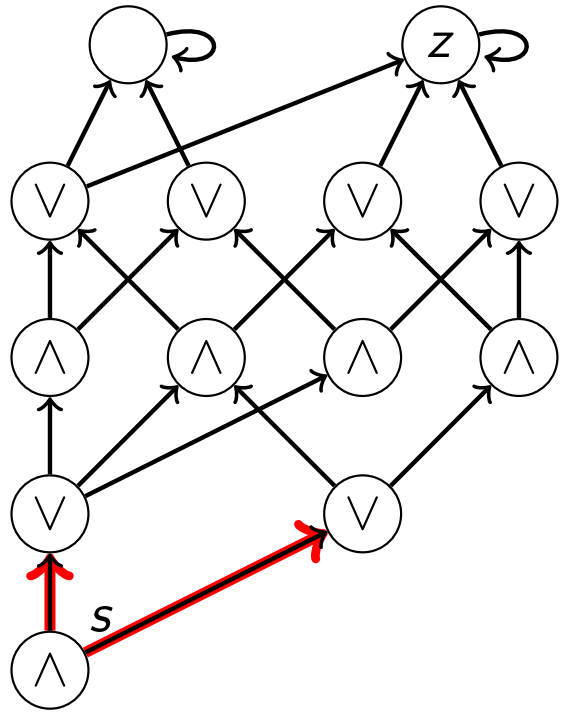
CTL *Model Checking ist* P*-vollständig.*



P-Härte des Model Checking für CTL

Theorem (Schnoebelen 2003)

CTL *Model Checking ist* P*-vollständig.*

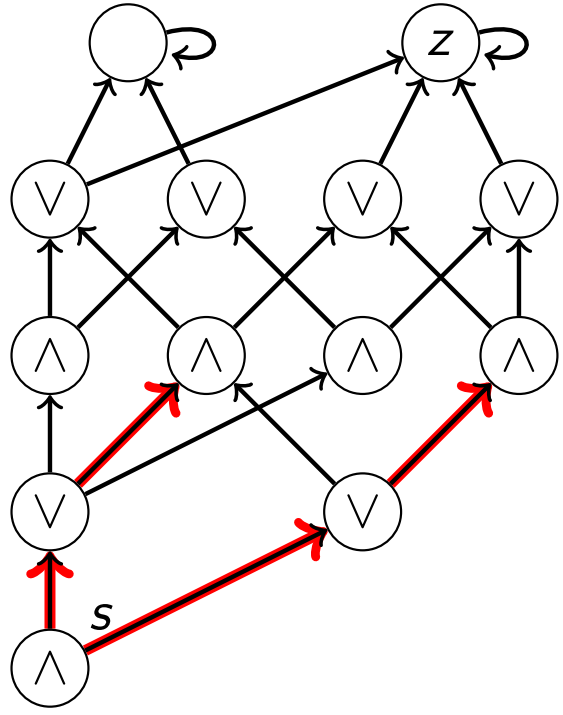


AX

P-Härte des Model Checking für CTL

Theorem (Schnoebelen 2003)

CTL *Model Checking ist* P*-vollständig.*

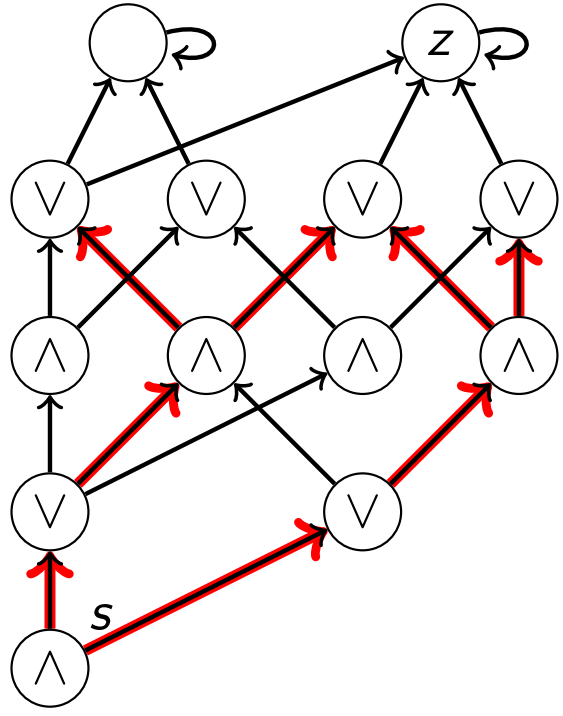


AX EX

P-Härte des Model Checking für CTL

Theorem (Schnoebelen 2003)

CTL *Model Checking ist* P*-vollständig.*

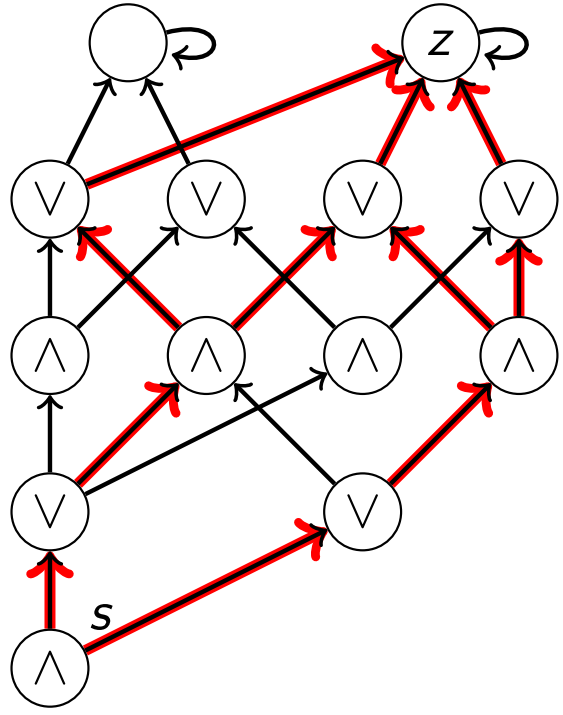


AX EX AX

P-Härte des Model Checking für CTL

Theorem (Schnoebelen 2003)

CTL *Model Checking ist* P*-vollständig.*

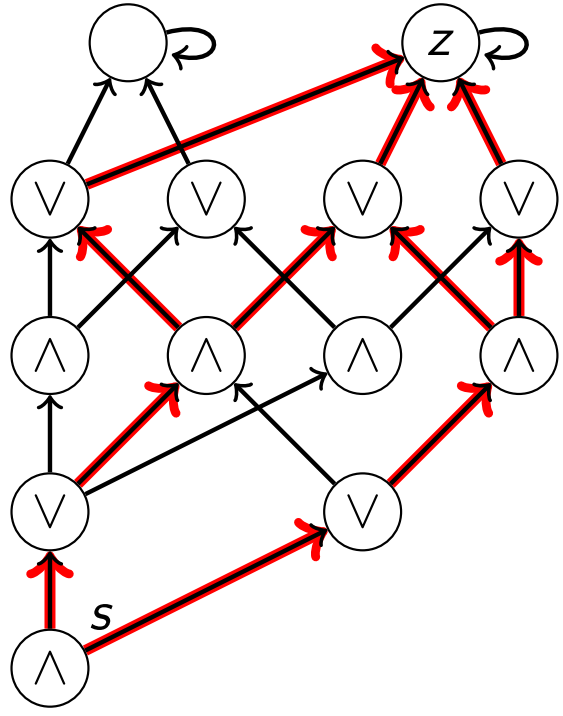


AX EX AX EX z wird in s erfüllt gdw. SK hat Ausgabe 1.

P-Härte des Model Checking für CTL

Theorem (Schnoebelen 2003)

CTL *Model Checking ist* P*-vollständig.*



AX EX AX EX z wird in s erfüllt gdw. SK hat Ausgabe 1.
AX¬AX¬AX¬AX¬z

CTL Model Checking – P-vollständig

Theorem (Beyersdorff et al, 2011)

Model Checking ist P*-vollständig für alle* CTL*-Fragmente mit*

- *nur einem* CTL*-Operator und*
- *allen Booleschen Operatoren* ($\wedge$, $\vee$, $\neg$).

Konjunktive und disjunktive Eigenschaften von Operatoren:

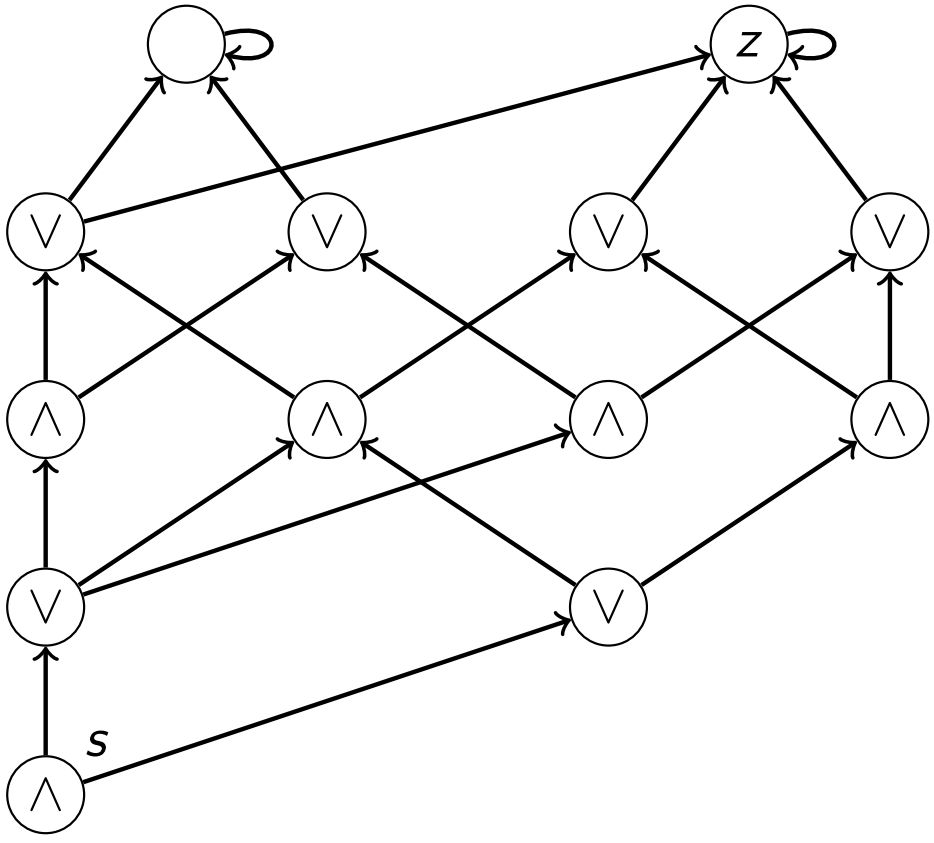
- A und G sind konjunktiv (also auch AG und AX)
- E und F sind disjunktiv (also auch EF und EX)
- U ist beides (also auch AU, EU, AF und EG)
- X ist nichts

Theorem (Beyersdorff et al, 2011)

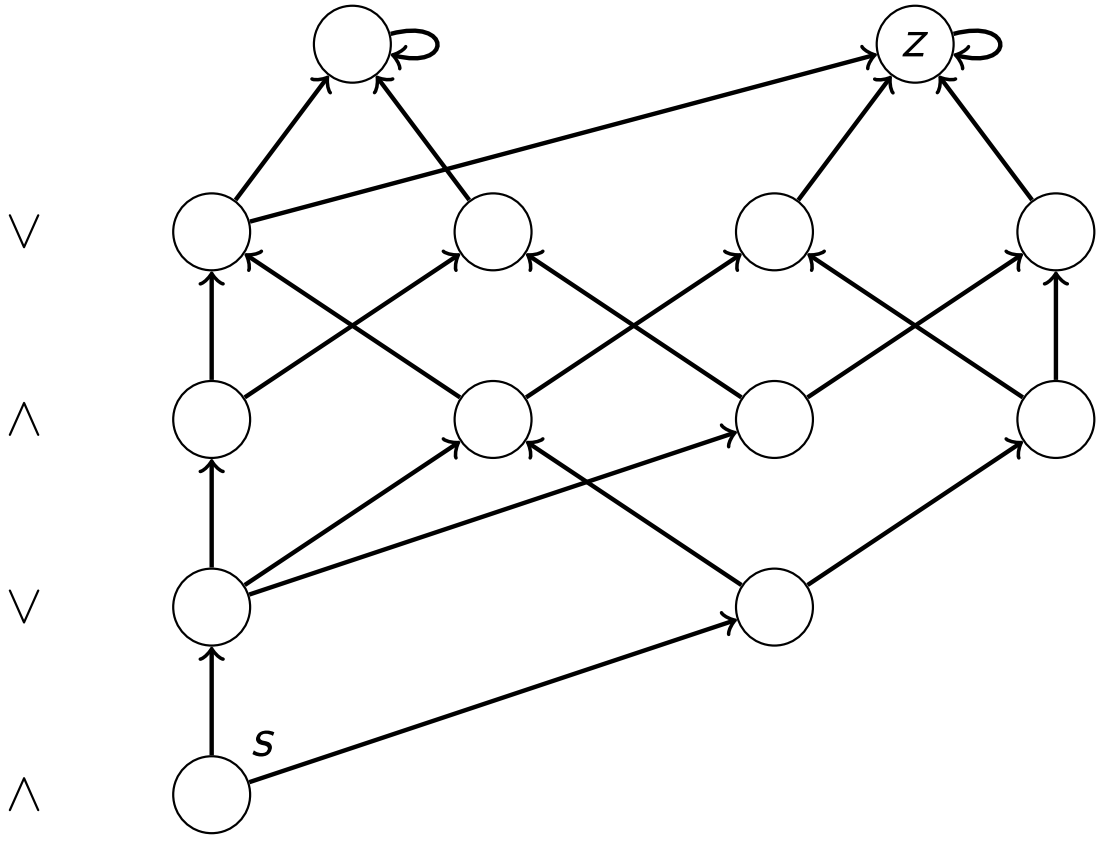
Model Checking ist P*-vollständig für alle* CTL*-Fragmente mit*

- *konjunktiven und disjunktiven* CTL*-Operator(en) und*
- *Konjunktion und Disjunktion* ($\wedge$, $\vee$).

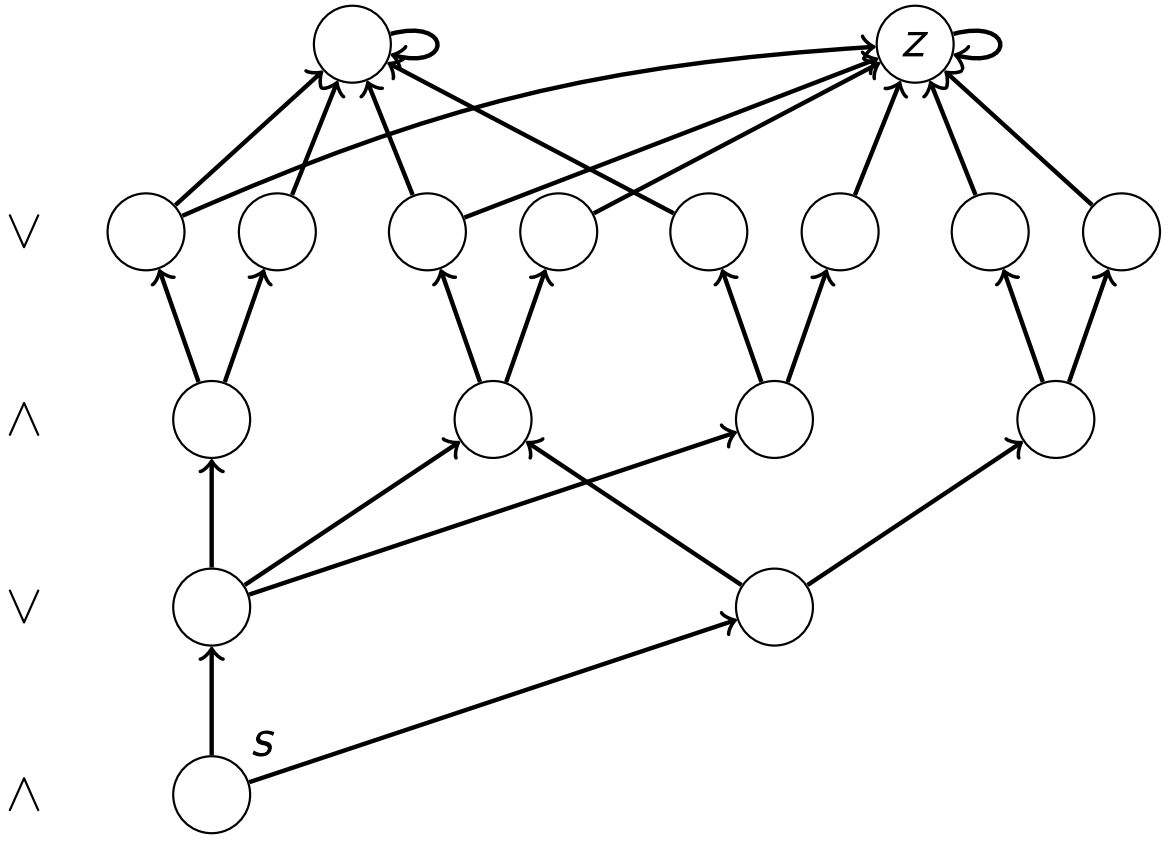
Was „einfache“ CTL-Operatoren können



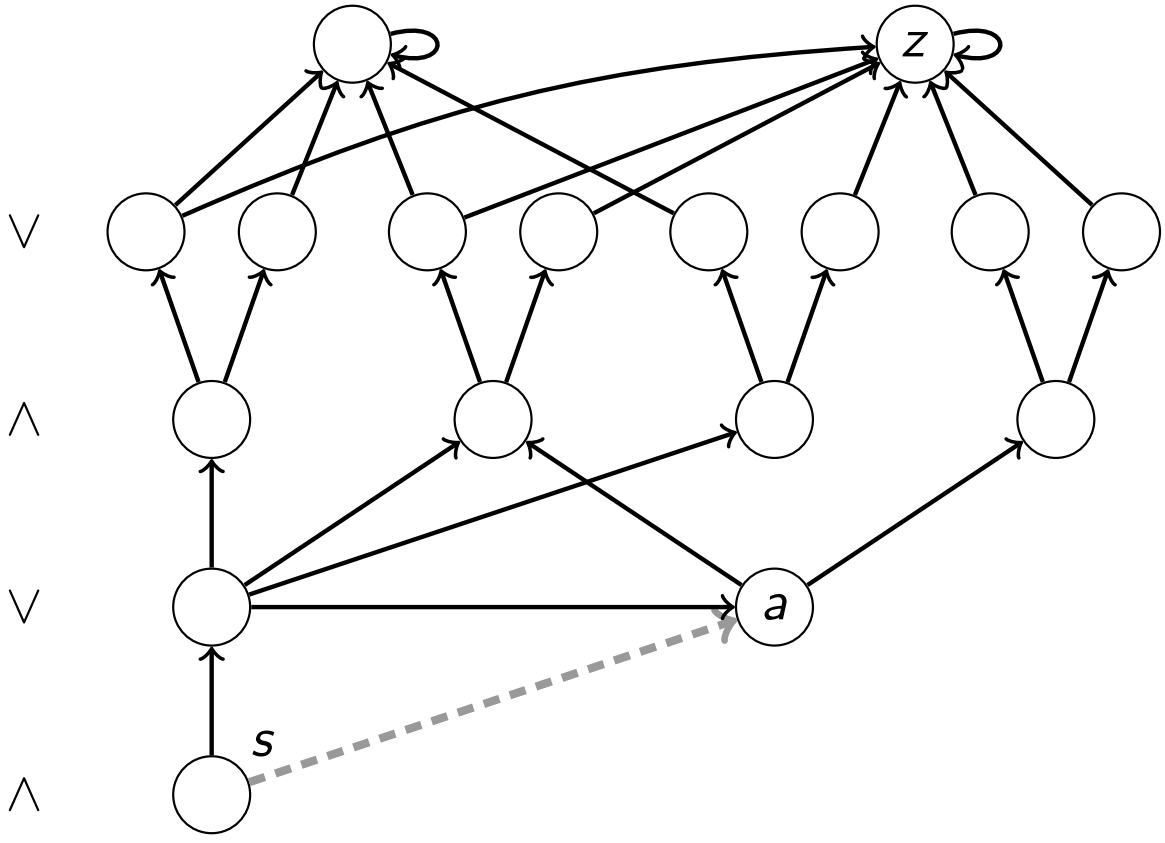
Was „einfache“ CTL-Operatoren können



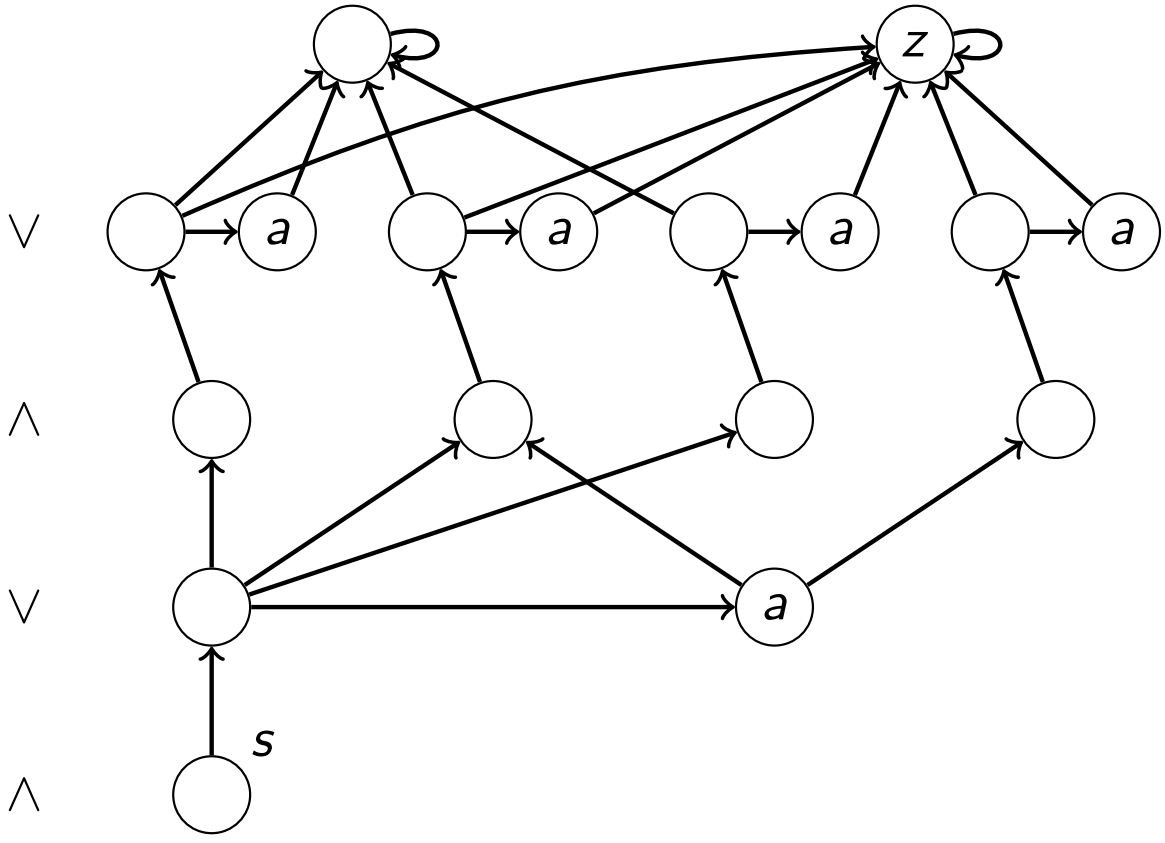
Was „einfache“ CTL-Operatoren können



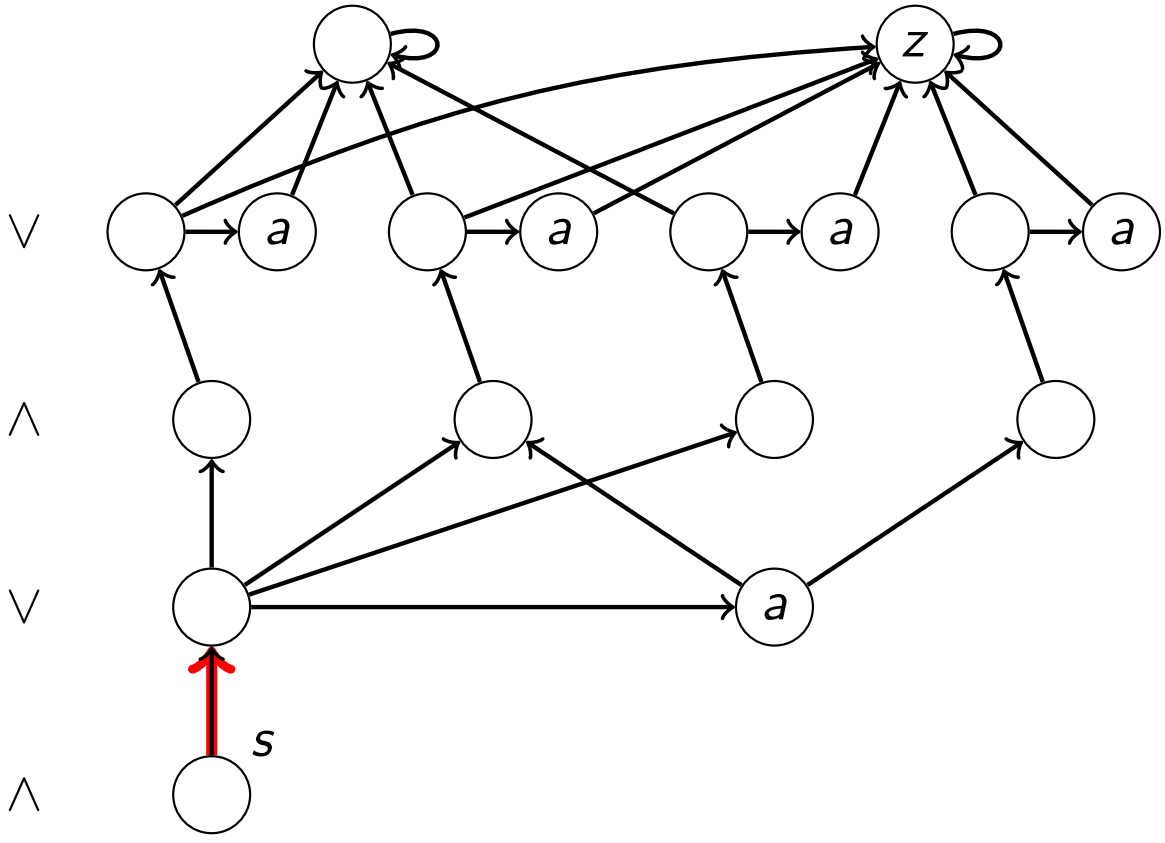
Was „einfache“ CTL-Operatoren können



Was „einfache“ CTL-Operatoren können

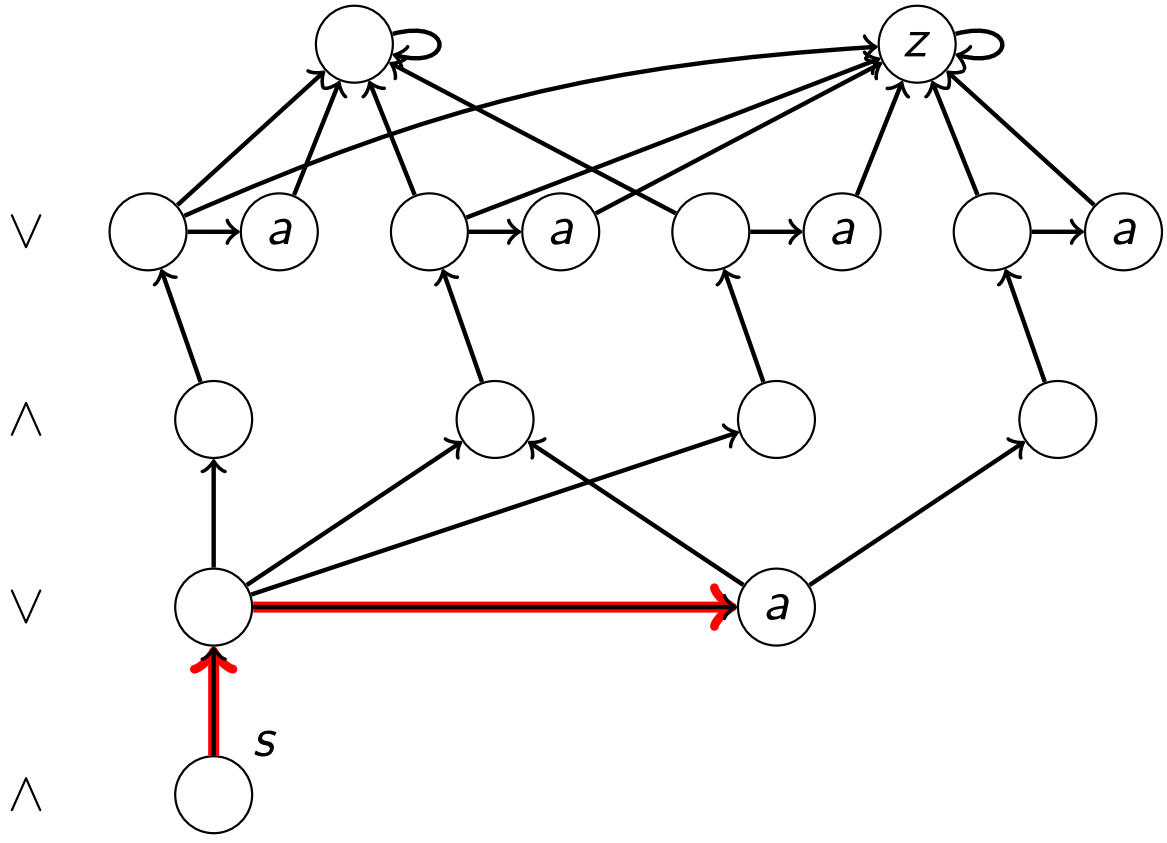


Was „einfache“ CTL-Operatoren können



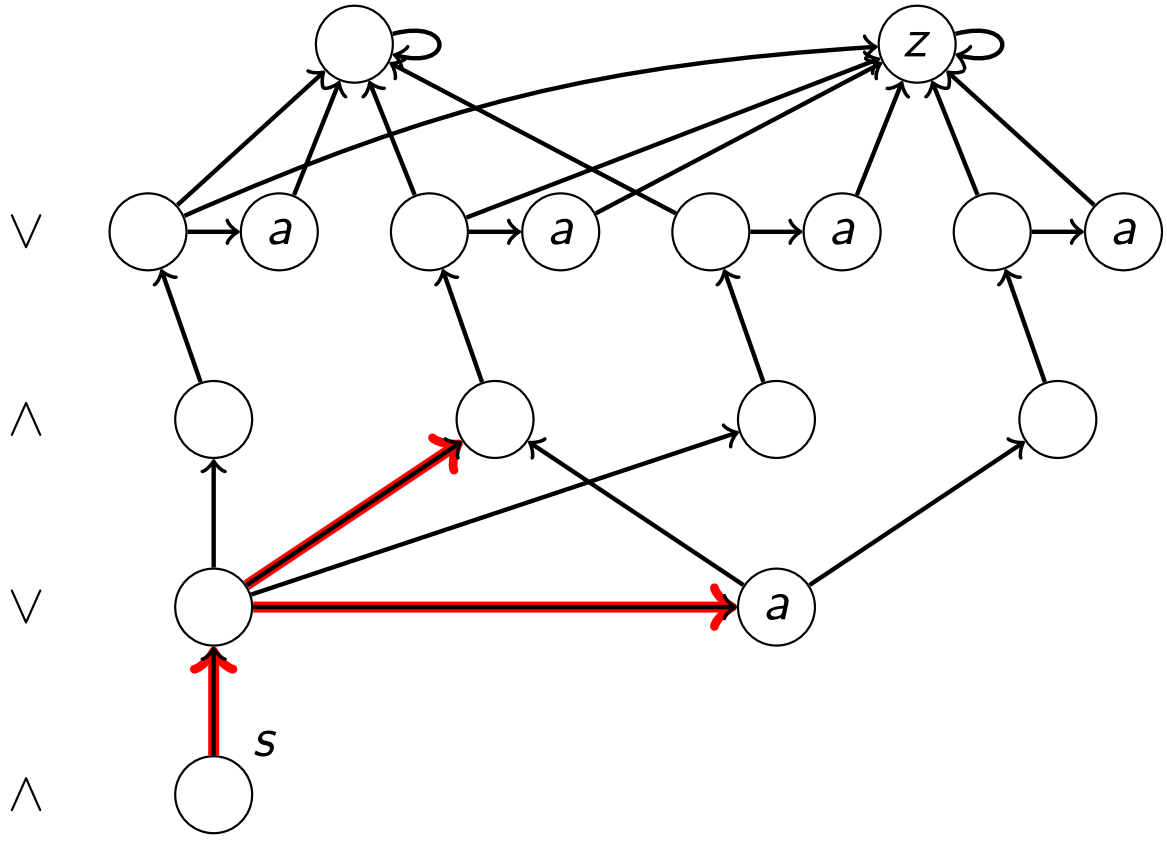
EX( )

Was „einfache“ CTL-Operatoren können



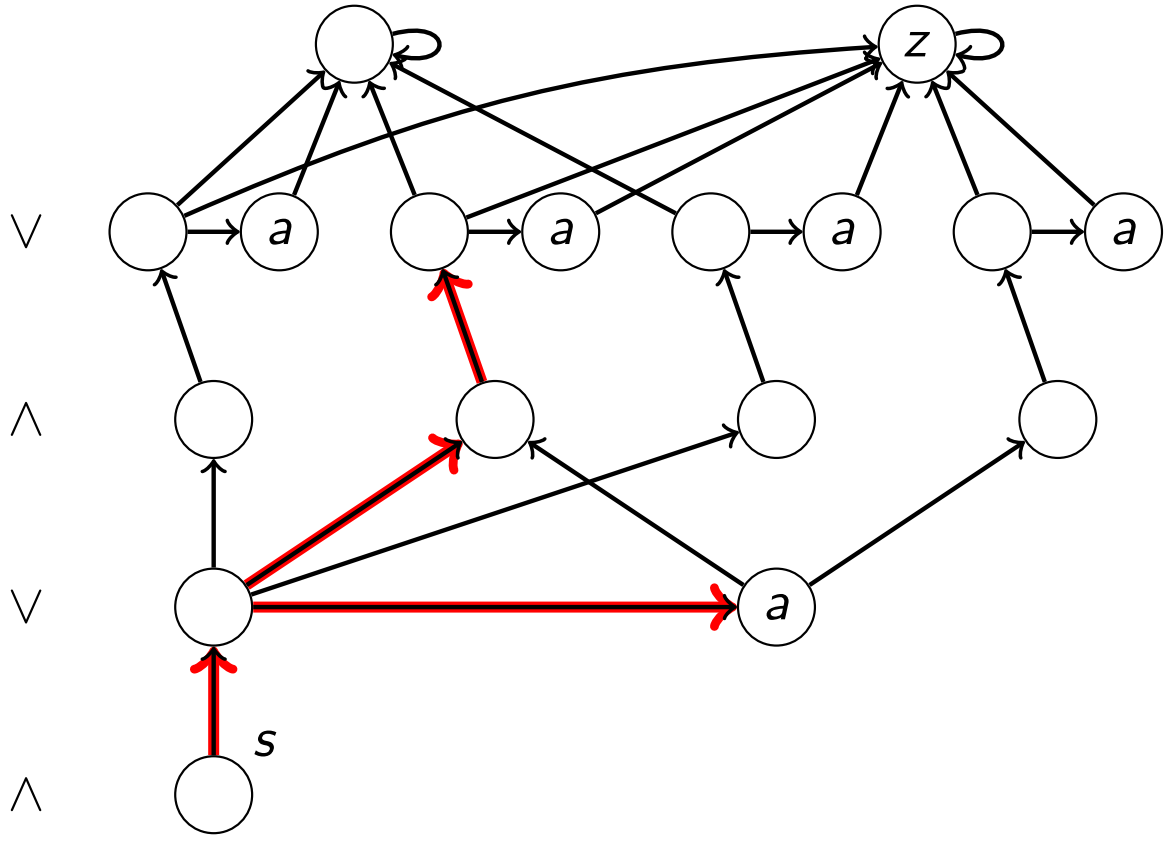
EX($\wedge$EX($a\wedge$))

Was „einfache“ CTL-Operatoren können



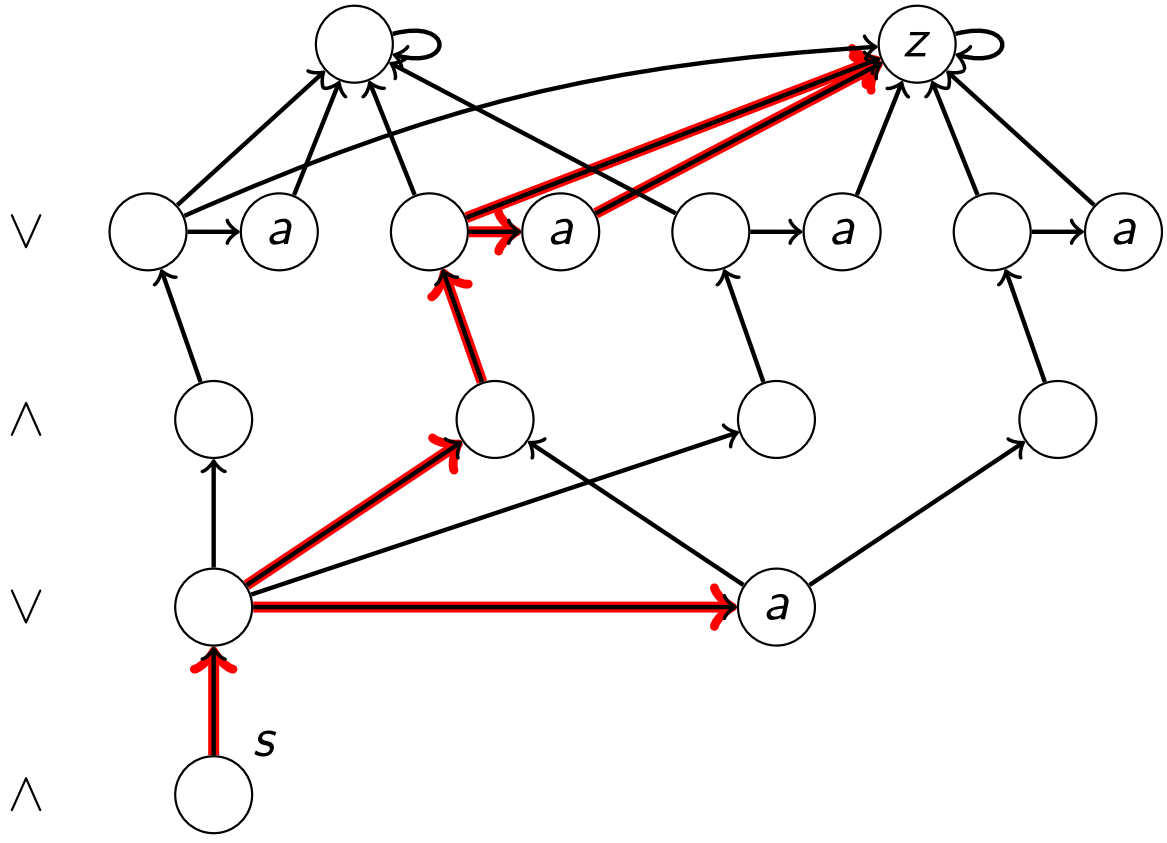
EX(EX $\wedge$EX($a\wedge$))

Was „einfache“ CTL-Operatoren können



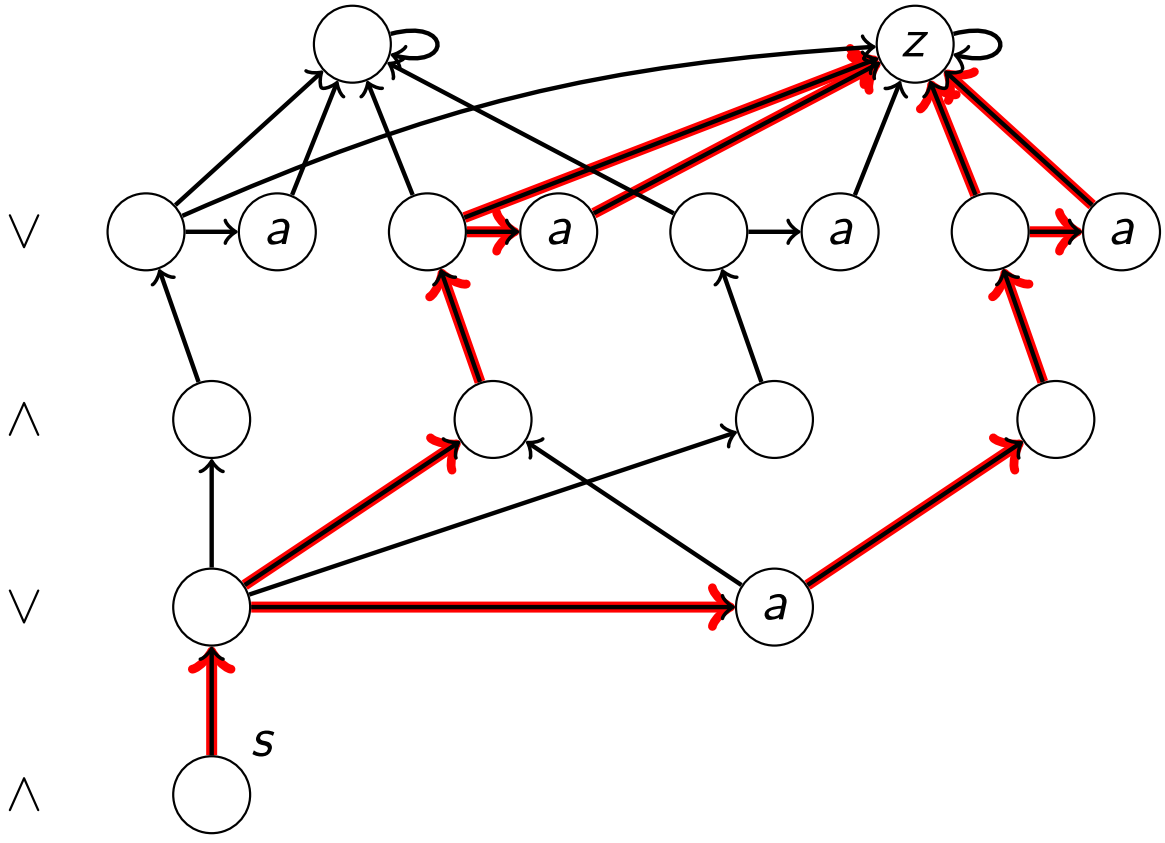
$\mathsf{EX}(\mathsf{EX}\,\mathsf{EX}(\qquad\qquad))\wedge\mathsf{EX}(a\wedge\qquad\qquad))$

Was „einfache“ CTL-Operatoren können



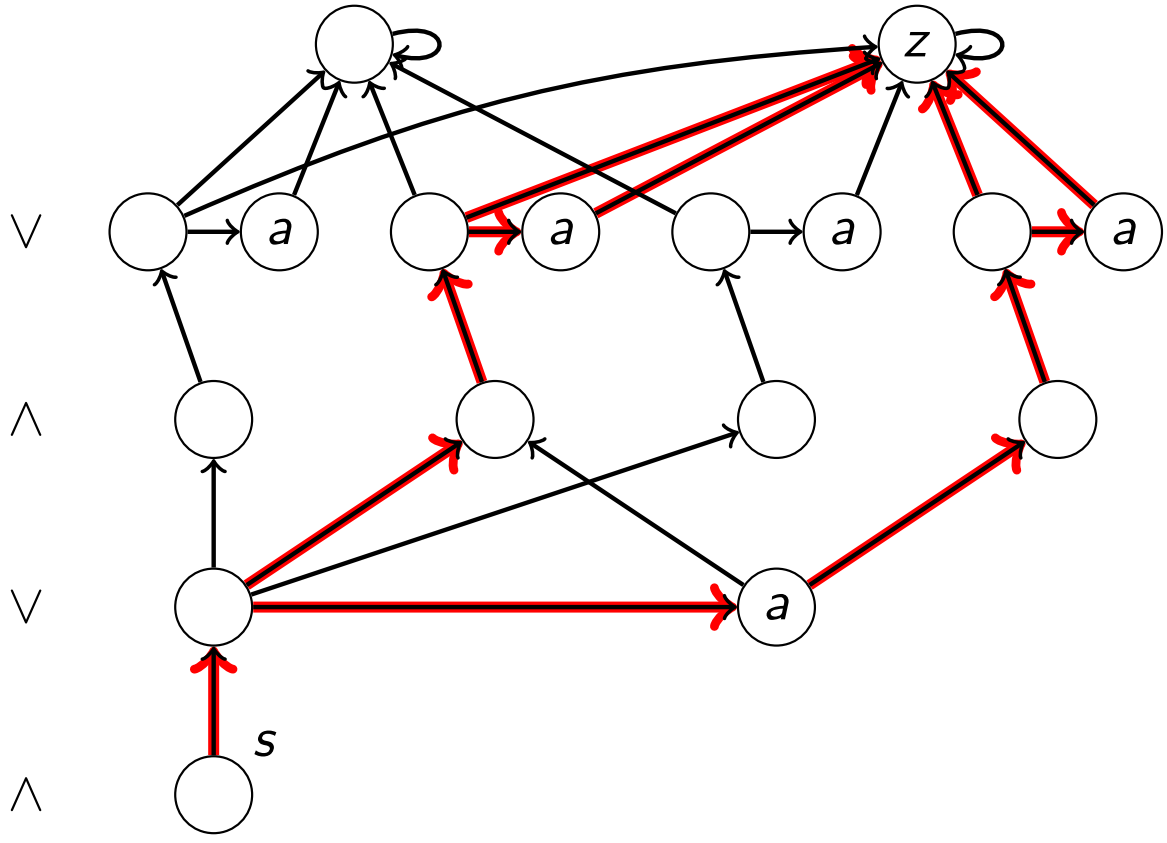
$$\mathrm{EX}(\mathrm{EX}\,\mathrm{EX}(\mathrm{EX}\,z \wedge \mathrm{EX}(a \wedge \mathrm{EX}\,z)) \wedge \mathrm{EX}(a \wedge \qquad\qquad))$$

Was „einfache“ CTL-Operatoren können



$$\mathsf{EX}(\mathsf{EX}\,\mathsf{EX}(\mathsf{EX}\,z \wedge \mathsf{EX}(a \wedge \mathsf{EX}\,z)) \wedge \mathsf{EX}(a \wedge \mathsf{EX}\,\mathsf{EX}(\mathsf{EX}\,z \wedge \mathsf{EX}(a \wedge \mathsf{EX}\,z))))$$

Was „einfache“ CTL-Operatoren können



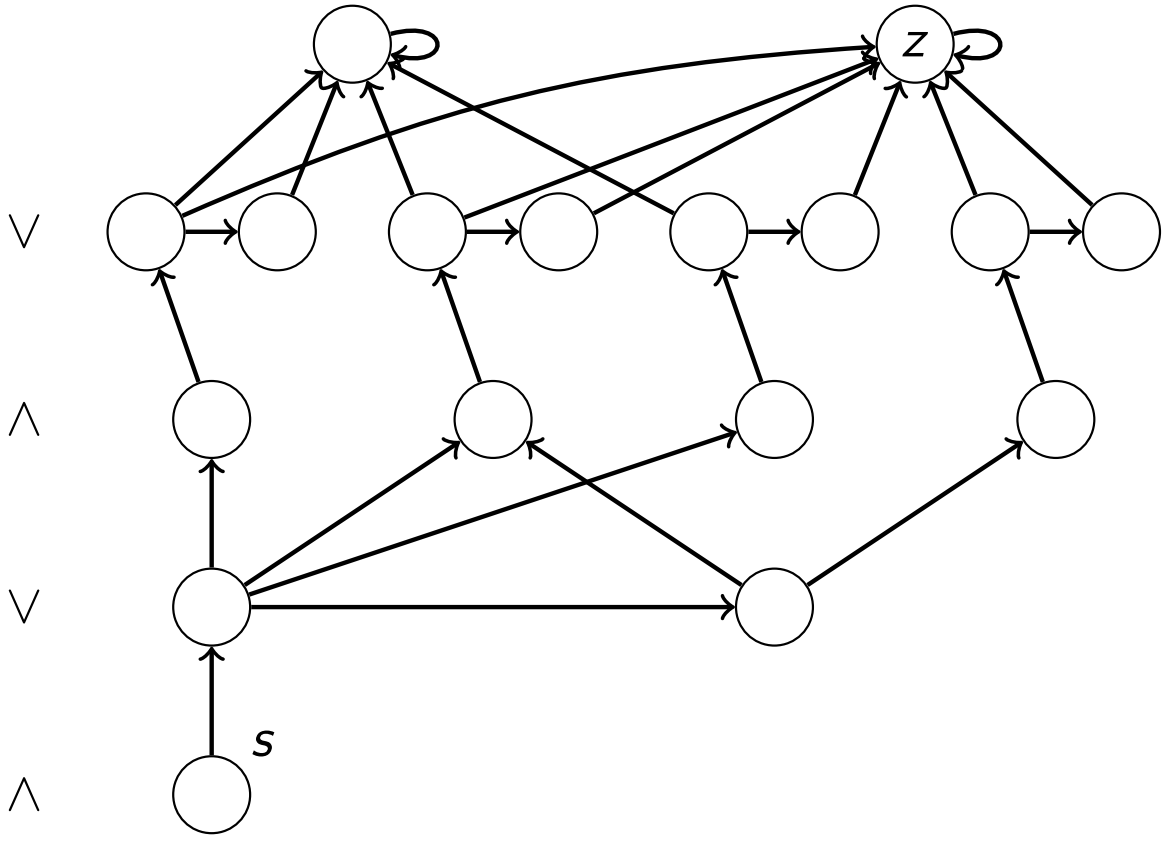
$$\mathsf{EX}(\mathsf{EX}\,\mathsf{EX}(\mathsf{EX}\,z \wedge \mathsf{EX}(a \wedge \mathsf{EX}\,z)) \wedge \mathsf{EX}(a \wedge \mathsf{EX}\,\mathsf{EX}(\mathsf{EX}\,z \wedge \mathsf{EX}(a \wedge \mathsf{EX}\,z))))$$

Theorem

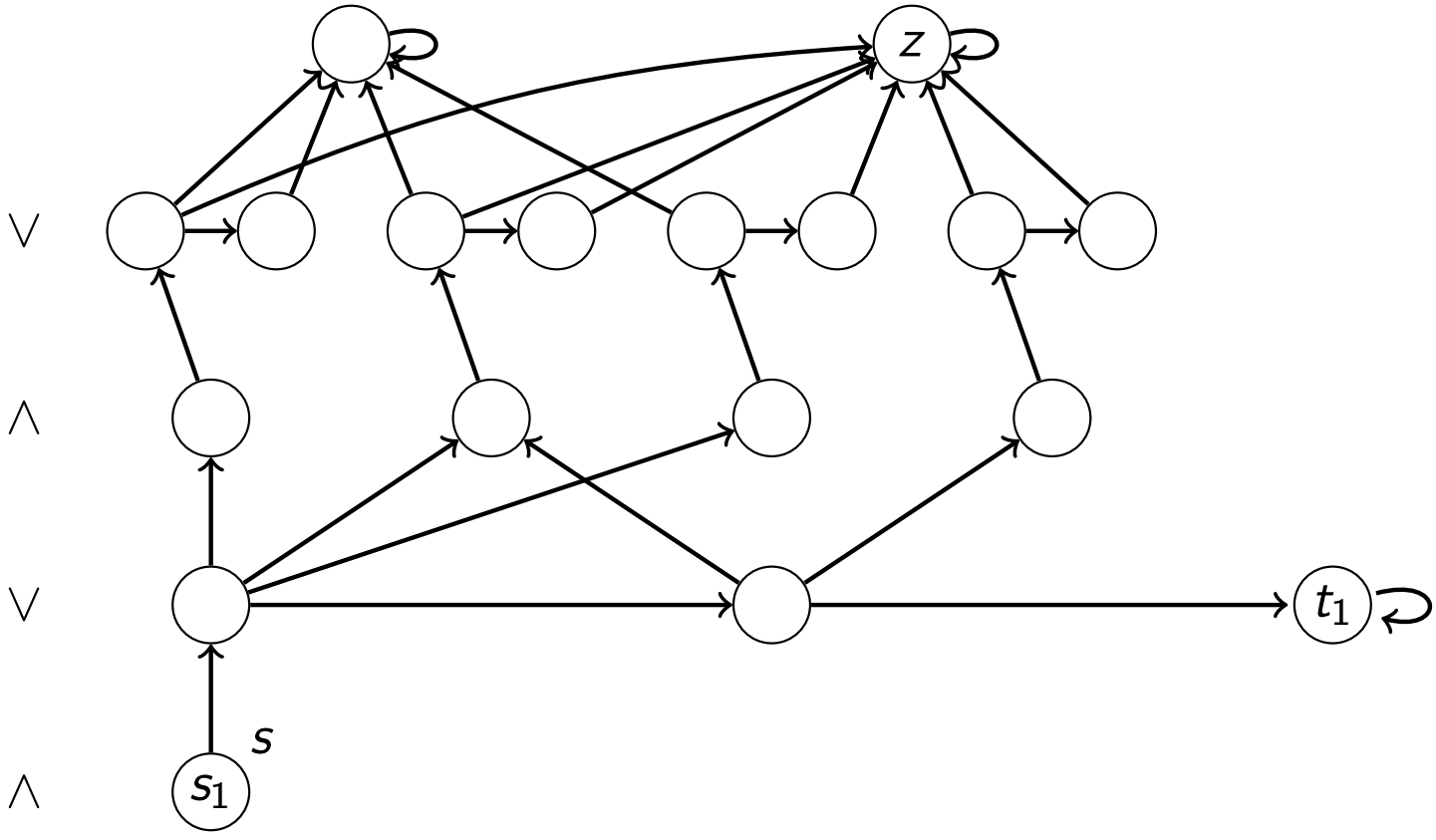
Model checking ist LOGCFL*-hart für alle* CTL*-Fragmente mit*

- *einem* CTL*-Operator und*
- *Konjunktion und Disjunktion.*

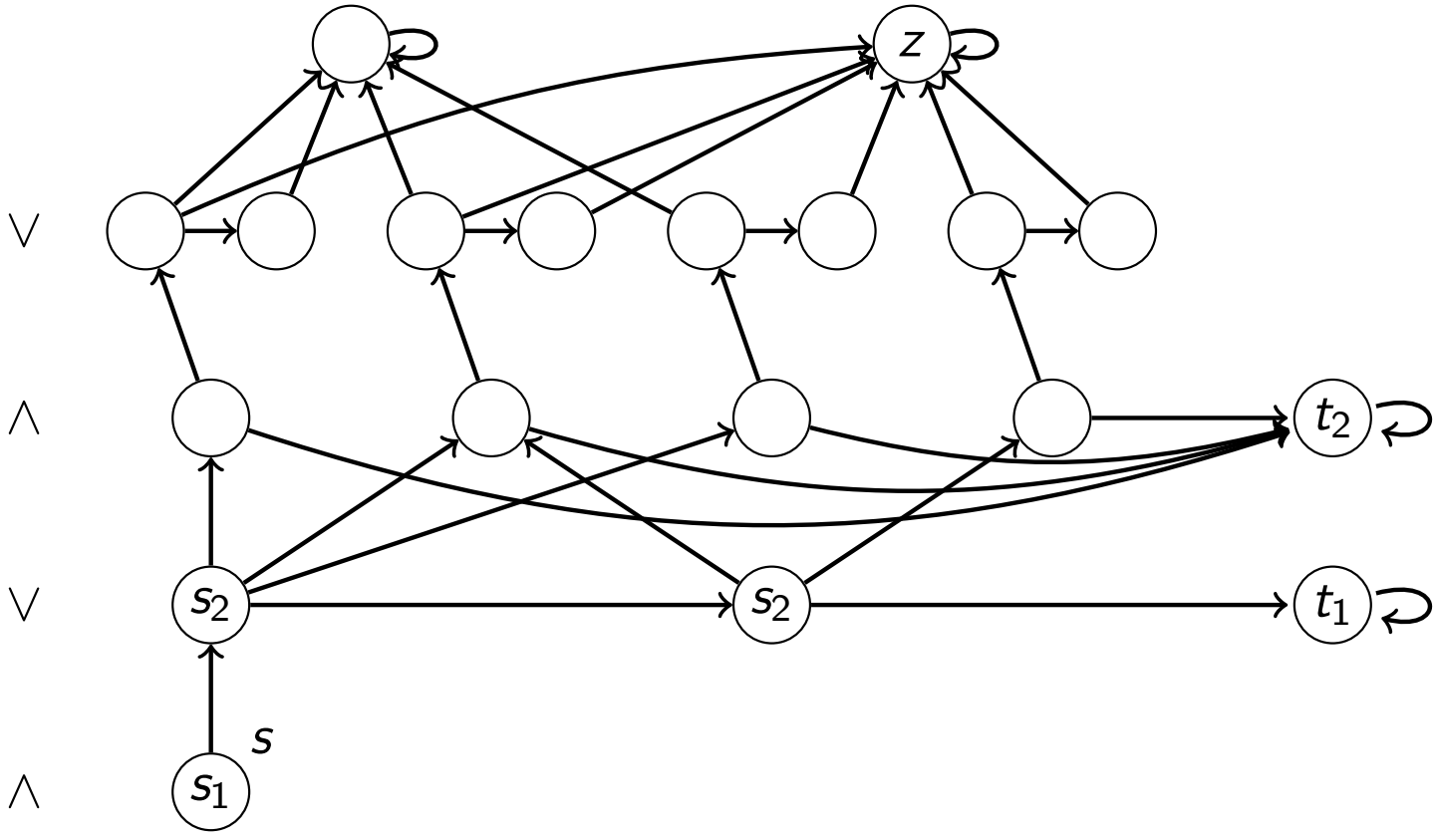
Was „doppelte“ CTL-Operatoren können



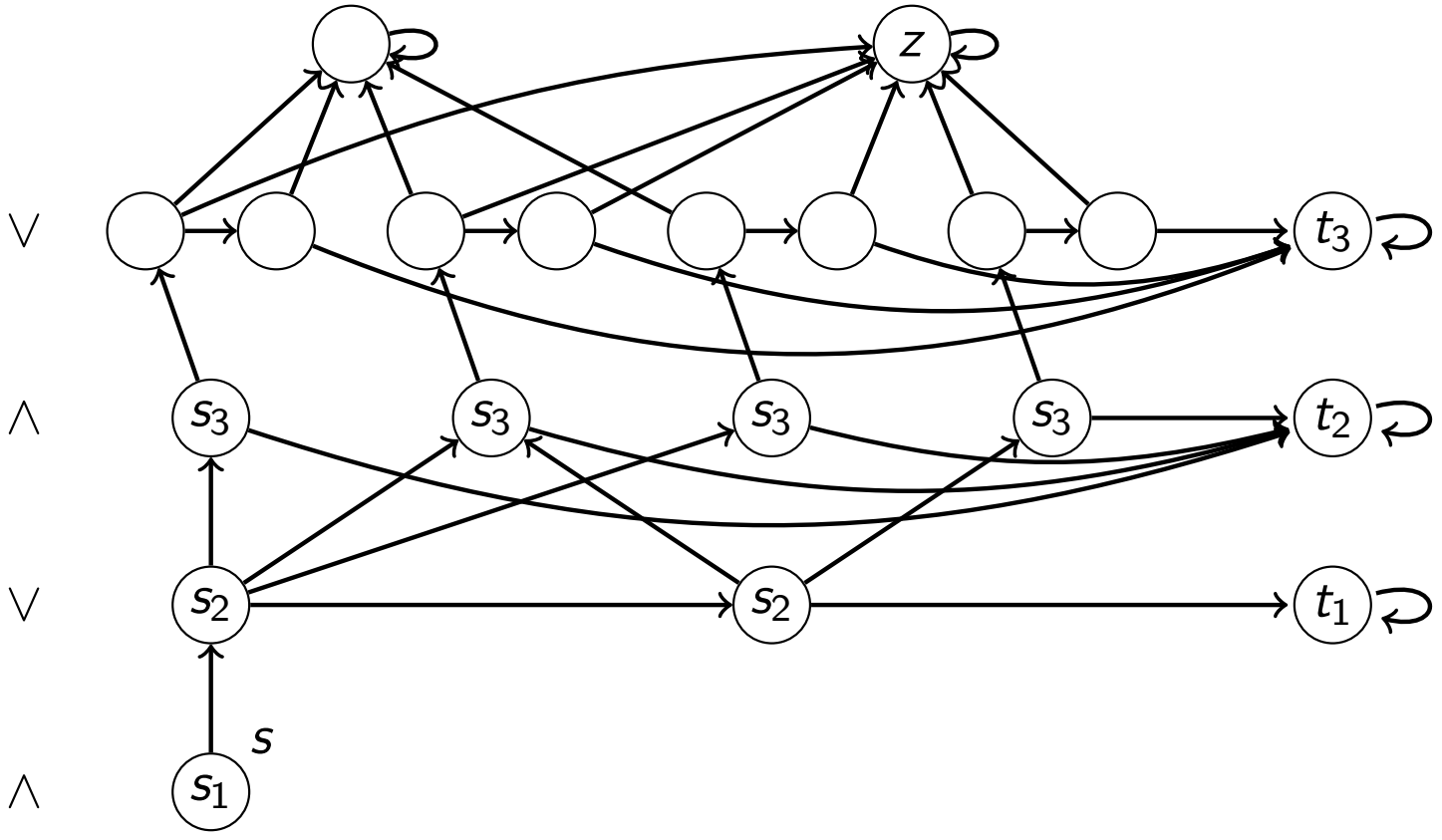
Was „doppelte“ CTL-Operatoren können



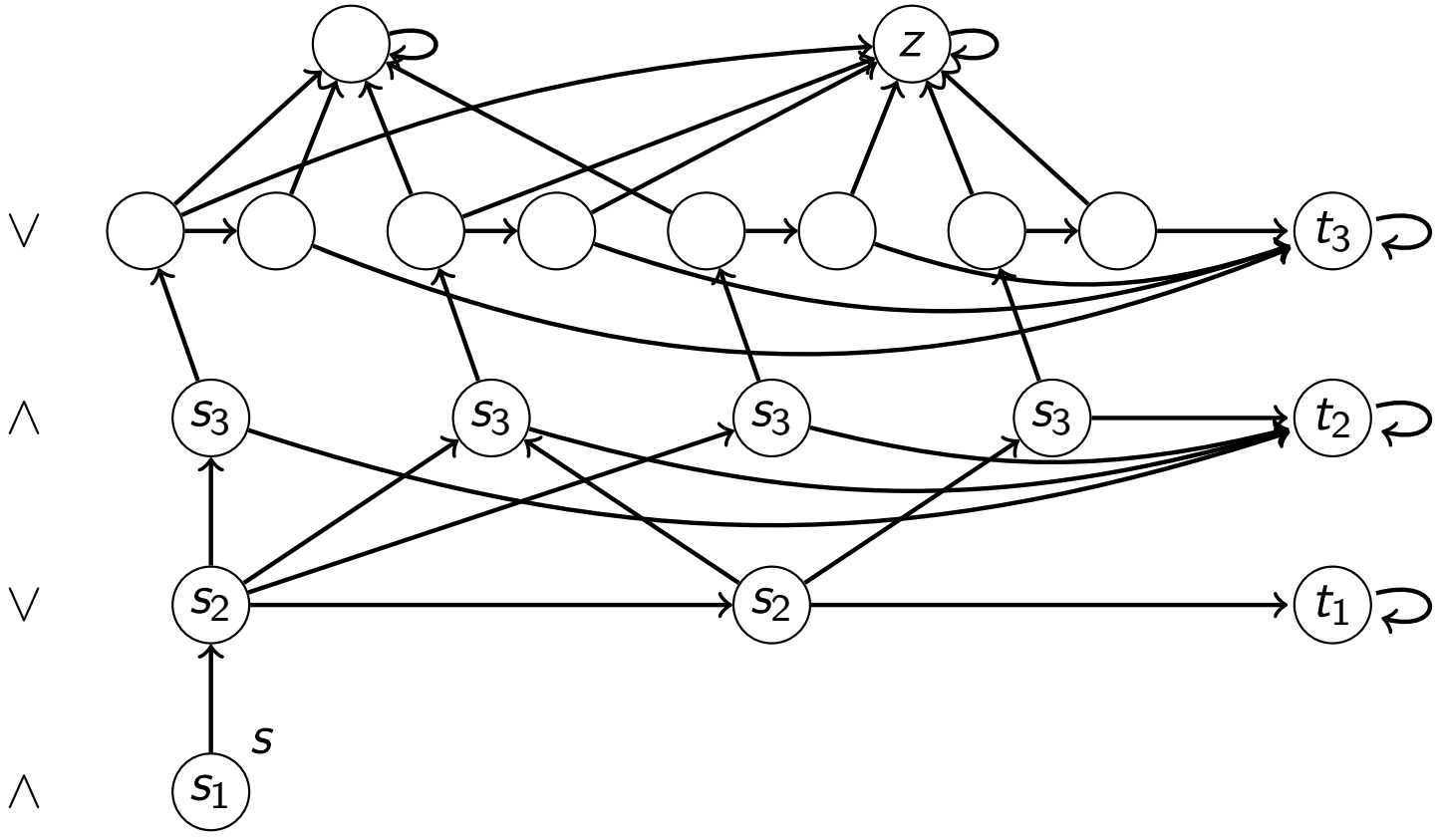
Was „doppelte“ CTL-Operatoren können



Was „doppelte“ CTL-Operatoren können



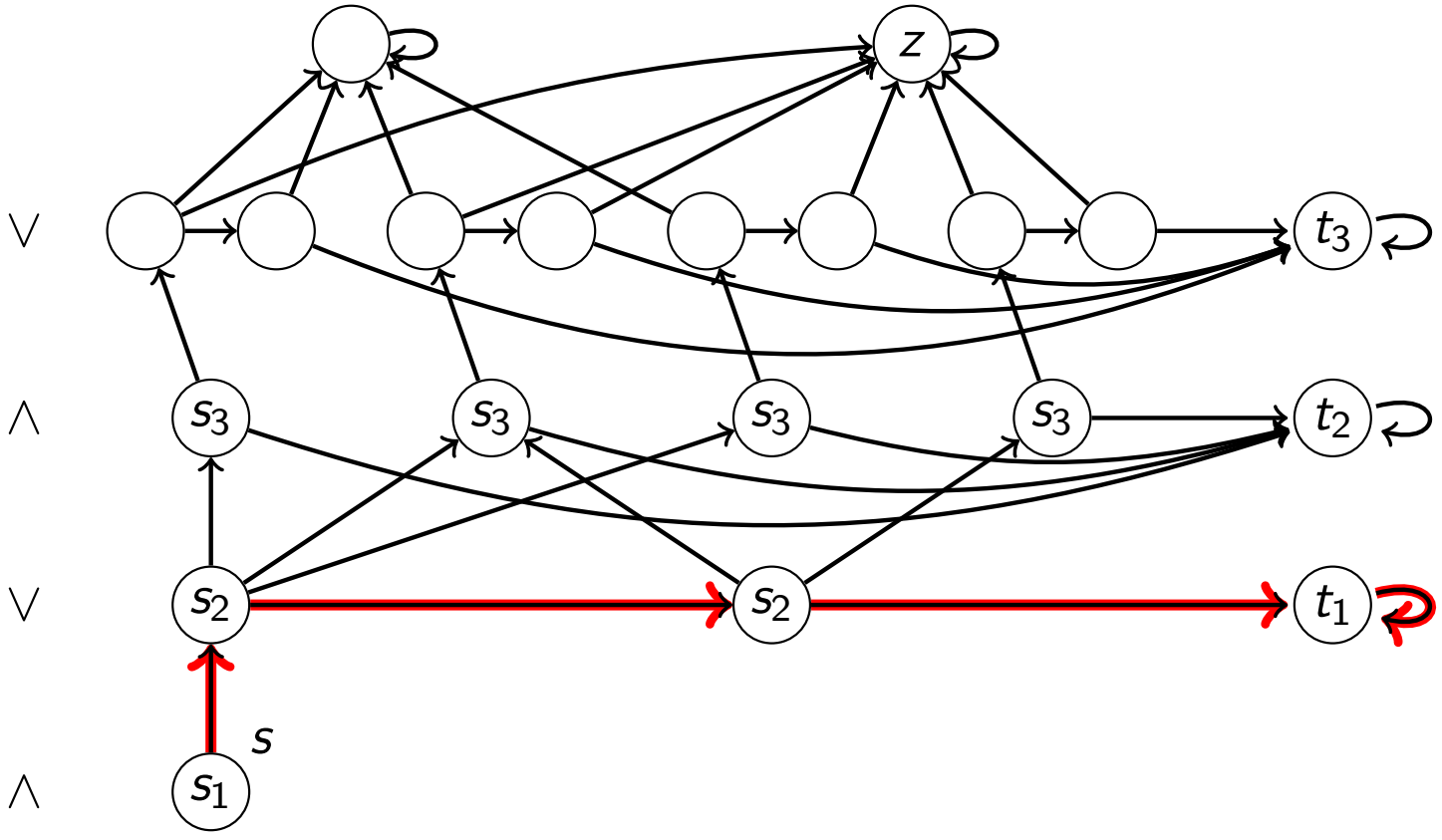
Was „doppelte“ CTL-Operatoren können



$\mathrm{EG}(s_1 \vee t_1 \vee (s_2 \wedge$

$))$

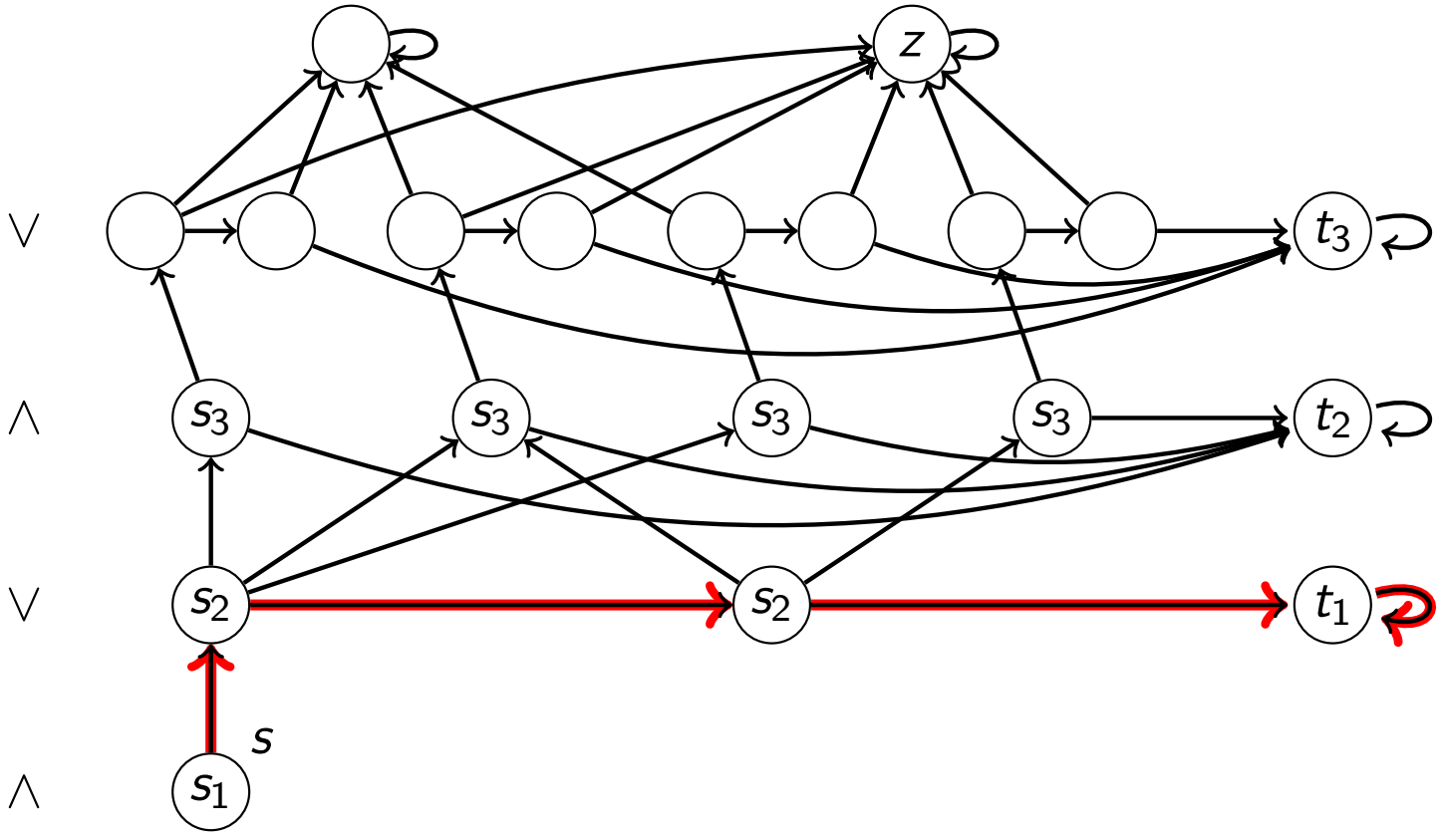
Was „doppelte“ CTL-Operatoren können



$\mathrm{EG}(s_1 \vee t_1 \vee (s_2 \wedge$

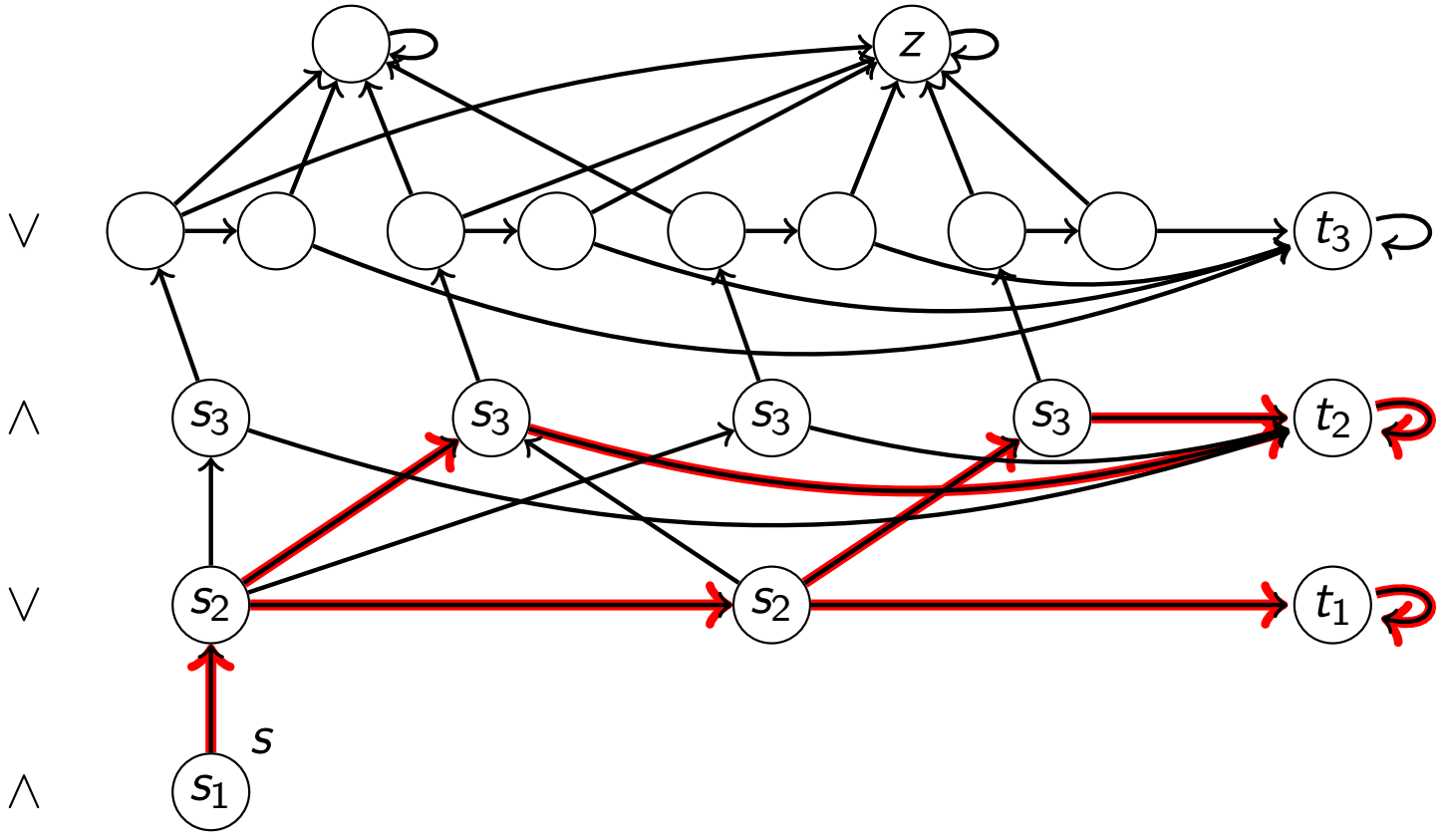
$))$

Was „doppelte“ CTL-Operatoren können



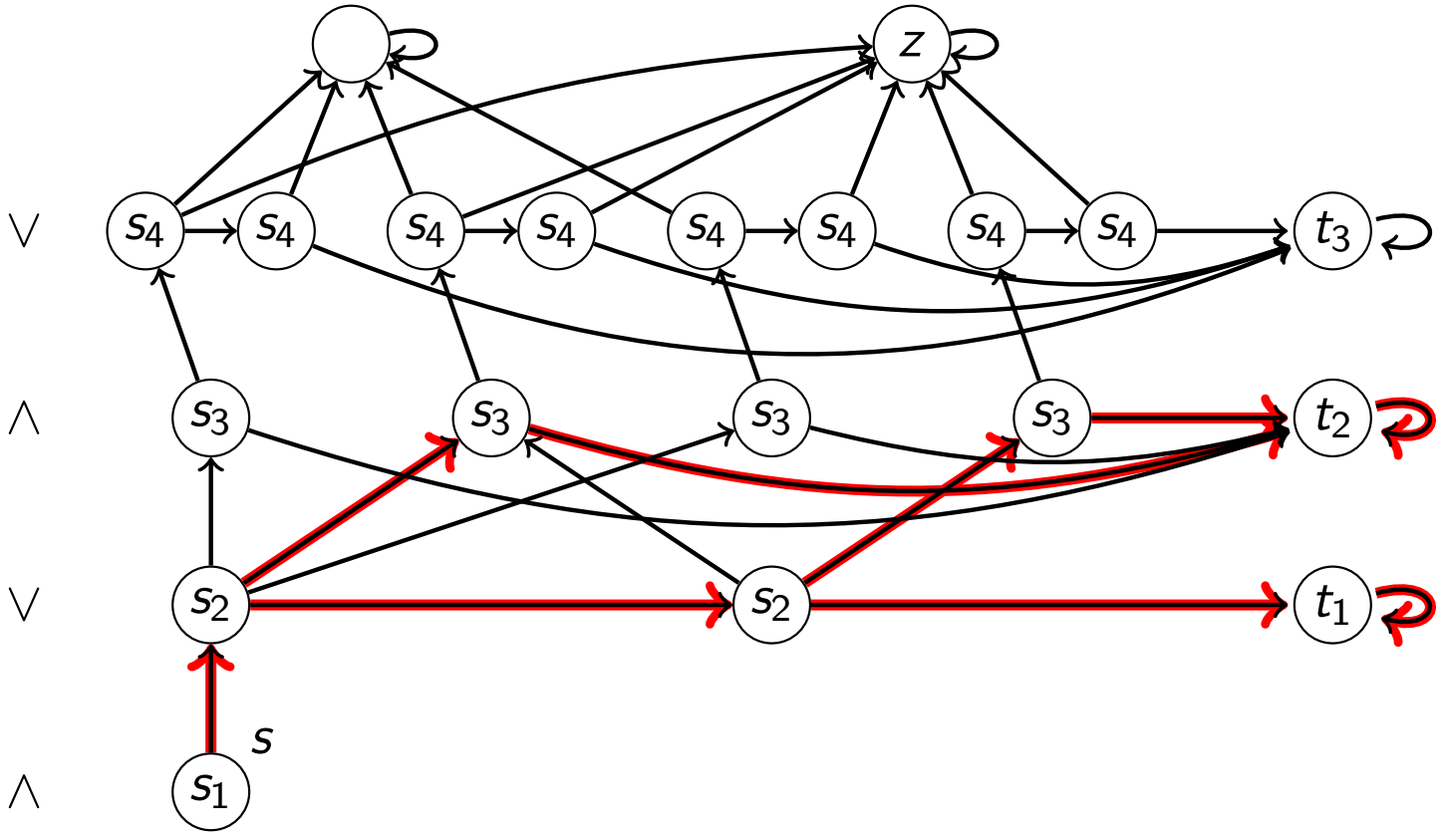
$$\mathrm{EG}(s_1 \vee t_1 \vee (s_2 \wedge \mathrm{EG}(s_2 \vee t_2 \vee (s_3 \wedge))))$$

Was „doppelte“ CTL-Operatoren können



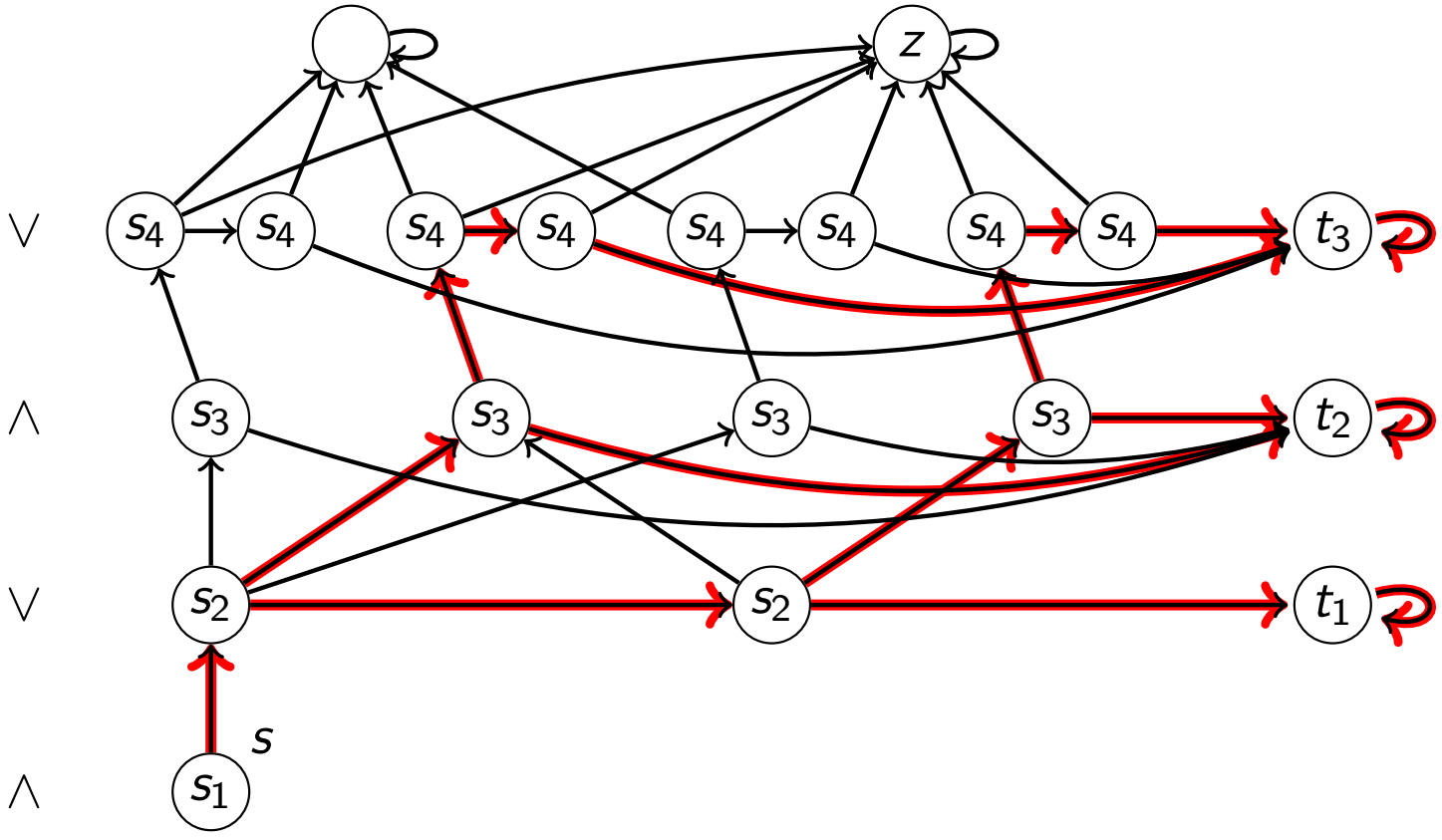
$$\mathrm{EG}(s_1 \vee t_1 \vee (s_2 \wedge \mathrm{EG}(s_2 \vee t_2 \vee (s_3 \wedge))))$$

Was „doppelte“ CTL-Operatoren können



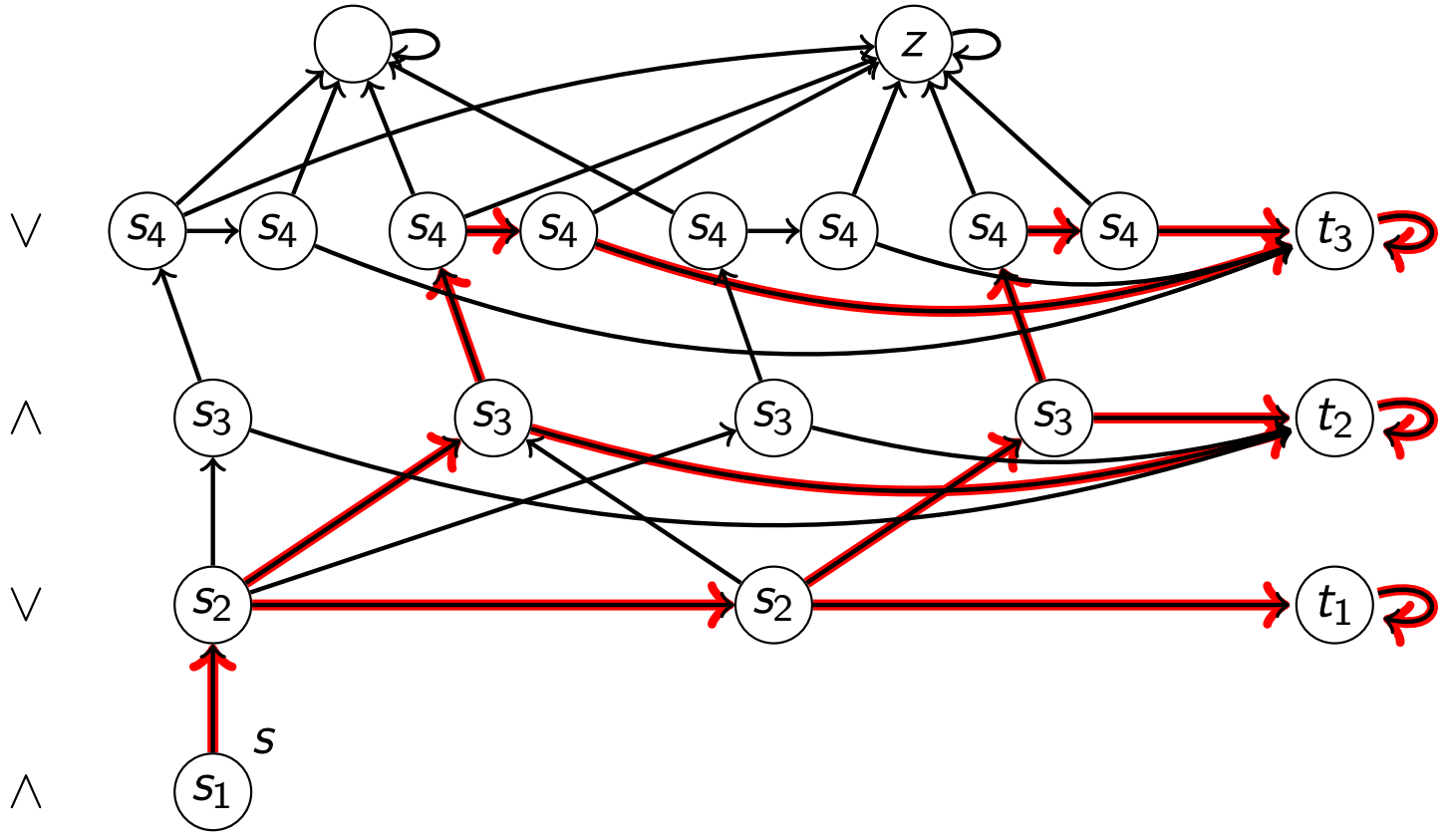
$$\mathrm{EG}(s_1 \vee t_1 \vee (s_2 \wedge \mathrm{EG}(s_2 \vee t_2 \vee (s_3 \wedge \mathrm{EG}(s_3 \vee t_3 \vee (s_4 \wedge \quad))))))$$

Was „doppelte“ CTL-Operatoren können



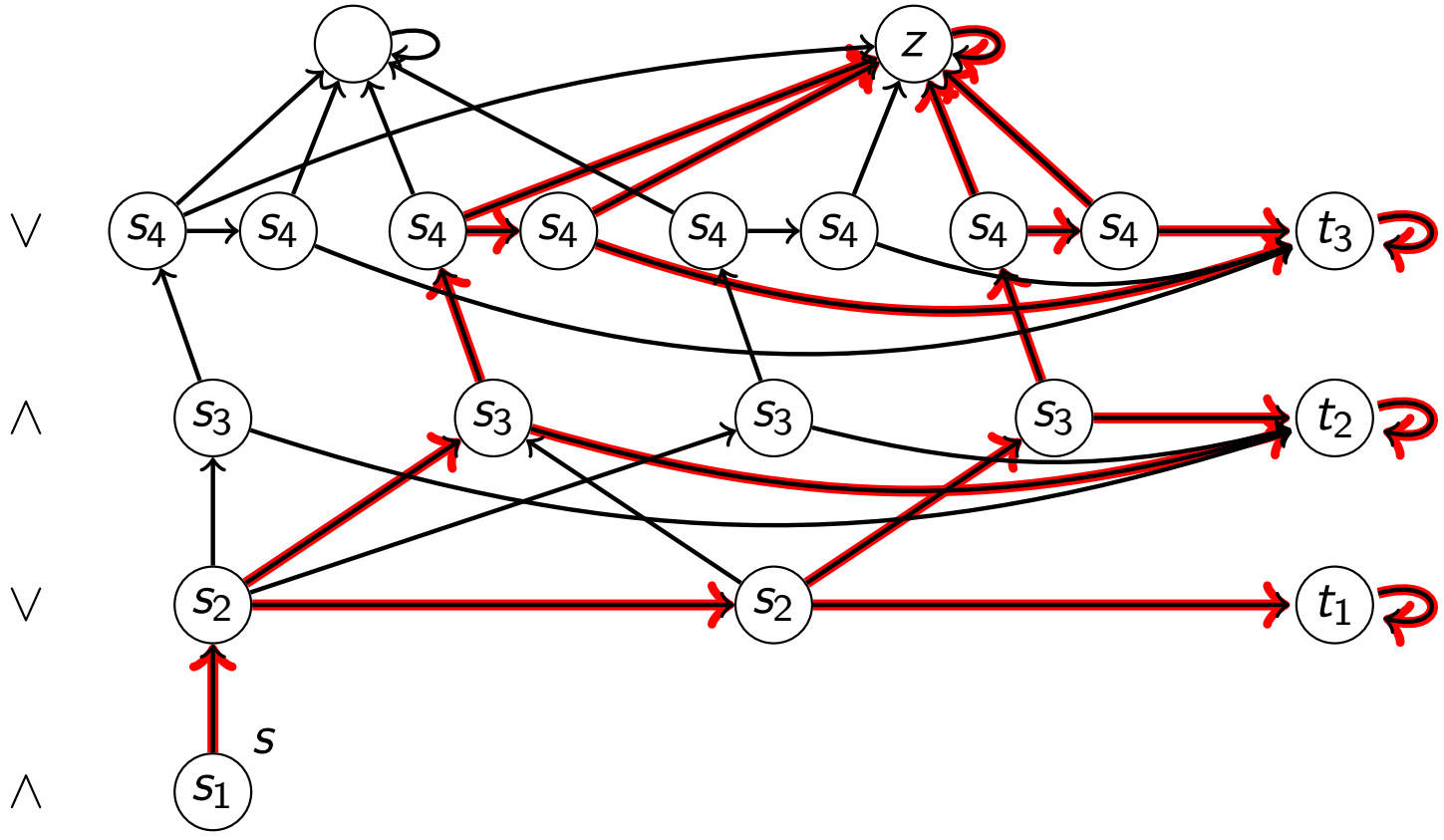
$$\mathrm{EG}(s_1 \vee t_1 \vee (s_2 \wedge \mathrm{EG}(s_2 \vee t_2 \vee (s_3 \wedge \mathrm{EG}(s_3 \vee t_3 \vee (s_4 \wedge \quad))))))$$

Was „doppelte“ CTL-Operatoren können



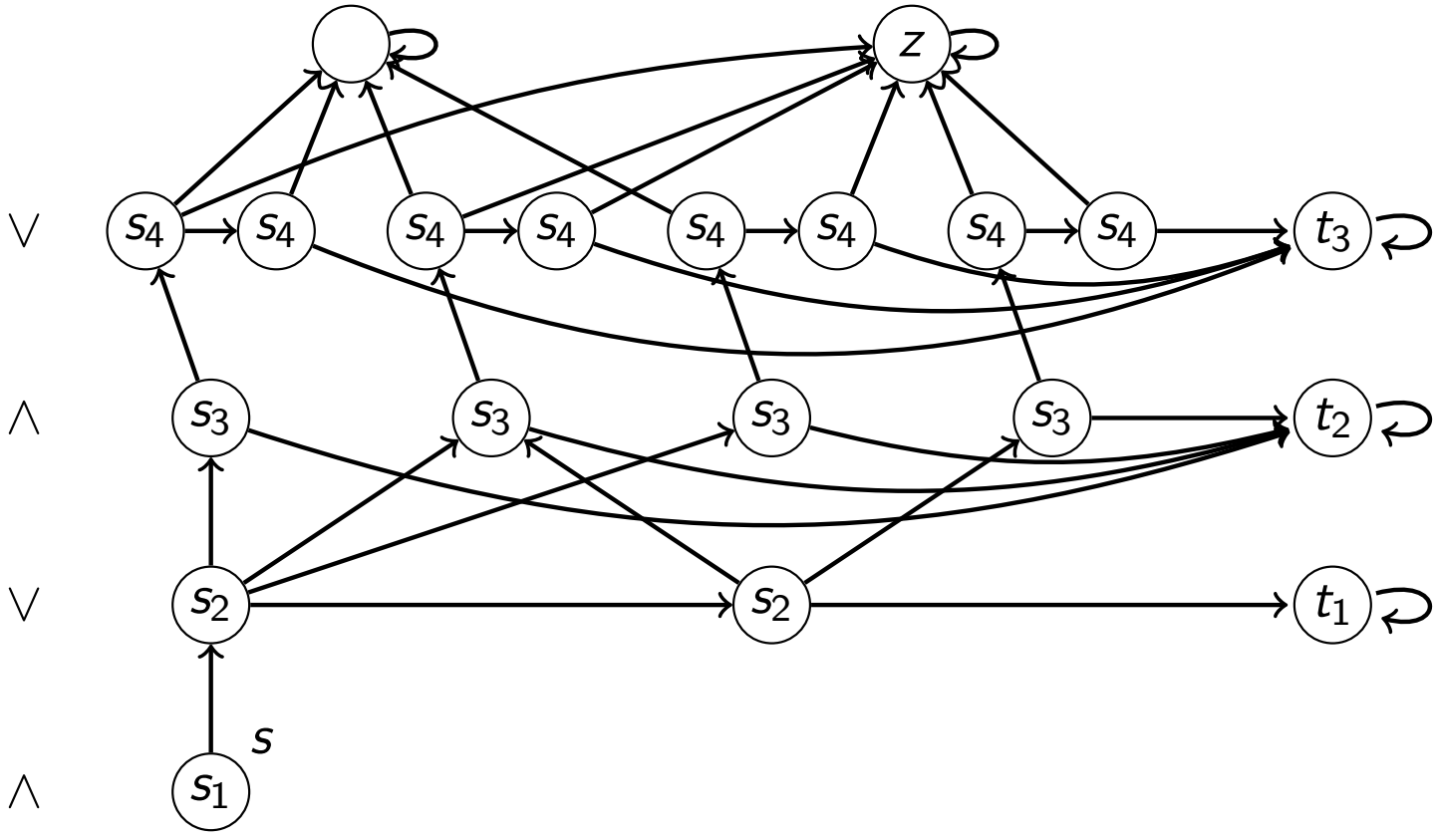
$$\mathrm{EG}(s_1 \vee t_1 \vee (s_2 \wedge \mathrm{EG}(s_2 \vee t_2 \vee (s_3 \wedge \mathrm{EG}(s_3 \vee t_3 \vee (s_4 \wedge \mathrm{EG}(s_4 \vee z)))))))$$

Was „doppelte“ CTL-Operatoren können



$$\mathrm{EG}(s_1 \vee t_1 \vee (s_2 \wedge \mathrm{EG}(s_2 \vee t_2 \vee (s_3 \wedge \mathrm{EG}(s_3 \vee t_3 \vee (s_4 \wedge \mathrm{EG}(s_4 \vee z)))))))$$

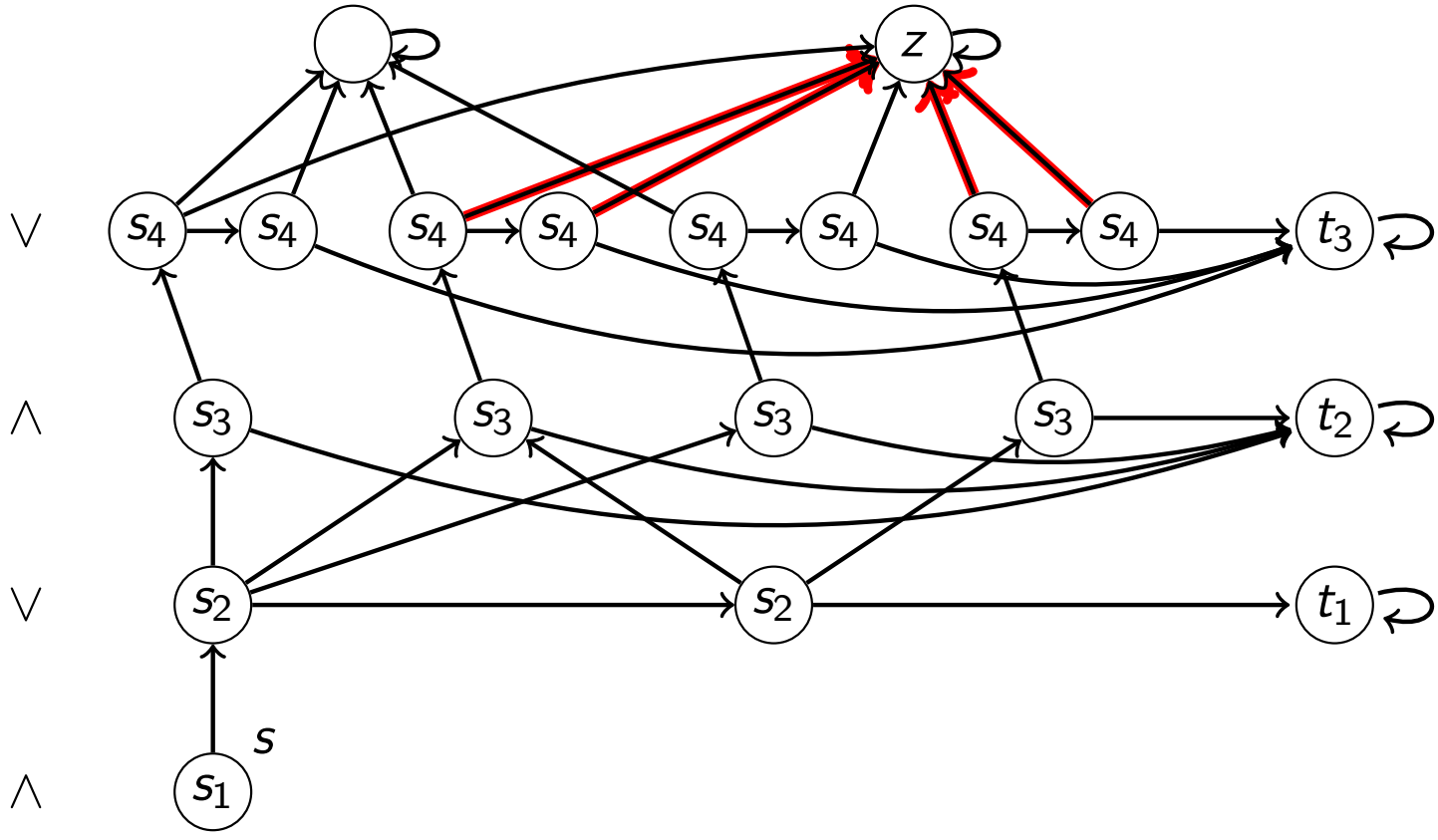
Was „doppelte“ CTL-Operatoren können



$$\mathrm{EG}(s_1 \vee t_1 \vee (s_2 \wedge \mathrm{EG}(s_2 \vee t_2 \vee (s_3 \wedge \mathrm{EG}(s_3 \vee t_3 \vee (s_4 \wedge \mathrm{EG}(s_4 \vee z)))))))$$

EU schafft's auch ohne Hilfe Boolescher Operatoren:

Was „doppelte“ CTL-Operatoren können

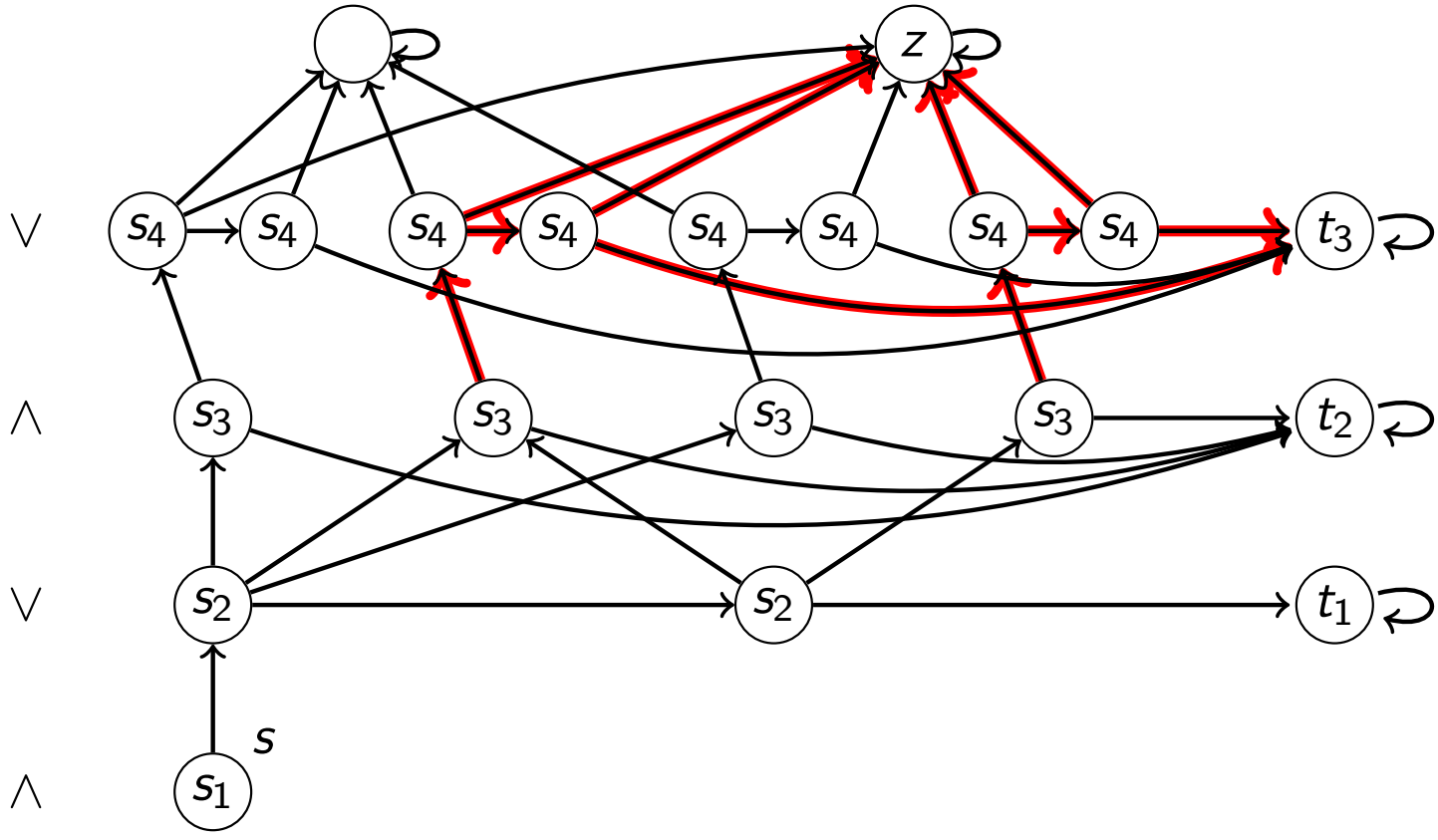


$$\mathrm{EG}(s_1 \vee t_1 \vee (s_2 \wedge \mathrm{EG}(s_2 \vee t_2 \vee (s_3 \wedge \mathrm{EG}(s_3 \vee t_3 \vee (s_4 \wedge \mathrm{EG}(s_4 \vee z)))))))$$

EU schafft's auch ohne Hilfe Boolescher Operatoren:

$$\mathrm{E}(s_4 \mathrm{U} z)$$

Was „doppelte“ CTL-Operatoren können



$$\mathrm{EG}(s_1 \vee t_1 \vee (s_2 \wedge \mathrm{EG}(s_2 \vee t_2 \vee (s_3 \wedge \mathrm{EG}(s_3 \vee t_3 \vee (s_4 \wedge \mathrm{EG}(s_4 \vee z)))))))$$

EU schafft's auch ohne Hilfe Boolescher Operatoren:

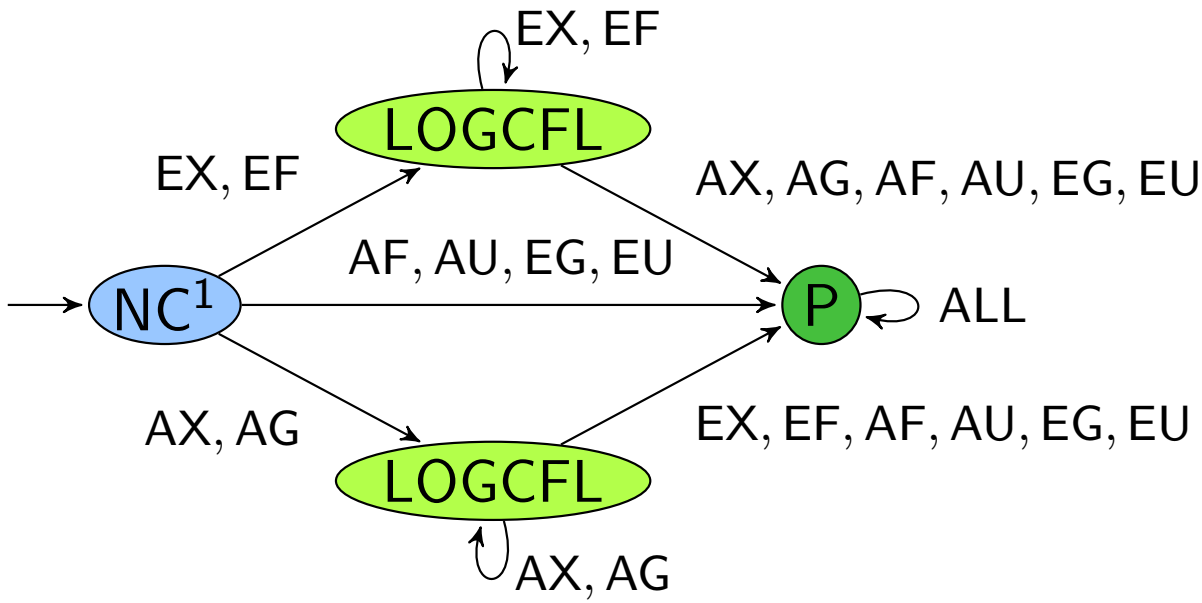
$\cdots \mathrm{E}(s_3 \ \mathrm{U} \ \mathrm{E}(\mathrm{E}(s_4 \mathrm{U} z) \ \mathrm{U} \ t_3)) \cdots$

Komplexität des Model Checking für CTL

Theorem (Beyersdorff et al 2011)

1. *Model Checking ist* P*-vollständig für alle* CTL*-Fragmente mit*
   - *konjunktiven und disjunktiven* CTL*-Operator(en) und*
   - *Konjunktion und Disjunktion* ($\wedge$, $\vee$).
2. *Model Checking ist* LOGCFL*-vollständig für alle* CTL*-Fragmente mit*
   - *nur-konjunktiven oder nur-disjunktiven* CTL*-Operator(en) und*
   - *Konjunktion und Disjunktion* ($\wedge$, $\vee$).

Komplexität des Model Checking für CTL mit $\wedge, \vee$



CTL^+ Model Checking

CTL^+:

- Formeln mit alternierenden Pfad-Quantoren und temporalen Operatoren

$$\mathsf{E}(\mathsf{G}x \to (y \mathsf{U} z))$$

Theorem (Laroussinie, Markey, Schnoebelen 2001)

CTL^+ *Model Checking ist* Δ_2^{p}*-vollständig.*

Theorem (Meier 2011)

CTL^+ *Model Checking für Fragmente mit den Operatoren* T *ist*

- NC^1*-vollständig, falls* $T \subseteq \{\mathsf{A}, \mathsf{E}\}$,
- P*-vollständig, falls* $\{\mathsf{X}\} \subsetneq T \subseteq \{\mathsf{A}, \mathsf{E}, \mathsf{X}\}$, *und*
- Δ_2^{p}*-vollständig sonst.*

CTL^+ Model Checking mit $\wedge, \vee$

Theorem (Meier 2011)

CTL^+ *Model Checking für Fragmente aus* $\wedge, \vee$ *und den Operatoren* T *ist*

- NC^1*-vollständig, falls* $T \subseteq \{\mathrm{A}, \mathrm{E}\}$,
- LOGCFL*-vollständig, falls* $T = \{\mathrm{A}, \mathrm{X}\}$ *oder* $T \subseteq \{\mathrm{E}, \mathrm{X}\}$,
- P*-vollständig, falls* $T = \{\mathrm{A}, \mathrm{E}, \mathrm{X}\}$,
- NP*-vollständig, falls* $T = \{\mathrm{E}, Y\}$ *mit* $Y \in \{\mathrm{F}, \mathrm{G}, \mathrm{U}\}$
- coNP*-vollständig, falls* $T = \{\mathrm{A}, Y\}$, *und*
- Δ_2^{p}*-vollständig sonst.*

LTL Model Checking

LTL: linear-time temporal logic

- Formeln mit temporalen Operatoren, ohne Pfad-Quantoren

$$\mathsf{F}(\mathsf{G}x \rightarrow y\mathsf{U}z)$$

- Modell erfüllt Formel, wenn sie auf einem Pfad erfüllt wird.

Theorem (Sistla und Clarke, 1985)

Model Checking ist PSPACE*-vollständig für* LTL*-Fragmente mit*

- F *und* X*, oder*
- G *und* X*, oder*
- U*,*
- *und allen Booleschen Operatoren.*

Ein PSPACE-vollständiges Domino-Problem

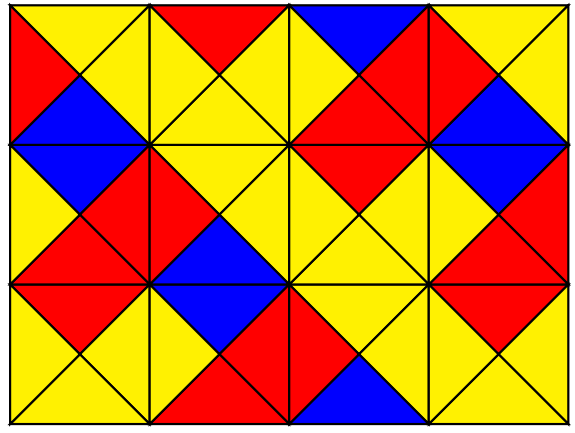
gegeben: Zahl n, Domino-Steine *ul* und *or*, Domino-Typen

gefragt: kann ein $n \times m$ Abschnitt der Ebene mit zueinander passenden Domino-Steinen bedeckt werden, sodass *ul* unten links und *or* oben rechts liegt?

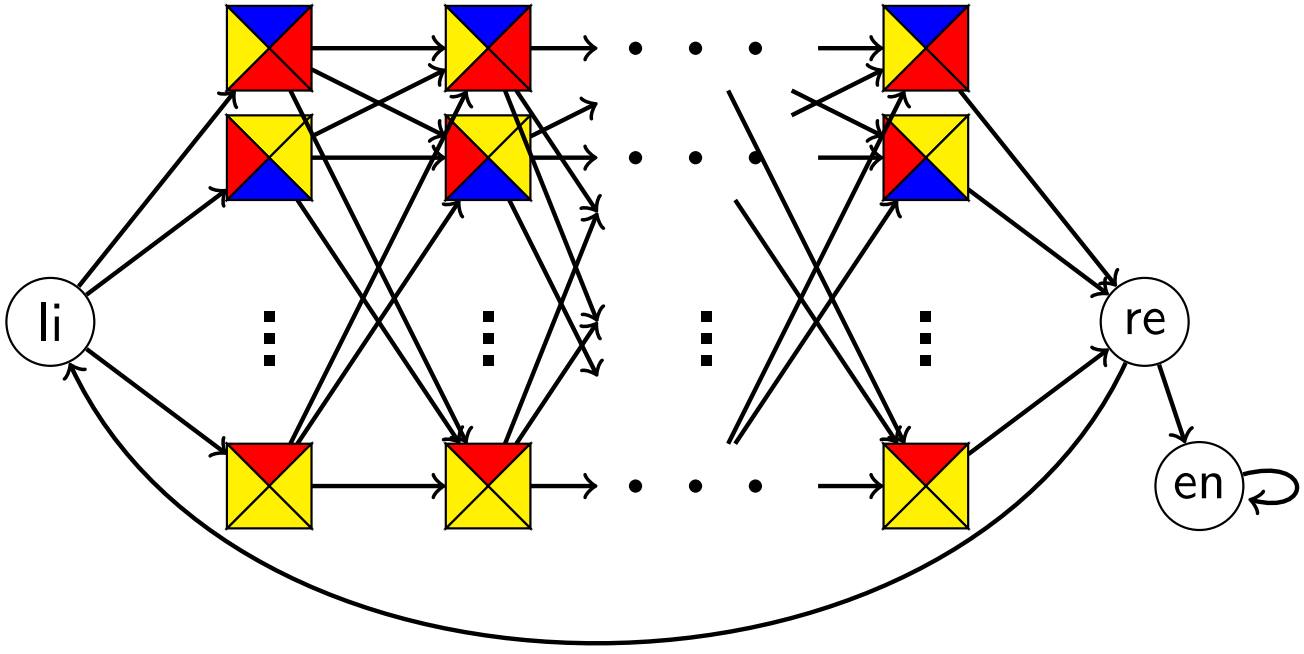
Beispiel: $n = 4$,

ul *or*

Lösung:

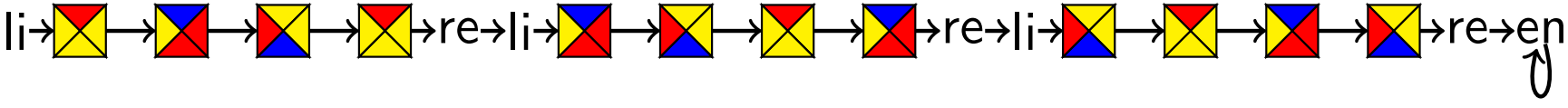


Domino reduziert zu LTL-MC

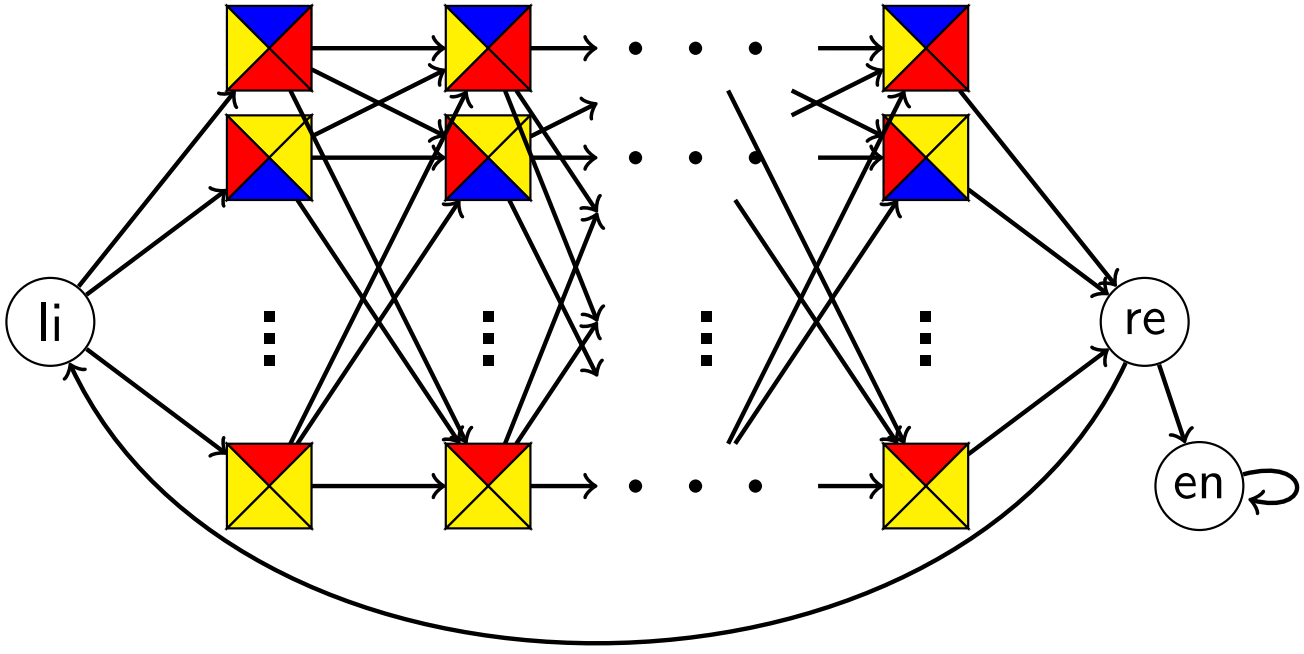


Jeder Pfad, der eine korrekte Bedeckung darstellt, erfüllt

$$\mathsf{X}\,\mathsf{ul} \;\wedge\; \mathsf{G}(\neg\mathsf{X}(\mathsf{re} \wedge \mathsf{X}\,\mathsf{en}) \vee \mathsf{or}) \;\wedge$$
$$\mathsf{G}[\mathsf{li} \vee \mathsf{re} \vee \mathsf{en} \vee$$
$$(\bigvee_c (\mathit{oben}.c \wedge \mathsf{X}^{n+2}\,\mathit{unten}.c) \wedge \bigvee_c (\mathit{rechts}.c \wedge \mathsf{X}\,\mathit{links}.c))\;]$$

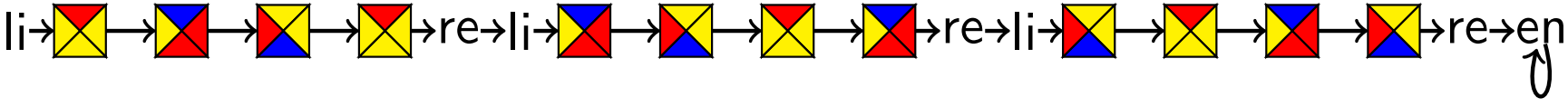


Domino reduziert zu LTL-MC



Jeder Pfad, der eine korrekte Bedeckung darstellt, erfüllt

$$\mathsf{X}\,\mathsf{ul} \;\wedge\; \mathsf{G}(\mathsf{X}(\neg \mathsf{re} \wedge \mathsf{X}\,\neg \mathsf{en}) \vee \mathsf{or}) \;\wedge$$
$$\mathsf{G}[\mathsf{li} \vee \mathsf{re} \vee \mathsf{en} \;\vee$$
$$\left(\bigvee\nolimits_c (\mathit{oben}.c \wedge \mathsf{X}^{n+2}\, \mathit{unten}.c) \wedge \bigvee\nolimits_c (\mathit{rechts}.c \wedge \mathsf{X}\, \mathit{links}.c)\right)]$$



Theorem (Sistla und Clarke, 1985, bzw. Folgerungen)

Model Checking ist PSPACE*-vollständig für* LTL*-Fragmente mit*

- G *und* X *sowie* $\wedge$ *und* $\vee$,
- U *sowie* $\wedge$ *und* $\vee$.

Theorem (Sistla und Clarke, 1985)

Model Checking ist NP*-vollständig für das* LTL*-Fragment mit*

- F *sowie* $\wedge$.

Theorem (Bauland et al, 2011)

Model Checking ist NP*-hart für* LTL*-Fragmente mit*

- G *und* X *sowie* $\vee$,
- U *ohne Booleschem Operator.*

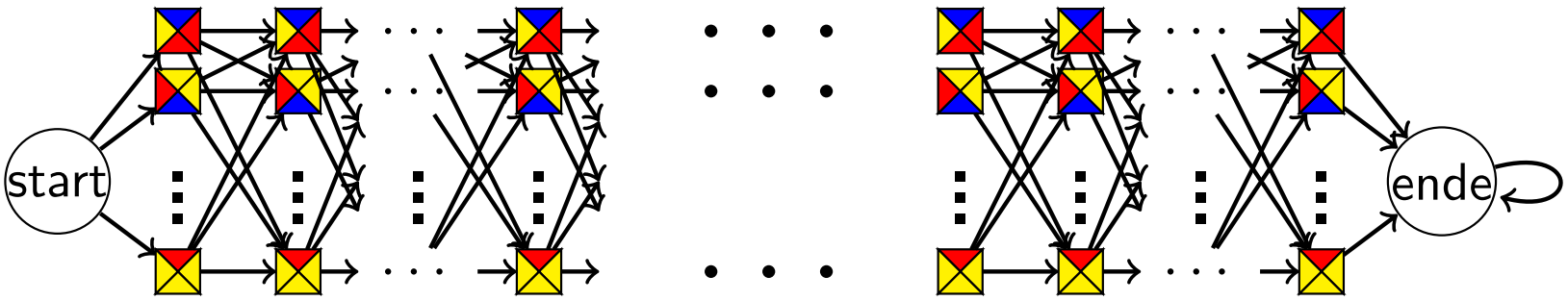
Model Checking ist NL*-vollständig für die* LTL*-Fragmente mit*

- G *sowie* $\vee$,
- G *und* X *sowie* $\wedge$.

Ein NP-vollständiges Domino-Problem

gegeben: Zahl n, Domino-Steine *ul* und *or*, Domino-Typen

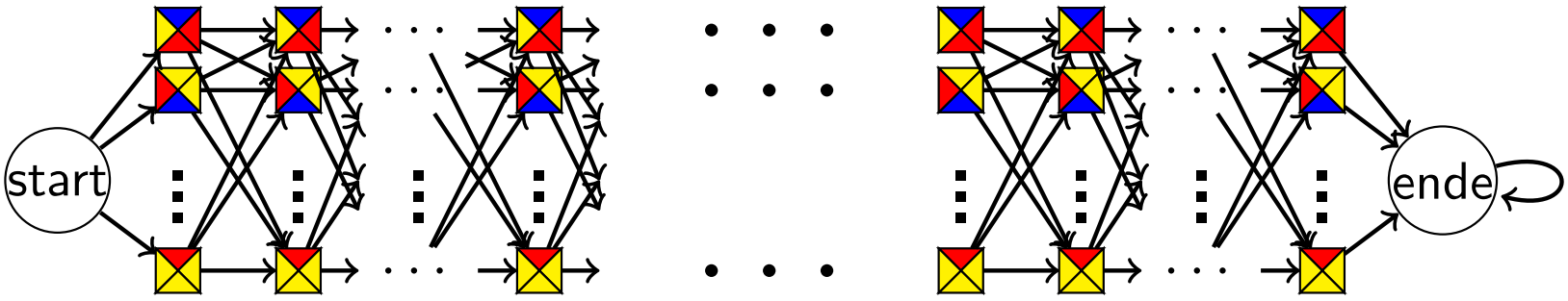
gefragt: kann ein $n \times n$ Abschnitt der Ebene mit zueinander passenden Domino-Steinen bedeckt werden, sodass *ul* unten links und *or* oben rechts liegt?



Ein NP-vollständiges Domino-Problem

gegeben: Zahl n, Domino-Steine *ul* und *or*, Domino-Typen

gefragt: kann ein $n \times n$ Abschnitt der Ebene mit zueinander passenden Domino-Steinen bedeckt werden, sodass *ul* unten links und *or* oben rechts liegt?

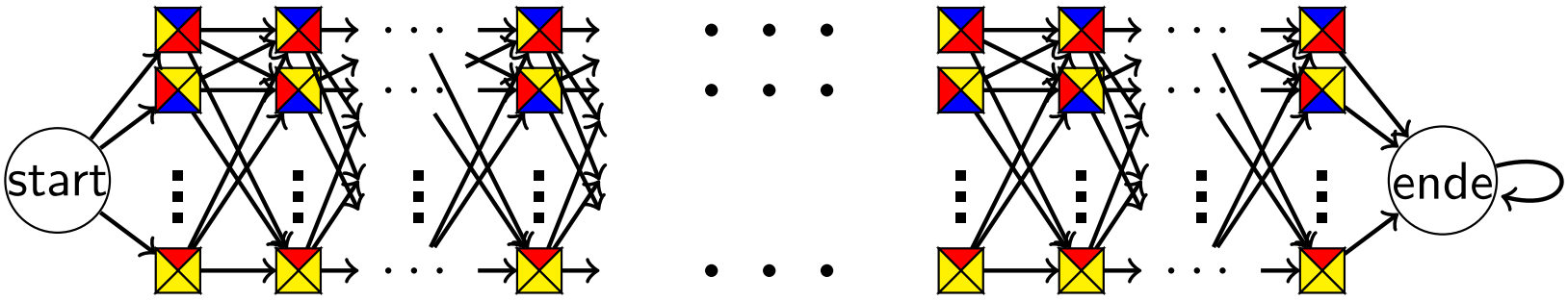


$$\bigwedge_{pos} \mathrm{X}^{pos} \bigvee_{c} \bigvee_{c'} [(\mathit{rechts}.c \wedge \mathrm{X}\ \mathit{links}.c) \wedge (\mathit{oben}.c' \wedge \mathrm{X}^{n}\ \mathit{unten}.c'))]$$

Ein NP-vollständiges Domino-Problem

gegeben: Zahl n, Domino-Steine *ul* und *or*, Domino-Typen

gefragt: kann ein $n \times n$ Abschnitt der Ebene mit zueinander passenden Domino-Steinen bedeckt werden, sodass *ul* unten links und *or* oben rechts liegt?



$$\mathsf{G} \bigvee_{pos} \bigvee_{c} \bigvee_{c'} [pos \wedge (rechts.c \wedge \mathsf{F}(pos + 1 \wedge links.c)) \wedge (oben.c' \wedge \mathsf{F}(pos + n \wedge unten.c'))]$$

Theorem (Sistla und Clarke, 1985, bzw. Folgerungen)

Model Checking ist NP*-vollständig für* LTL*-Fragmente mit*

- F *und* $\wedge$,
- G *sowie* $\wedge$ *und* $\vee$,
- X *sowie* $\wedge$ *und* $\vee$,
- F *und* X *sowie* $\wedge$ *und* $\vee$,
- F *und* G *sowie* $\wedge$ *und* $\vee$.

Theorem (Sistla und Clarke, 1985, bzw. Folgerungen)

Model Checking ist PSPACE*-vollständig für* LTL*-Fragmente mit*

- G *und* X *sowie* $\wedge$ *und* $\vee$,
- U *sowie* $\wedge$ *und* $\vee$.

Theorem (Bauland et al, 2011)

Model Checking ist NP*-hart für* LTL*-Fragmente mit*

- G *und* X *sowie* $\vee$,
- U *ohne Booleschem Operator.*

Theorem (Sistla und Clarke, 1985, bzw. Folgerungen)

Model Checking ist NP*-vollständig für* LTL*-Fragmente mit*

- F *und* $\wedge$,
- G *sowie* $\wedge$ *und* $\vee$,
- X *sowie* $\wedge$ *und* $\vee$,
- F *und* X *sowie* $\wedge$ *und* $\vee$,
- F *und* G *sowie* $\wedge$ *und* $\vee$.

Theorem (Sistla und Clarke, 1985, bzw. Folgerungen)

Model Checking ist PSPACE*-vollständig für* LTL*-Fragmente mit*

- G *und* X *sowie* $\wedge$ *und* $\vee$,
- U *sowie* $\wedge$ *und* $\vee$.

Theorem (Bauland et al, 2011)

Model Checking ist NL*-vollständig für die* LTL*-Fragmente mit*

- G *sowie* $\vee$,
- G *und* X *sowie* $\wedge$.

CTL* Model Checking

CTL*:

- Formeln mit Pfad-Quantoren und temporalen Operatoren

$$\mathsf{A}(\mathsf{F}x \wedge \mathsf{E}(\mathsf{GF}x \rightarrow (y\mathsf{U}z)))$$

Theorem (Clarke, Emerson, Sistla 1986)

CTL* *Model Checking ist* PSPACE-*vollständig.*

Die Komplexität des Model Checking für Fragmente mit nur 2 temporalen Operatoren ist noch weitgehend offen ...

Ende

- vollständige Probleme für viele Komplexitätsklassen
- nicht immer sind Strukturen erkennbar
- viele offene obere Schranken (insbesondere für U)
- fixed-parameter complexity

Ende

- vollständige Probleme für viele Komplexitätsklassen
- nicht immer sind Strukturen erkennbar
- viele offene obere Schranken (insbesondere für U)
- fixed-parameter complexity

(Dank an Arne Meier (Hannover) für Bilder in tikz)