Wie schwer ist CTL Model Checking wirklich?

Leo Kayser & Tobias Brockmeyer

31. Mai 2022

Inhalt

Einige Klassen in P

Wir betrachten die folgenden Komplexitätsklassen

$$\mathrm{P} \supseteq \mathrm{AC}^1 \supseteq \mathrm{LOGCFL} \supseteq \mathrm{NL}.$$

1. $\mathrm{P} = \mathrm{TIME}(n^{O(1)})$
   - $= \mathrm{ALOGSPACE}$
2. $\mathrm{AC}^1 = \mathrm{SIZE{-}DEPTH}(n^{O(1)}, \log n)$ (log-uniform)
   - $= \mathrm{ALOGSPACE}$ mit logarithmisch vielen Alternierungen
3. $\mathrm{LOGCFL} = \{\, A \mid A \leq_{\mathrm{m}}^{\log} B,\ B \text{ kontextfrei} \,\}$
   - Entscheidbar von NL-Maschine mit Hilfskeller in poly. Zeit
   - $= \mathrm{ALOGSPACE}$ mit polynomiell beschränkter Baumgröße
4. $\mathrm{NL} = \mathrm{NSPACE}(\log n)$

Mind the GAP

Definition

Problem	GAP (Graph Accessibility Problem)
Instanzen	Gerichtete Graphen $G = (V, E)$, Knoten $s, t \in V$
Frage	Gibt es in G einen Pfad von s nach t?

- Auch bekannt als PATH, STCON (s-t-connection)
- GAP $\in \mathrm{NL}$: Rate Pfad einen Schritt nach dem anderen; nach $|V|$ erfolglosen Schritten verwerfe
- GAP ist NL-schwer: G Konfigurationsgraph der NTM, s ist Startkonfiguration. t ist Endkonfiguration, $\mathrm{NSPACE}(\log n)$ stellt sicher, dass Graph polynomiell groß
- Analoges Problem für ungerichtete Graphen: UGAP $\in \mathrm{L}$

Alternierende Graphen

Definition

Ein *alternierender Graph* ist ein gerichteter bipartiter Graph $G = (V_\exists \dot\cup V_\forall, E)$, d.h. $E \subseteq (V_\exists \times V_\forall) \cup (V_\forall \times V_\exists)$.

Die Relation $\text{apath}_G(s, T)$ für $s \in V$, $T \subseteq V$ ist erfüllt, falls

(0) $s \in T$, oder

(1) $s \in V_\exists$ und $\exists (s, y) \in E$: $\text{apath}_G(y, T)$, oder

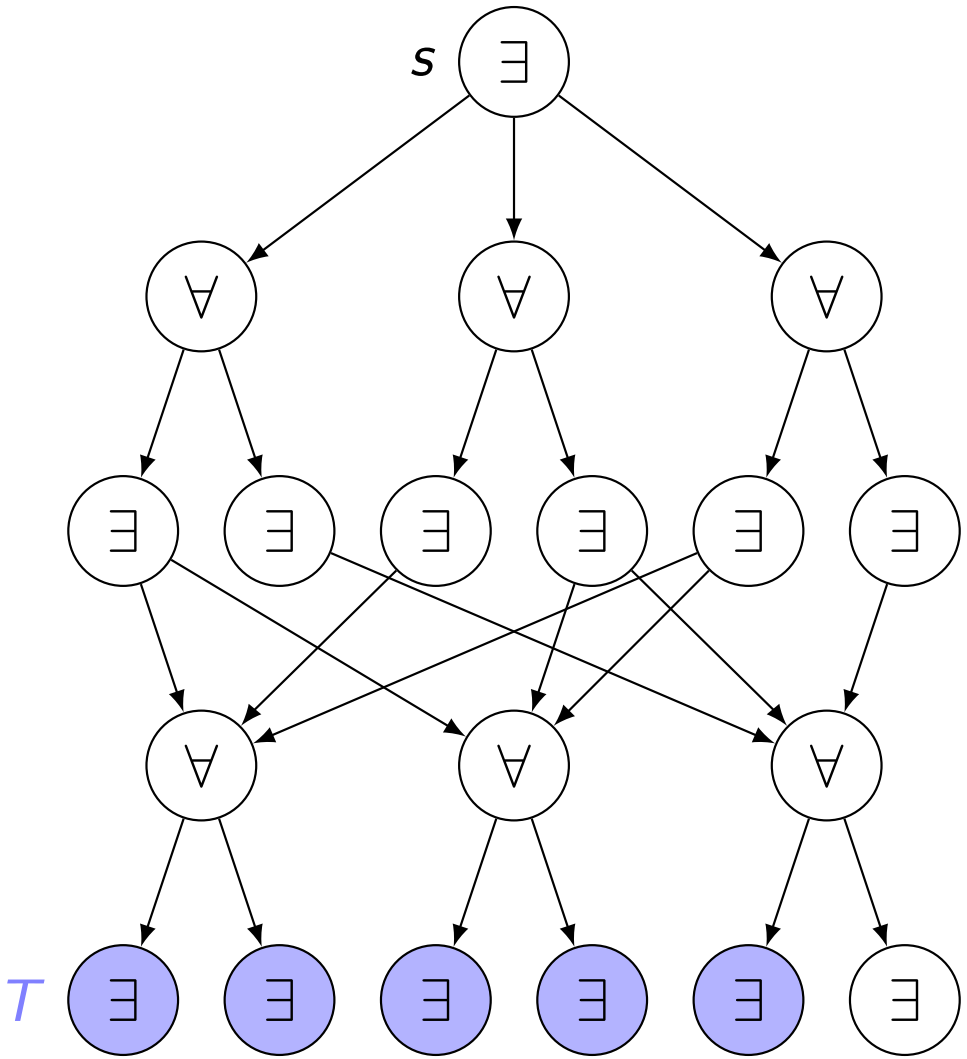
(2) $s \in V_\forall$ und $\forall (s, y) \in E$: $\text{apath}_G(y, T)$.

Definition

Problem	AGAP (Alternating GAP)
Instanzen	Alternierende Graphen $G = (V_\exists \cup V_\forall, E)$, Knoten $s \in V$, $T \subseteq V$
Frage	Gilt $\text{apath}_G(s, T)$?

Ein Beispielgraph

Gilt $\mathrm{apath}_G(s, T)$?



Alternierende Graphen mit Schichten

Definition

Ein *alternierender geschichteter Graph* ist ein alternierender Graph mit einer Partition $V = V_0 \dot\cup \cdots \dot\cup V_m$, sodass

$$V_\exists = \bigcup_{i \text{ gerade}} V_i, \quad V_\forall = \bigcup_{i \text{ ungerade}} V_i, \quad E \subseteq \bigcup_{i=1}^{m} (V_{i-1} \times V_i).$$

Wir nehmen zudem $\deg_{\text{out}}(v) > 0$ an für $v \in V \setminus V_m$.

Definition

Problem ASGAP (Alternating slice GAP)

Instanzen Alternierende Graphen $G = (V_\exists \cup V_\forall, E)$ mit m Schichten, m gerade, $s \in V_0$, $T \subseteq V_m$.

Frage Gilt $\text{apath}_G(s, T)$?

GAP-Varianten liefern vollständige Probleme

- ASGAP$(\forall_{\text{out}} = 2, \exists_{\text{in}} = 1)$: Einschränkung an G:
  - $\deg_{\text{out}}(v) = 2$ für $v \in V_\forall$
  - $\deg_{\text{in}}(v) = 1$ für $v \in V_\exists \setminus V_0$
- $\text{ASGAP}_{\log}$: Einschränkung logarithmischer Tiefe: $m \leq \log |V|$

Satz

Wir haben folgende vollständige Probleme bzgl. $\leq_{\text{m}}^{\log}$:

Klasse	Vollständige Probleme
P	AGAP, ASGAP, ASGAP$(\forall_{\text{out}} = 2, \exists_{\text{in}} = 1)$
AC^1	$\text{ASGAP}_{\log}$
LOGCFL	ASGAP$(\forall_{\text{out}} = 2, \exists_{\text{in}} = 1)_{\log}$
NL	GAP

Syntax und Semantik der Computational Tree Logic

Syntax: Sei PROP eine abzählbare Menge von Symbolen.

$$\varphi ::= \top \mid p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \oplus \varphi \mid \neg\varphi \mid \mathcal{O}\varphi \mid \varphi\mathcal{O}'\varphi$$

$p \in \mathrm{PROP},\ \mathcal{O} \in \{\mathsf{EX}, \mathsf{AX}, \mathsf{EG}, \mathsf{AG}, \mathsf{EF}, \mathsf{AF}\},\ \mathcal{O}' \in \{\mathsf{EU}, \mathsf{AU}, \mathsf{ER}, \mathsf{AR}\}$

Semantik (Auszug): Es sei $\mathcal{M} = (\mathcal{F}, \xi)$ ein Modell zum *totalen* Rahmen $\mathcal{F} = (W, R)$ und $w \in W$.
Sei Π die Menge der unendlichen Pfade beginnend bei w.

- $\mathcal{M}, w \models \mathsf{EX}\,\varphi$ gdw. $\exists \pi = (\pi_1, \pi_2, \dots) \in \Pi$: $\mathcal{M}, \pi_2 \models \varphi$
- $\mathcal{M}, w \models \mathsf{EF}\,\varphi$ gdw. $\exists \pi \in \Pi\ \exists k \geq 1$: $\mathcal{M}, \pi_k \models \varphi$
- $\mathcal{M}, w \models \mathsf{EG}\,\varphi$ gdw. $\exists \pi \in \Pi\ \forall k \geq 1$: $\mathcal{M}, \pi_k \models \varphi$
- $\mathsf{AX}\,\varphi \equiv \neg\mathsf{EX}\,\neg\varphi$, $\mathsf{AF}\,\varphi \equiv \neg\mathsf{EG}\,\neg\varphi$, $\mathsf{AG}\,\varphi \equiv \neg\mathsf{EF}\,\neg\varphi$
- $\mathcal{M}, w \models \psi\,\mathsf{EU}\,\varphi$ gdw. $\exists \pi \in \Pi\ \exists k \geq 1$: $\mathcal{M}, \pi_k \models \varphi$ und $\forall i < k$: $\mathcal{M}, \pi_i \models \psi$

Das Model Checking Problem

Definition

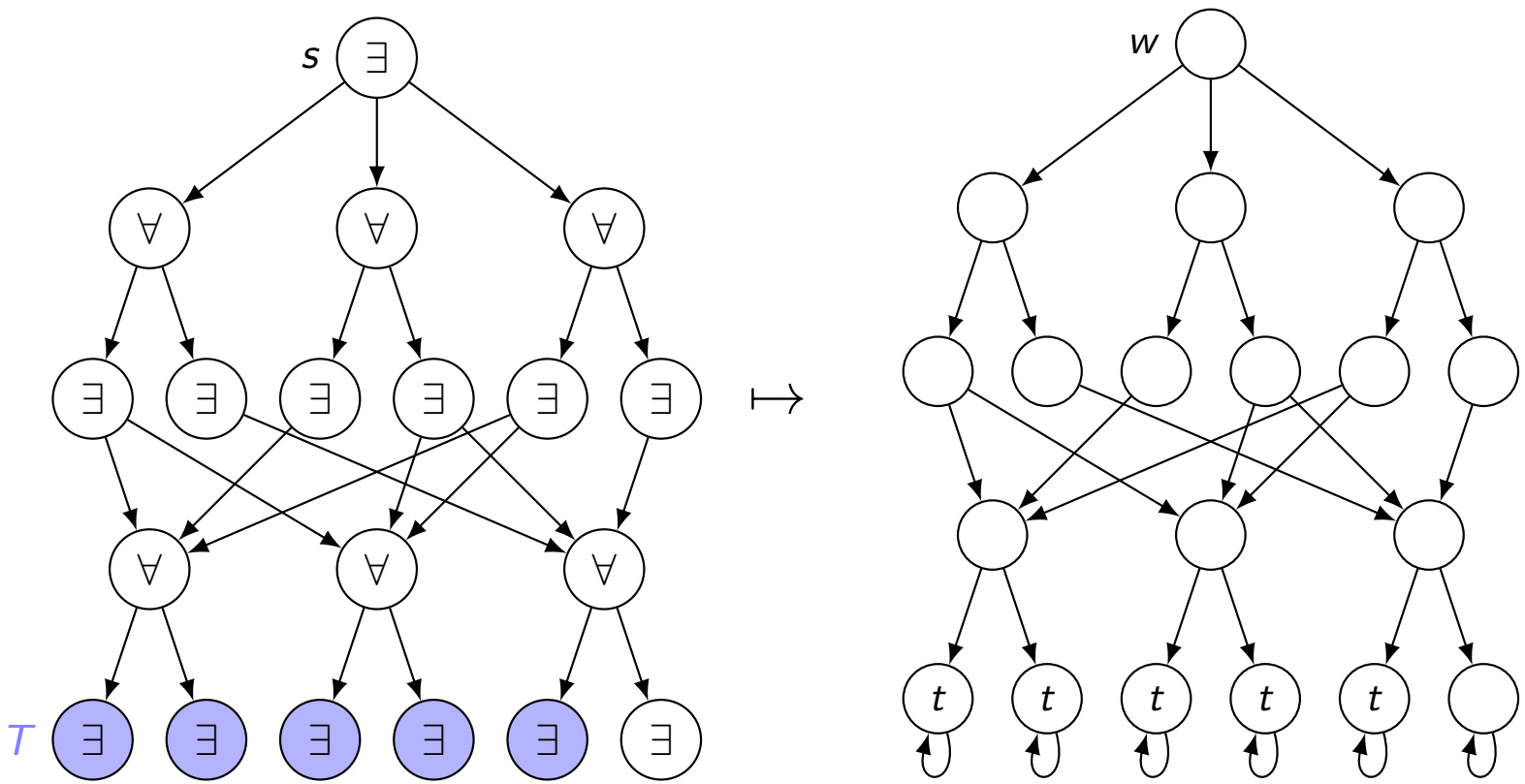
Problem	CTL-MC(T)
Instanzen	CTL-Formel ϕ mit Operatoren aus $T \subseteq \{\wedge, \vee, \oplus, \neg, \mathsf{EX}, \ldots, \mathsf{AR}\}$, ein Kripkemodell $\mathcal{M} = ((W, R), \xi)$, eine Welt $w_0 \in W$.
Frage	Gilt $\mathcal{M}, w_0 \models \phi$?

- CTL-MC(T) $\in \mathrm{P}$ für beliebiges T
- CTL-MC(T) ist P-vollständig wenn alle Operatoren erlaubt
- Klassifikation mittels Post'schen Clones? $\rightsquigarrow$ später mehr
- Beispielhaft: ASGAP($\forall_{\mathsf{out}} = 2, \exists_{\mathsf{in}} = 1$) $\leq_{\mathsf{m}}^{\log}$ CTL-MC(EX, AX)

Beweis. $\mathrm{ASGAP}(\forall_{\mathrm{out}} = 2, \exists_{\mathrm{in}} = 1) \leq_{\mathrm{m}}^{\log} \text{CTL-MC}(\mathrm{EX}, \mathrm{AX})$

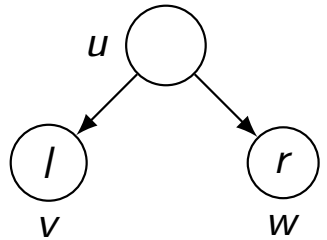
$\langle G = (V, E), s, T \rangle \mapsto \langle \mathcal{M} = ((V, E'), \xi), s, \phi \rangle$, wobei

- $E' = E \cup \{ (v, v) \mid v \in V_m \}$
- $\xi(w) = \{t\}$ wenn $w \in T$, $\emptyset$ sonst
- $\phi = \mathrm{EX}\,\mathrm{AX}\,\mathrm{EX} \cdots \mathrm{AX}\, t$

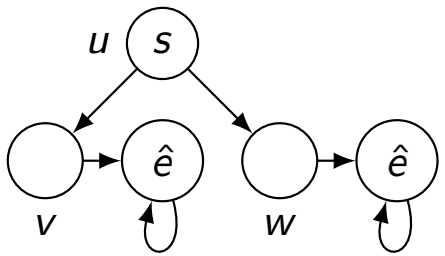


Ausblick: Wie können wir EX und AX simulieren?

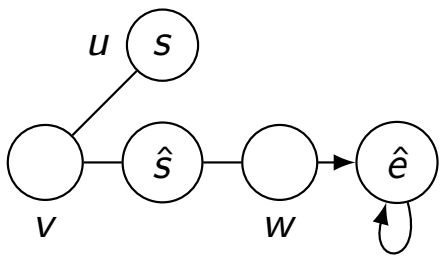
- AX mit $\{\mathsf{EX}, \wedge\}$: $\mathcal{M}', u \models \mathsf{EX}(\alpha \wedge l) \wedge \mathsf{EX}(\alpha \wedge r)$



- EX mit $\{\mathsf{EU}\}$: $\mathcal{M}', u \models s\,\mathsf{EU}\,((\hat{s}\,\mathsf{EU}\,\alpha)\,\mathsf{EU}\,\hat{e})$



- AX mit $\{\mathsf{EU}\}$: $\mathcal{M}', u \models s\,\mathsf{EU}\,((\hat{s}\,\mathsf{EU}\,\alpha)\,\mathsf{EU}\,\hat{e})$



Halbzeitpause

Was haben wir heute gelernt?

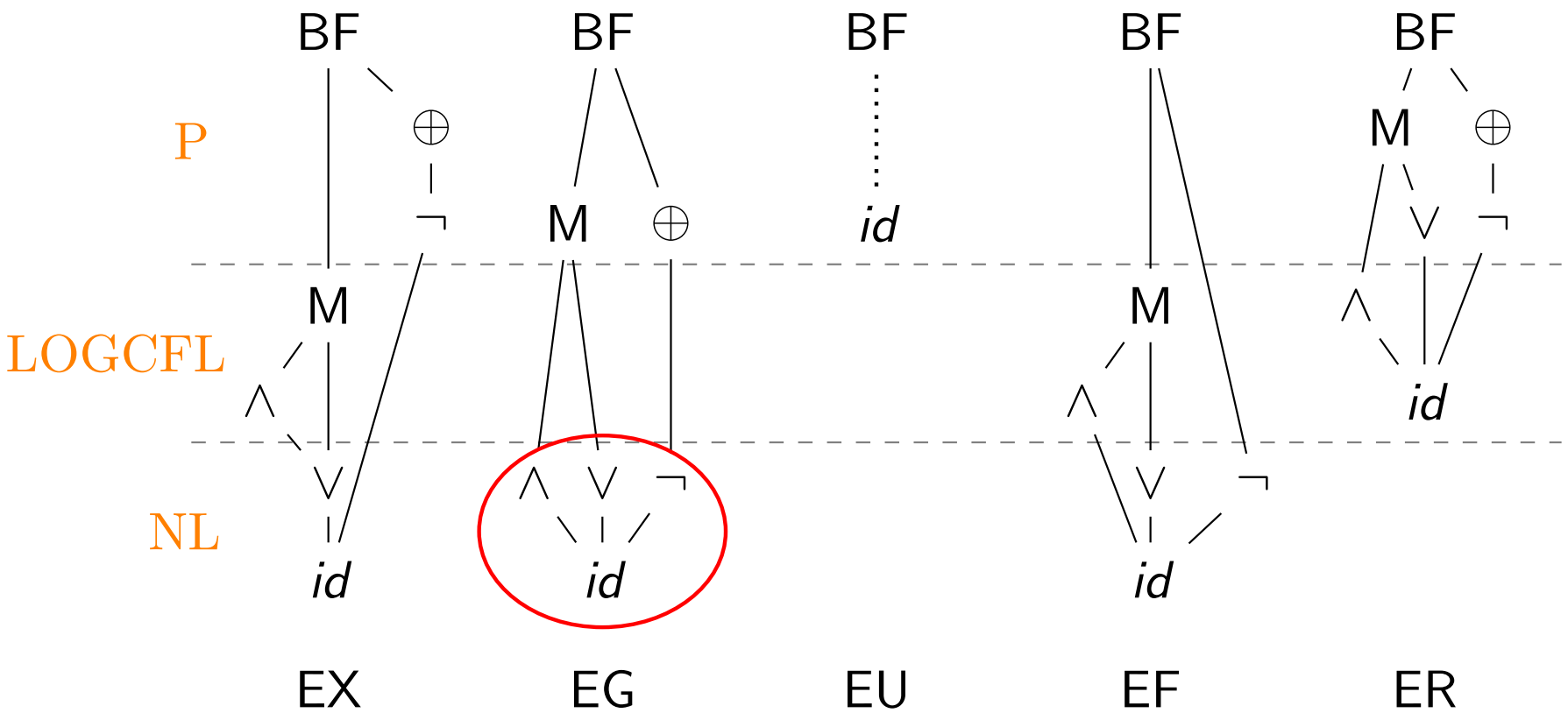
- GAP, ASGAP und verwandte Probleme sind vollständige Probleme interessanter Klassen
- Was das Model Checking Problem für CTL ist
- Wie man alternierende geschichtete Graphen in Kripkemodelle transformieren kann

Literatur:

Krebs, Meier und Mundhenk.
„The model checking fingerprints of CTL operators“
In: Acta Informatica 56.6 (2019), S. 487-519

Mundhenk und Weiß.
„The Complexity of Model Checking for Intuitionistic Logics and Their Modal Companions“
In: A. Kučera and I. Potapov (Eds.): RP 2010, LNCS 6227, pp. 146–160, 2010.

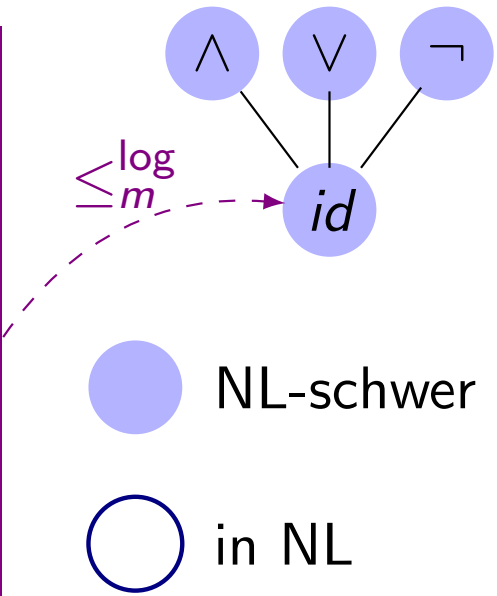
CTL-MC-Fragmente mit Konstanten



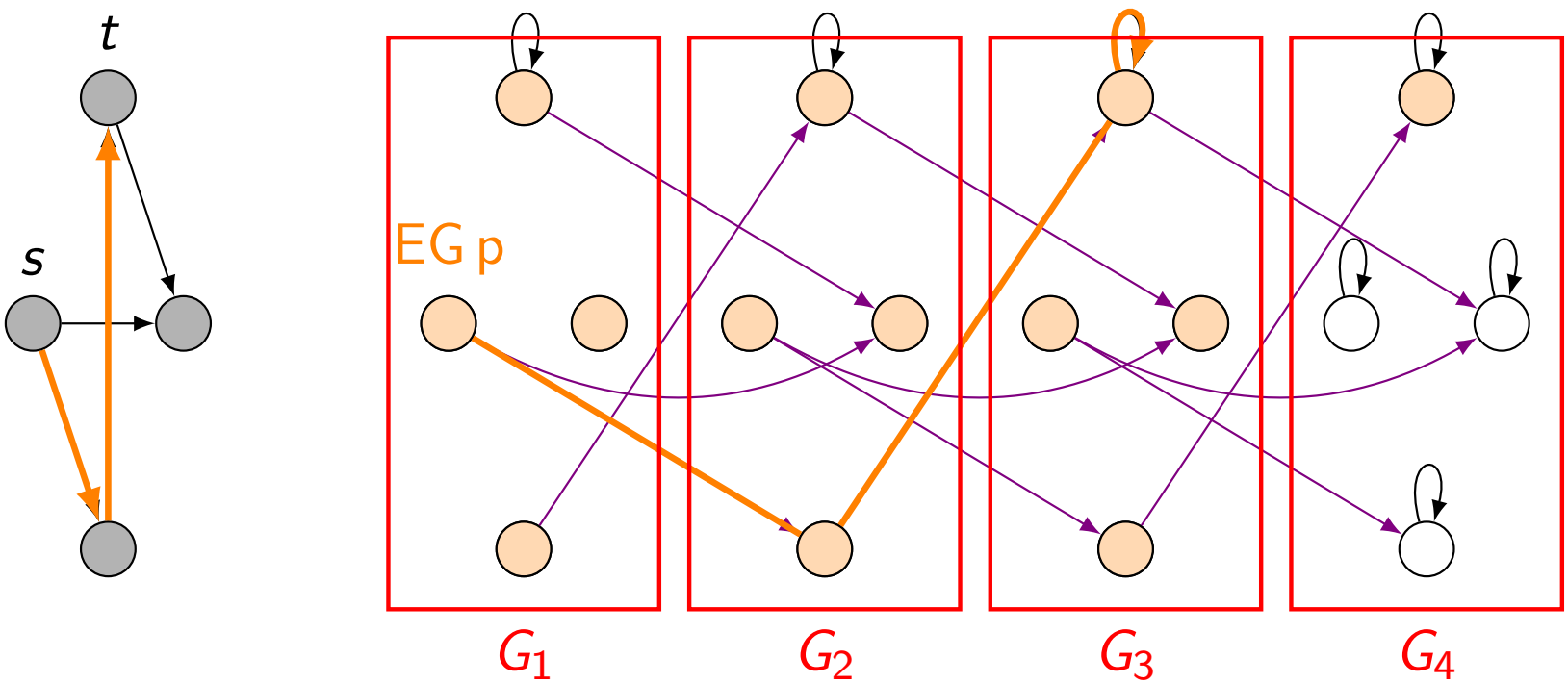
Warum alles so schwer ist

Definition

Problem	GAP
Instanzen	Gerichtete Graphen $G = (V, E)$, Knoten $s, t \in V$
Frage	Gibt es in G einen Pfad von s nach t?



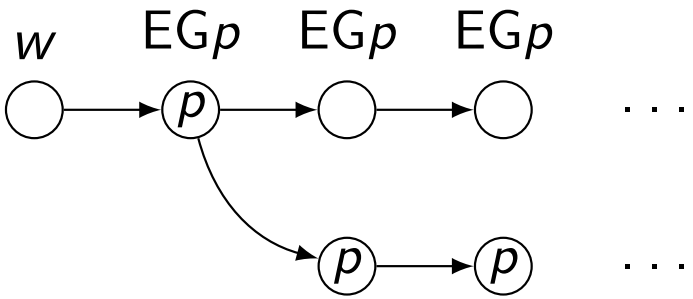
Der Pfad zur NL-Schwere



Satz

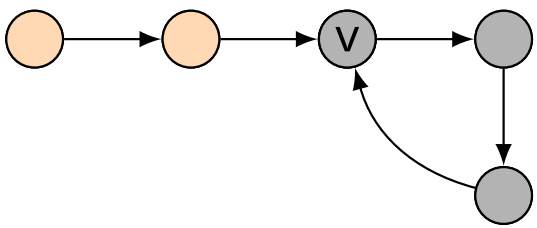
CTL-MC(EG) ist NL-schwer. $\langle G, s, t\rangle \mapsto \langle M, s_1, EG\ p\rangle$

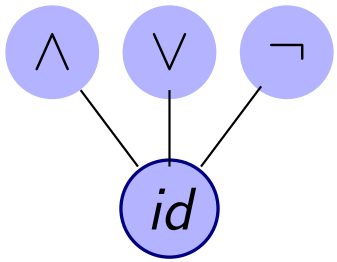
Wie hilft GAP bei CTL-MC(EG)?



$$EG\ EG\ p \equiv EG\ p$$

Wie wertet man nun EG p aus?



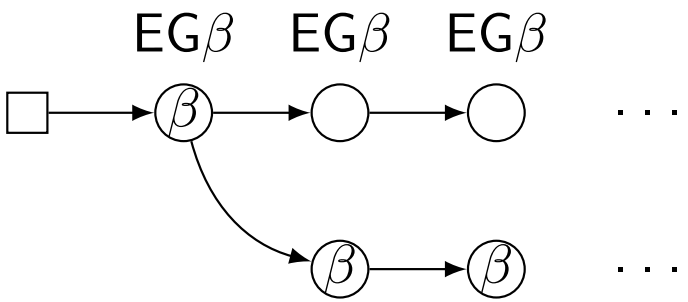


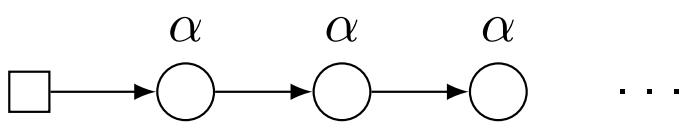
NL-schwer

in NL

Warum (EG,$\wedge$) genauso geht

Wie sieht ein Modell für $\mathsf{EG}(\alpha \wedge \mathsf{EG}\,\beta)$ aus?





$\models \mathsf{EG}(\alpha \wedge \beta)$

(EG, $\wedge$)-Formeln äquivalent zu $\alpha \wedge \bigwedge_{i=1}^{n} \mathsf{EG}\,\beta_i$

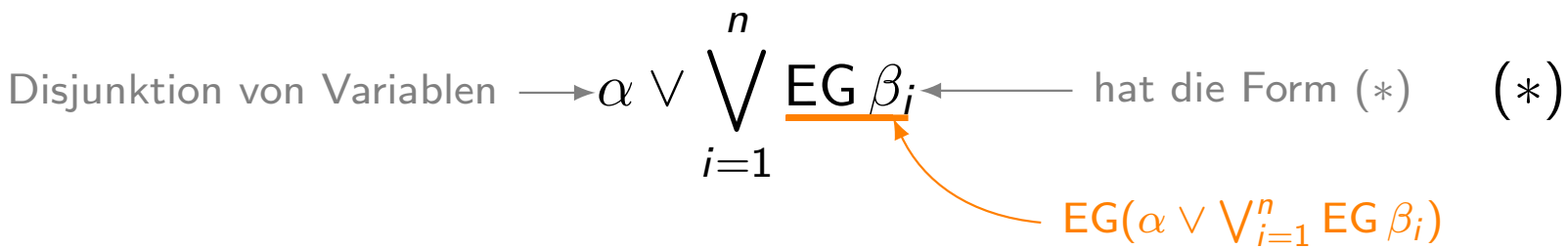
Konjunktionen von Variablen

NL-Algorithmus:

- α in gegebener Welt auswerten
- $\mathsf{EG}\,\beta_i$ wie im Fall ohne „und“ auswerten
- akzeptieren, falls jede Teilformel erfüllt ist

Vorüberlegungen für (EG,∨)-Formeln

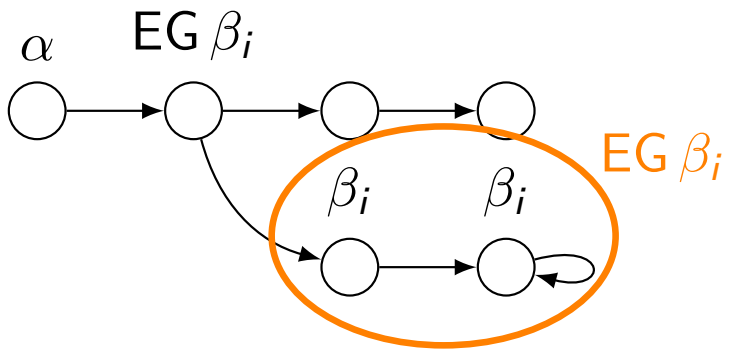
Jede (EG, ∨)-Formel ist äquivalent zu



$$(*)$$

Was brauchen wir zum Erfüllen dieser Formel?

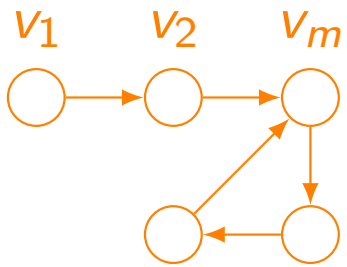
- Pfad, der Länge $\geq n+1$ hat und auf dem stets α gilt oder
- Pfad $v_1, \ldots, v_{m-1}$, auf dem stets α gilt sowie $M, u_i \models \mathit{EG}\,\beta_i$

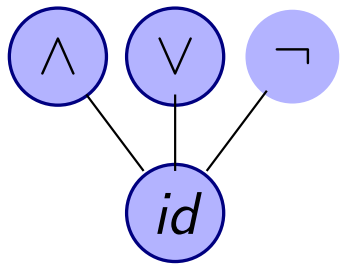


Ein NL-Algorithmus für $(\mathrm{EG}, \vee)$

Frage: $M, w \models \alpha \vee \bigvee_{i=1}^{n} EG\, \beta_i$?

1 **if** *Pfad für Fall 1 geraten* **then**
2 | **return** *true* — auf unendlichem Pfad gilt α
3 **else**
4 | Rate v_m und Pfad $v_1 (= w), \ldots, v_m$.
5 | **if** α *gilt in* $v_1, \ldots, v_{m-1}$ **then**
6 | | Rate i mit $M, v_m \models \beta_i$.
7 | | **return** $M, v_m \models \beta_i$
8 | **else**
9 | | **return** *false*
10 | **end**
11 **end**





NL-schwer

in NL

Nicht ist auch nicht schwerer

Satz

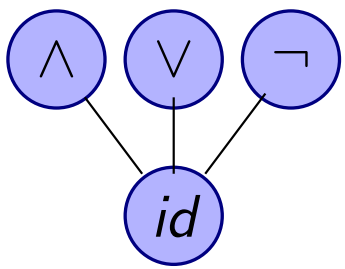
Jede (EG, $\neg$)-Formel ist äquivalent zu

1. $\mathsf{EG}\,\mathsf{AF}\,\varphi$ oder
2. $\mathsf{AF}\,\mathsf{EG}\,\varphi$ oder
3. $\mathsf{EG}\,\varphi$ oder
4. $\mathsf{AF}\,\varphi$

wobei $\varphi = p$ oder $\varphi = \neg p$

Durch scharfes Hinsehen

- $AF\,\varphi \equiv \neg EG\,\neg\varphi$
- CTL-MC(AF) $\in \mathrm{coNL} = \mathrm{NL}$



Mit CTL-MC(EG) und CTL-MC(AF):

- $\mathsf{EG}\,\mathsf{AF}\,\varphi$ via $\mathsf{EG}\,\psi$ prüfen
- $\mathsf{AF}\,\mathsf{EG}\,\varphi$ via $\mathsf{AF}\,\psi$ prüfen

Rückblick

Was haben wir heute gelernt?

- CTL-MC(EG) ist NL-vollständig
- Problem bleibt NL-vollständig mit Clones V, E, N

Literatur:

Krebs, Meier und Mundhenk.
„The model checking fingerprints of CTL operators“
In: Acta Informatica 56.6 (2019), S. 487-519