Logikbeschreibungen regulärer Sprachen

Manfred Kufleitner

Workshop „Automaten und Logik“, Ilmenau
25. September 2013

Reguläre Sprachen

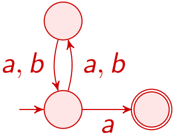
- Reguläre Ausdrücke $((a+b)^2)^*a$

Reguläre Sprachen

- Reguläre Ausdrücke

$$((a+b)^2)^* a$$

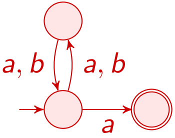
- Nichtdeterministische Automaten



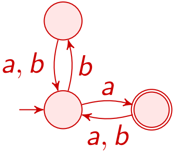
Reguläre Sprachen

- Reguläre Ausdrücke $((a+b)^2)^* a$

- Nichtdeterministische Automaten

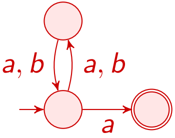


- Deterministische Automaten

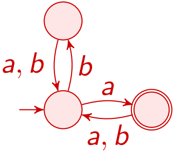


Reguläre Sprachen

- Reguläre Ausdrücke $((a+b)^2)^* a$
- Nichtdeterministische Automaten



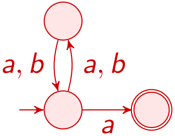
- Deterministische Automaten



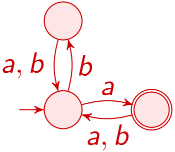
- Rechtslineare Grammatiken $S \to a \mid aT \mid bT, \;\; T \to aS \mid bS$

Reguläre Sprachen

- Reguläre Ausdrücke $((a+b)^2)^* a$
- Nichtdeterministische Automaten



- Deterministische Automaten



- Rechtslineare Grammatiken $S \to a \mid aT \mid bT,\ \ T \to aS \mid bS$
- Monadische Logik zweiter Stufe (MSO)
  $a(\max) \wedge \exists Y : \{\min, \max\} \subseteq Y \wedge \forall x (x \in Y \leftrightarrow x + 1 \notin Y)$

MSO-Logik

- MSO $=$ *monadic second-order*

MSO-Logik

- MSO $=$ *monadic second-order*
- Variablen $x, y, z, \ldots$

MSO-Logik

- MSO $=$ *monadic second-order*
- Variablen $x, y, z, \ldots$
- Mengenvariablen $X, Y, Z, \ldots$

MSO-Logik

- MSO $=$ *monadic second-order*
- Variablen $x, y, z, \ldots$
- Mengenvariablen $X, Y, Z, \ldots$
- Atomare Formeln $x = y$, $x < y$, $x \in X$, $a(x)$ für $a \in A$

MSO-Logik

- MSO = *monadic second-order*
- Variablen $x, y, z, \ldots$
- Mengenvariablen $X, Y, Z, \ldots$
- Atomare Formeln $x = y$, $x < y$, $x \in X$, $a(x)$ für $a \in A$
- Boolesche Verknüpfungen $\wedge$, $\vee$, $\neg$

MSO-Logik

- MSO = *monadic second-order*
- Variablen $x, y, z, \ldots$
- Mengenvariablen $X, Y, Z, \ldots$
- Atomare Formeln $x = y$, $x < y$, $x \in X$, $a(x)$ für $a \in A$
- Boolesche Verknüpfungen $\wedge$, $\vee$, $\neg$
- Quantoren $\exists x$, $\forall x$, $\exists X$, $\forall X$

MSO-Logik

- MSO = *monadic second-order*
- Variablen $x, y, z, \ldots$
- Mengenvariablen $X, Y, Z, \ldots$
- Atomare Formeln $x = y$, $x < y$, $x \in X$, $a(x)$ für $a \in A$
- Boolesche Verknüpfungen $\wedge$, $\vee$, $\neg$
- Quantoren $\exists x$, $\forall x$, $\exists X$, $\forall X$
- Makros min, max, $x = y + 1$, $X \subseteq Y$

MSO-Logik

- MSO = *monadic second-order*
- Variablen $x, y, z, \ldots$
- Mengenvariablen $X, Y, Z, \ldots$
- Atomare Formeln $x = y$, $x < y$, $x \in X$, $a(x)$ für $a \in A$
- Boolesche Verknüpfungen $\wedge$, $\vee$, $\neg$
- Quantoren $\exists x$, $\forall x$, $\exists X$, $\forall X$
- Makros min, max, $x = y + 1$, $X \subseteq Y$
- Semantik: Auswertung an Positionen eines Wortes $w \models \varphi$

MSO-Logik

- MSO = *monadic second-order*
- Variablen $x, y, z, \ldots$
- Mengenvariablen $X, Y, Z, \ldots$
- Atomare Formeln $x = y$, $x < y$, $x \in X$, $a(x)$ für $a \in A$
- Boolesche Verknüpfungen $\wedge$, $\vee$, $\neg$
- Quantoren $\exists x$, $\forall x$, $\exists X$, $\forall X$
- Makros min, max, $x = y + 1$, $X \subseteq Y$
- Semantik: Auswertung an Positionen eines Wortes $w \models \varphi$
- Definierte Sprache $L(\varphi) = \{w \in A^* \mid w \models \varphi\}$

MSO-Logik

- MSO = *monadic second-order*
- Variablen $x, y, z, \ldots$
- Mengenvariablen $X, Y, Z, \ldots$
- Atomare Formeln $x = y$, $x < y$, $x \in X$, $a(x)$ für $a \in A$
- Boolesche Verknüpfungen $\wedge$, $\vee$, $\neg$
- Quantoren $\exists x$, $\forall x$, $\exists X$, $\forall X$
- Makros $\min$, $\max$, $x = y + 1$, $X \subseteq Y$
- Semantik: Auswertung an Positionen eines Wortes $w \models \varphi$
- Definierte Sprache $L(\varphi) = \{w \in A^* \mid w \models \varphi\}$

- Beispiel:
  $\varphi = a(\max) \wedge \exists Y\colon \{\min, \max\} \subseteq Y \wedge \forall x (x \in Y \leftrightarrow x + 1 \notin Y)$

MSO-Logik

- MSO $=$ *monadic second-order*
- Variablen $x, y, z, \ldots$
- Mengenvariablen $X, Y, Z, \ldots$
- Atomare Formeln $x = y$, $x < y$, $x \in X$, $a(x)$ für $a \in A$
- Boolesche Verknüpfungen $\wedge$, $\vee$, $\neg$
- Quantoren $\exists x$, $\forall x$, $\exists X$, $\forall X$
- Makros min, max, $x = y + 1$, $X \subseteq Y$
- Semantik: Auswertung an Positionen eines Wortes $w \models \varphi$
- Definierte Sprache $L(\varphi) = \{w \in A^* \mid w \models \varphi\}$

- Beispiel:
  $\varphi = a(\mathrm{max}) \wedge \exists Y\colon \{\mathrm{min}, \mathrm{max}\} \subseteq Y \wedge \forall x (x \in Y \leftrightarrow x + 1 \notin Y)$
  - $baaabba \models \varphi$

MSO-Logik

- MSO = *monadic second-order*
- Variablen $x, y, z, \dots$
- Mengenvariablen $X, Y, Z, \dots$
- Atomare Formeln $x = y$, $x < y$, $x \in X$, $a(x)$ für $a \in A$
- Boolesche Verknüpfungen $\wedge$, $\vee$, $\neg$
- Quantoren $\exists x$, $\forall x$, $\exists X$, $\forall X$
- Makros $\min$, $\max$, $x = y + 1$, $X \subseteq Y$
- Semantik: Auswertung an Positionen eines Wortes $w \models \varphi$
- Definierte Sprache $L(\varphi) = \{w \in A^* \mid w \models \varphi\}$

- Beispiel:
  $\varphi = a(\max) \wedge \exists Y : \{\min, \max\} \subseteq Y \wedge \forall x (x \in Y \leftrightarrow x + 1 \notin Y)$
  - $baaabba \models \varphi$
  - $baaabbba \not\models \varphi$

Formale Verifikation

- Sequentielle Systeme werden (z.B.) durch Automaten $\mathcal{A}$ implementiert

Formale Verifikation

- Sequentielle Systeme werden (z.B.) durch Automaten $\mathcal{A}$ implementiert
- Anforderungen werden (z.B.) durch MSO-Formel φ spezifiziert

Formale Verifikation

- Sequentielle Systeme werden (z.B.) durch Automaten $\mathcal{A}$ implementiert
- Anforderungen werden (z.B.) durch MSO-Formel φ spezifiziert
- $\mathcal{A}$ erfüllt φ, wenn $L(\mathcal{A}) \subseteq L(\varphi)$ gilt. *(Inklusionsproblem)*

Formale Verifikation

- Sequentielle Systeme werden (z.B.) durch Automaten $\mathcal{A}$ implementiert
- Anforderungen werden (z.B.) durch MSO-Formel φ spezifiziert
- $\mathcal{A}$ erfüllt φ, wenn $L(\mathcal{A}) \subseteq L(\varphi)$ gilt. *(Inklusionsproblem)*
- Spezifikation widersprüchlich, wenn $L(\varphi) = \emptyset$ gilt. *(Leerheitsproblem)*

Formale Verifikation

- Sequentielle Systeme werden (z.B.) durch Automaten $\mathcal{A}$ implementiert
- Anforderungen werden (z.B.) durch MSO-Formel φ spezifiziert
- $\mathcal{A}$ erfüllt φ, wenn $L(\mathcal{A}) \subseteq L(\varphi)$ gilt. *(Inklusionsproblem)*
- Spezifikation widersprüchlich, wenn $L(\varphi) = \emptyset$ gilt. *(Leerheitsproblem)*
- Ein Ablauf $w \in L(\mathcal{A})$ erfüllt φ, wenn $w \in L(\varphi)$ gilt. *(Wortproblem)*

Formale Verifikation

- Sequentielle Systeme werden (z.B.) durch Automaten $\mathcal{A}$ implementiert
- Anforderungen werden (z.B.) durch MSO-Formel φ spezifiziert
- $\mathcal{A}$ erfüllt φ, wenn $L(\mathcal{A}) \subseteq L(\varphi)$ gilt. *(Inklusionsproblem)*
- Spezifikation widersprüchlich, wenn $L(\varphi) = \emptyset$ gilt. *(Leerheitsproblem)*
- Ein Ablauf $w \in L(\mathcal{A})$ erfüllt φ, wenn $w \in L(\varphi)$ gilt. *(Wortproblem)*

- **Problem:** Berechnungsprobleme wie *Inklusionsproblem* und *Leerheitsproblem* sind für MSO-Logik nicht elementar.

Formale Verifikation

- Sequentielle Systeme werden (z.B.) durch Automaten $\mathcal{A}$ implementiert
- Anforderungen werden (z.B.) durch MSO-Formel φ spezifiziert
- $\mathcal{A}$ erfüllt φ, wenn $L(\mathcal{A}) \subseteq L(\varphi)$ gilt. *(Inklusionsproblem)*
- Spezifikation widersprüchlich, wenn $L(\varphi) = \emptyset$ gilt. *(Leerheitsproblem)*
- Ein Ablauf $w \in L(\mathcal{A})$ erfüllt φ, wenn $w \in L(\varphi)$ gilt. *(Wortproblem)*
- **Problem:** Berechnungsprobleme wie *Inklusionsproblem* und *Leerheitsproblem* sind für MSO-Logik nicht elementar.
- Andererseits ist für viele Eigenschaften nicht die volle Ausdrucksstärke von MSO erforderlich.

Formale Verifikation

- Sequentielle Systeme werden (z.B.) durch Automaten $\mathcal{A}$ implementiert
- Anforderungen werden (z.B.) durch MSO-Formel φ spezifiziert
- $\mathcal{A}$ erfüllt φ, wenn $L(\mathcal{A}) \subseteq L(\varphi)$ gilt. *(Inklusionsproblem)*
- Spezifikation widersprüchlich, wenn $L(\varphi) = \emptyset$ gilt. *(Leerheitsproblem)*
- Ein Ablauf $w \in L(\mathcal{A})$ erfüllt φ, wenn $w \in L(\varphi)$ gilt. *(Wortproblem)*

- **Problem:** Berechnungsprobleme wie *Inklusionsproblem* und *Leerheitsproblem* sind für MSO-Logik nicht elementar.

- Andererseits ist für viele Eigenschaften nicht die volle Ausdrucksstärke von MSO erforderlich. $\Rightarrow$ Logikfragmente

Formale Verifikation

- Sequentielle Systeme werden (z.B.) durch Automaten $\mathcal{A}$ implementiert
- Anforderungen werden (z.B.) durch MSO-Formel φ spezifiziert
- $\mathcal{A}$ erfüllt φ, wenn $L(\mathcal{A}) \subseteq L(\varphi)$ gilt. *(Inklusionsproblem)*
- Spezifikation widersprüchlich, wenn $L(\varphi) = \emptyset$ gilt. *(Leerheitsproblem)*
- Ein Ablauf $w \in L(\mathcal{A})$ erfüllt φ, wenn $w \in L(\varphi)$ gilt. *(Wortproblem)*
- **Problem:** Berechnungsprobleme wie *Inklusionsproblem* und *Leerheitsproblem* sind für MSO-Logik nicht elementar.
- Andererseits ist für viele Eigenschaften nicht die volle Ausdrucksstärke von MSO erforderlich. $\Rightarrow$ Logikfragmente
- **Lösungsansatz:** Eingeschränktere Logikfragmente ermöglichen häufig effizientere Algorithmen.

FO-Logik (1/3)

- FO $=$ *first-order*, keine Mengenvariablen

FO-Logik (1/3)

- FO = *first-order*, keine Mengenvariablen
- Beispiel:
  - $\varphi = a(\min) \wedge b(\max) \wedge \forall x \big(a(x) \leftrightarrow b(x+1)\big)$

FO-Logik (1/3)

- FO = *first-order*, keine Mengenvariablen
- Beispiel:
  - $\varphi = a(\min) \wedge b(\max) \wedge \forall x \big(a(x) \leftrightarrow b(x+1) \big)$
  - $L(\varphi) = (ab)^+$

FO-Logik (1/3)

- FO = *first-order*, keine Mengenvariablen
- Beispiel:
  - $\varphi = a(\min) \land b(\max) \land \forall x \big(a(x) \leftrightarrow b(x+1)\big)$
  - $L(\varphi) = (ab)^+$
- Sternfreier Ausdruck: wie regulärer Ausdruck, aber mit Komplement anstelle von Stern

FO-Logik (1/3)

- FO = *first-order*, keine Mengenvariablen
- Beispiel:
  - $\varphi = a(\min) \wedge b(\max) \wedge \forall x \big(a(x) \leftrightarrow b(x+1)\big)$
  - $L(\varphi) = (ab)^+$
- Sternfreier Ausdruck: wie regulärer Ausdruck, aber mit Komplement anstelle von Stern
- Sternfreie Sprachen: Abschluss der endlichen Sprachen unter Konkatenation und booleschen Operationen

FO-Logik (1/3)

- FO = *first-order*, keine Mengenvariablen
- Beispiel:
  - $\varphi = a(\min) \wedge b(\max) \wedge \forall x \big(a(x) \leftrightarrow b(x+1)\big)$
  - $L(\varphi) = (ab)^+$
- Sternfreier Ausdruck: wie regulärer Ausdruck, aber mit Komplement anstelle von Stern
- Sternfreie Sprachen: Abschluss der endlichen Sprachen unter Konkatenation und booleschen Operationen
- Beispiel: $A = \{a, b\}$, $A^* = \overline{\emptyset}$

FO-Logik (1/3)

- FO = *first-order*, keine Mengenvariablen
- Beispiel:
  - $\varphi = a(\min) \wedge b(\max) \wedge \forall x \big(a(x) \leftrightarrow b(x+1)\big)$
  - $L(\varphi) = (ab)^+$
- Sternfreier Ausdruck: wie regulärer Ausdruck, aber mit Komplement anstelle von Stern
- Sternfreie Sprachen: Abschluss der endlichen Sprachen unter Konkatenation und booleschen Operationen
- Beispiel: $A = \{a, b\}$, $A^* = \overline{\emptyset}$
  - $(ab)^+ = aA^* \cap A^*b \cap \overline{A^*aaA^*} \cap \overline{A^*bbA^*}$

FO-Logik (1/3)

- FO = *first-order*, keine Mengenvariablen
- Beispiel:
  - $\varphi = a(\min) \wedge b(\max) \wedge \forall x\big(a(x) \leftrightarrow b(x+1)\big)$
  - $L(\varphi) = (ab)^+$
- Sternfreier Ausdruck: wie regulärer Ausdruck, aber mit Komplement anstelle von Stern
- Sternfreie Sprachen: Abschluss der endlichen Sprachen unter Konkatenation und booleschen Operationen
- Beispiel: $A = \{a, b\}$, $A^* = \overline{\emptyset}$
  - $(ab)^+ = aA^* \cap A^*b \cap \overline{A^*aaA^*} \cap \overline{A^*bbA^*}$
- Syntaktisches Monoid: Transitionsmonoid des Minimalautomaten; effektiv berechenbar

FO-Logik (1/3)

- FO = *first-order*, keine Mengenvariablen
- Beispiel:
  - $\varphi = a(\min) \wedge b(\max) \wedge \forall x \big(a(x) \leftrightarrow b(x+1) \big)$
  - $L(\varphi) = (ab)^+$
- Sternfreier Ausdruck: wie regulärer Ausdruck, aber mit Komplement anstelle von Stern
- Sternfreie Sprachen: Abschluss der endlichen Sprachen unter Konkatenation und booleschen Operationen
- Beispiel: $A = \{a, b\}$, $A^* = \overline{\emptyset}$
  - $(ab)^+ = aA^* \cap A^*b \cap \overline{A^*aaA^*} \cap \overline{A^*bbA^*}$
- Syntaktisches Monoid: Transitionsmonoid des Minimalautomaten; effektiv berechenbar
- aperiodisch: keine Unterhalbgruppe bildet eine Gruppe

FO-Logik (1/3)

- FO = *first-order*, keine Mengenvariablen
- Beispiel:
  - $\varphi = a(\min) \wedge b(\max) \wedge \forall x \big(a(x) \leftrightarrow b(x+1)\big)$
  - $L(\varphi) = (ab)^+$
- Sternfreier Ausdruck: wie regulärer Ausdruck, aber mit Komplement anstelle von Stern
- Sternfreie Sprachen: Abschluss der endlichen Sprachen unter Konkatenation und booleschen Operationen
- Beispiel: $A = \{a, b\}$, $A^* = \overline{\emptyset}$
  - $(ab)^+ = aA^* \cap A^*b \cap \overline{A^*aaA^*} \cap \overline{A^*bbA^*}$
- Syntaktisches Monoid: Transitionsmonoid des Minimalautomaten; effektiv berechenbar
- aperiodisch: keine Unterhalbgruppe bildet eine Gruppe
- Beispiel: Syntaktisches Monoid von $(aa)^+$ ist $\mathbb{Z}/2\mathbb{Z}$.

FO-Logik (2/3)

Theorem: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- L ist sternfrei.

FO-Logik (2/3)

Theorem: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- L ist sternfrei.
- L ist definierbar in FO-Logik. [McNaughton, Papert 1971]

FO-Logik (2/3)

Theorem: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- L ist sternfrei.
- L ist definierbar in FO-Logik. [McNaughton, Papert 1971]
- L ist definierbar in linearer Temporallogik. [Kamp 1968]

FO-Logik (2/3)

Theorem: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- L ist sternfrei.
- L ist definierbar in FO-Logik. [McNaughton, Papert 1971]
- L ist definierbar in linearer Temporallogik. [Kamp 1968]
- Das syntaktische Monoid von L ist aperiodisch. [Schützenberger 1965]

FO-Logik (2/3)

Theorem: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- L ist sternfrei.
- L ist definierbar in FO-Logik. [McNaughton, Papert 1971]
- L ist definierbar in linearer Temporallogik. [Kamp 1968]
- Das syntaktische Monoid von L ist aperiodisch. [Schützenberger 1965]

Korollar: Es ist entscheidbar, ob eine reguläre Sprache FO-definierbar ist.

FO-Logik (2/3)

Theorem: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- L ist sternfrei.
- L ist definierbar in FO-Logik. [McNaughton, Papert 1971]
- L ist definierbar in linearer Temporallogik. [Kamp 1968]
- Das syntaktische Monoid von L ist aperiodisch. [Schützenberger 1965]

Korollar: Es ist entscheidbar, ob eine reguläre Sprache FO-definierbar ist.

Beispiel:

- $(ab)^+$ ist FO-definierbar, $(aa)^+$ nicht.

FO-Logik (2/3)

Theorem: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- L ist sternfrei.
- L ist definierbar in FO-Logik. [McNaughton, Papert 1971]
- L ist definierbar in linearer Temporallogik. [Kamp 1968]
- Das syntaktische Monoid von L ist aperiodisch. [Schützenberger 1965]

Korollar: Es ist entscheidbar, ob eine reguläre Sprache FO-definierbar ist.

Beispiel:

- $(ab)^+$ ist FO-definierbar, $(aa)^+$ nicht.
- $((a+b)^2)^*a$ ist auch nicht FO-definierbar.

FO-Logik (3/3)

- Typische Berechnungsprobleme für lineare Temporallogik (LTL) sind in PSPACE.

FO-Logik (3/3)

- Typische Berechnungsprobleme für lineare Temporallogik (LTL) sind in PSPACE.
- Für FO-Logik sind die meisten Probleme aber weiterhin nicht elementar (Wortproblem ist PSPACE-vollständig).

FO-Logik (3/3)

- Typische Berechnungsprobleme für lineare Temporallogik (LTL) sind in PSPACE.
- Für FO-Logik sind die meisten Probleme aber weiterhin nicht elementar (Wortproblem ist PSPACE-vollständig).
- FO^k = FO-Logik mit nur k verschiedenen Variablennamen

FO-Logik (3/3)

- Typische Berechnungsprobleme für lineare Temporallogik (LTL) sind in PSPACE.
- Für FO-Logik sind die meisten Probleme aber weiterhin nicht elementar (Wortproblem ist PSPACE-vollständig).
- FO^k = FO-Logik mit nur k verschiedenen Variablennamen
- $\mathsf{FO} \subseteq \mathsf{LTL} \subseteq \mathsf{FO}^3 \subseteq \mathsf{FO}$, d.h. $\mathsf{FO} = \mathsf{FO}^3$

FO-Logik (3/3)

- Typische Berechnungsprobleme für lineare Temporallogik (LTL) sind in PSPACE.
- Für FO-Logik sind die meisten Probleme aber weiterhin nicht elementar (Wortproblem ist PSPACE-vollständig).
- FO^k = FO-Logik mit nur k verschiedenen Variablennamen
- $\mathrm{FO} \subseteq \mathrm{LTL} \subseteq \mathrm{FO}^3 \subseteq \mathrm{FO}$, d.h. $\mathrm{FO} = \mathrm{FO}^3$
- $\mathrm{FO}^2 \subsetneq \mathrm{FO}$, z.B. ist $(ab)^*$ nicht FO^2-definierbar.

FO-Logik (3/3)

- Typische Berechnungsprobleme für lineare Temporallogik (LTL) sind in PSPACE.
- Für FO-Logik sind die meisten Probleme aber weiterhin nicht elementar (Wortproblem ist PSPACE-vollständig).
- FO^k = FO-Logik mit nur k verschiedenen Variablennamen
- $\mathrm{FO} \subseteq \mathrm{LTL} \subseteq \mathrm{FO}^3 \subseteq \mathrm{FO}$, d.h. $\mathrm{FO} = \mathrm{FO}^3$
- $\mathrm{FO}^2 \subsetneq \mathrm{FO}$, z.B. ist $(ab)^*$ nicht FO^2-definierbar.
- Leerheitsproblem für FO^2 ist coNP-vollständig. [Weis 2011]

FO-Logik (3/3)

- Typische Berechnungsprobleme für lineare Temporallogik (LTL) sind in PSPACE.
- Für FO-Logik sind die meisten Probleme aber weiterhin nicht elementar (Wortproblem ist PSPACE-vollständig).
- FO^k = FO-Logik mit nur k verschiedenen Variablennamen
- $\mathrm{FO} \subseteq \mathrm{LTL} \subseteq \mathrm{FO}^3 \subseteq \mathrm{FO}$, d.h. $\mathrm{FO} = \mathrm{FO}^3$
- $\mathrm{FO}^2 \subsetneq \mathrm{FO}$, z.B. ist $(ab)^*$ nicht FO^2-definierbar.
- Leerheitsproblem für FO^2 ist coNP-vollständig. [Weis 2011]

- Beispiel:
  - $\varphi = \exists x\colon a_1(x) \wedge \exists y > x\colon \Big(a_2(y) \wedge \exists x > y\colon \big(a_3(x) \wedge \cdots\big)\Big)$

FO-Logik (3/3)

- Typische Berechnungsprobleme für lineare Temporallogik (LTL) sind in PSPACE.
- Für FO-Logik sind die meisten Probleme aber weiterhin nicht elementar (Wortproblem ist PSPACE-vollständig).
- FO^k = FO-Logik mit nur k verschiedenen Variablennamen
- $\mathrm{FO} \subseteq \mathrm{LTL} \subseteq \mathrm{FO}^3 \subseteq \mathrm{FO}$, d.h. $\mathrm{FO} = \mathrm{FO}^3$
- $\mathrm{FO}^2 \subsetneq \mathrm{FO}$, z.B. ist $(ab)^*$ nicht FO^2-definierbar.
- Leerheitsproblem für FO^2 ist coNP-vollständig. [Weis 2011]

- Beispiel:
  - $\varphi = \exists x\colon a_1(x) \wedge \exists y > x\colon \Big(a_2(y) \wedge \exists x > y\colon \big(a_3(x) \wedge \cdots\big)\Big)$
  - $L(\varphi) = A^* a_1 A^* a_2 \cdots A^* a_k A^*$

FO-Logik (3/3)

- Typische Berechnungsprobleme für lineare Temporallogik (LTL) sind in PSPACE.
- Für FO-Logik sind die meisten Probleme aber weiterhin nicht elementar (Wortproblem ist PSPACE-vollständig).
- FO^k = FO-Logik mit nur k verschiedenen Variablennamen
- $\mathrm{FO} \subseteq \mathrm{LTL} \subseteq \mathrm{FO}^3 \subseteq \mathrm{FO}$, d.h. $\mathrm{FO} = \mathrm{FO}^3$
- $\mathrm{FO}^2 \subsetneq \mathrm{FO}$, z.B. ist $(ab)^*$ nicht FO^2-definierbar.
- Leerheitsproblem für FO^2 ist coNP-vollständig. [Weis 2011]

- Beispiel:
  - $\varphi = \exists x\colon a_1(x) \wedge \exists y > x\colon \Big(a_2(y) \wedge \exists x > y\colon \big(a_3(x) \wedge \cdots\big)\Big)$
  - $L(\varphi) = A^* a_1 A^* a_2 \cdots A^* a_k A^*$

- „Verstehen“ wir nun FO^2?

Fragmente (1/3)

Was gehört zum Verständnis eines Fragments $\mathcal{F}$? (z.B. $\mathcal{F} = \mathsf{FO}$)

Fragmente (1/3)

Was gehört zum Verständnis eines Fragments $\mathcal{F}$? (z.B. $\mathcal{F} = \mathrm{FO}$)

- Komplexität von Berechnungsproblemen für $\mathcal{F}$

Fragmente (1/3)

Was gehört zum Verständnis eines Fragments $\mathcal{F}$? (z.B. $\mathcal{F} = \mathrm{FO}$)

- Komplexität von Berechnungsproblemen für $\mathcal{F}$
- Welche Sprachen lassen sich mit $\mathcal{F}$ definieren? ($\mathrm{FO} =$ sternfrei)

Fragmente (1/3)

Was gehört zum Verständnis eines Fragments $\mathcal{F}$? (z.B. $\mathcal{F} = \mathrm{FO}$)

- Komplexität von Berechnungsproblemen für $\mathcal{F}$
- Welche Sprachen lassen sich mit $\mathcal{F}$ definieren? (FO = sternfrei)
- Wie kann man entscheiden, ob eine gegebene reguläre Sprache L in $\mathcal{F}$ definierbar ist? (FO = aperiodisch)

Fragmente (1/3)

Was gehört zum Verständnis eines Fragments $\mathcal{F}$? (z.B. $\mathcal{F} = \mathrm{FO}$)

- Komplexität von Berechnungsproblemen für $\mathcal{F}$
- Welche Sprachen lassen sich mit $\mathcal{F}$ definieren? (FO = sternfrei)
- Wie kann man entscheiden, ob eine gegebene reguläre Sprache L in $\mathcal{F}$ definierbar ist? (FO = aperiodisch)
- Welche Abschlusseigenschaften haben die $\mathcal{F}$-definierbaren Sprachen? (FO abgeschlossen unter inversen Homomorphismen)

Fragmente (1/3)

Was gehört zum Verständnis eines Fragments $\mathcal{F}$? (z.B. $\mathcal{F} = \mathrm{FO}$)

- Komplexität von Berechnungsproblemen für $\mathcal{F}$
- Welche Sprachen lassen sich mit $\mathcal{F}$ definieren? (FO = sternfrei)
- Wie kann man entscheiden, ob eine gegebene reguläre Sprache L in $\mathcal{F}$ definierbar ist? (FO = aperiodisch)
- Welche Abschlusseigenschaften haben die $\mathcal{F}$-definierbaren Sprachen? (FO abgeschlossen unter inversen Homomorphismen)
- Welches andere Fragment beschreibt ebenfalls die $\mathcal{F}$-definierbaren Sprachen? ($\mathrm{FO} = \mathrm{FO}^3$)

Fragmente (1/3)

Was gehört zum Verständnis eines Fragments $\mathcal{F}$? (z.B. $\mathcal{F} = \mathrm{FO}$)

- Komplexität von Berechnungsproblemen für $\mathcal{F}$
- Welche Sprachen lassen sich mit $\mathcal{F}$ definieren? (FO = sternfrei)
- Wie kann man entscheiden, ob eine gegebene reguläre Sprache L in $\mathcal{F}$ definierbar ist? (FO = aperiodisch)
- Welche Abschlusseigenschaften haben die $\mathcal{F}$-definierbaren Sprachen? (FO abgeschlossen unter inversen Homomorphismen)
- Welches andere Fragment beschreibt ebenfalls die $\mathcal{F}$-definierbaren Sprachen? ($\mathrm{FO} = \mathrm{FO}^3$)
- Ist Separierbarkeit durch $\mathcal{F}$-definierbare Sprachen entscheidbar? Berechnung von Separatoren

Fragmente (1/3)

Was gehört zum Verständnis eines Fragments $\mathcal{F}$? (z.B. $\mathcal{F} = \mathrm{FO}$)

- Komplexität von Berechnungsproblemen für $\mathcal{F}$
- Welche Sprachen lassen sich mit $\mathcal{F}$ definieren? (FO = sternfrei)
- Wie kann man entscheiden, ob eine gegebene reguläre Sprache L in $\mathcal{F}$ definierbar ist? (FO = aperiodisch)
- Welche Abschlusseigenschaften haben die $\mathcal{F}$-definierbaren Sprachen? (FO abgeschlossen unter inversen Homomorphismen)
- Welches andere Fragment beschreibt ebenfalls die $\mathcal{F}$-definierbaren Sprachen? ($\mathrm{FO} = \mathrm{FO}^3$)
- Ist Separierbarkeit durch $\mathcal{F}$-definierbare Sprachen entscheidbar? Berechnung von Separatoren

Der Ansatz liefert auch eine deskriptive Komplexitätstheorie innerhalb der regulären Sprachen.

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen
- Typische Ressourcen:
  - erlaubte Modalitäten / Prädikate

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen
- Typische Ressourcen:
  - erlaubte Modalitäten / Prädikate
  - Quantorentiefe

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen
- Typische Ressourcen:
  - erlaubte Modalitäten / Prädikate
  - Quantorentiefe
  - Alternierungstiefe

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen
- Typische Ressourcen:
  - erlaubte Modalitäten / Prädikate
  - Quantorentiefe
  - Alternierungstiefe
  - Anzahl der Variablen

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen
- Typische Ressourcen:
  - erlaubte Modalitäten / Prädikate
  - Quantorentiefe
  - Alternierungstiefe
  - Anzahl der Variablen
- Beispiel:
  - $\varphi \ = \ \exists x \big(\, a(x) \, \wedge$

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen
- Typische Ressourcen:
  - erlaubte Modalitäten / Prädikate
  - Quantorentiefe
  - Alternierungstiefe
  - Anzahl der Variablen
- Beispiel:
  - $\varphi = \exists x\big(a(x) \wedge$
    $\exists y(b(y) \wedge x < y \wedge$

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen
- Typische Ressourcen:
  - erlaubte Modalitäten / Prädikate
  - Quantorentiefe
  - Alternierungstiefe
  - Anzahl der Variablen
- Beispiel:
  - $\varphi = \exists x\big(a(x) \wedge$
    $\exists y(b(y) \wedge x < y \wedge \forall x(x \leq y \vee c(x))) \wedge$

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen
- Typische Ressourcen:
  - erlaubte Modalitäten / Prädikate
  - Quantorentiefe
  - Alternierungstiefe
  - Anzahl der Variablen
- Beispiel:
  - $$\begin{aligned}\varphi \;=\; \exists x\big(\, & a(x) \wedge \\ & \exists y(b(y) \wedge x < y \wedge \forall x(x \leq y \vee c(x))) \wedge \\ & \forall y(y \geq x)\,\big)\end{aligned}$$

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen
- Typische Ressourcen:
  - erlaubte Modalitäten / Prädikate
  - Quantorentiefe
  - Alternierungstiefe
  - Anzahl der Variablen
- Beispiel:
  - $\varphi = \exists x\big(a(x) \wedge$
    $\exists y(b(y) \wedge x < y \wedge \forall x(x \leq y \vee c(x))) \wedge$
    $\forall y(y \geq x)\big)$
  - φ hat Quantorentiefe 3

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen
- Typische Ressourcen:
  - erlaubte Modalitäten / Prädikate
  - Quantorentiefe
  - Alternierungstiefe
  - Anzahl der Variablen
- Beispiel:
  - $\varphi = \exists x\big(a(x) \wedge$
    $\exists y(b(y) \wedge x < y \wedge \forall x(x \leq y \vee c(x))) \wedge$
    $\forall y(y \geq x)\big)$
  - φ hat Quantorentiefe 3
  - φ hat Alternierungstiefe 2

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen
- Typische Ressourcen:
  - erlaubte Modalitäten / Prädikate
  - Quantorentiefe
  - Alternierungstiefe
  - Anzahl der Variablen
- Beispiel:
  - $\varphi = \exists x\big(a(x) \wedge$
    $\exists y(b(y) \wedge x < y \wedge \forall x(x \leq y \vee c(x))) \wedge$
    $\forall y(y \geq x)\big)$
  - φ hat Quantorentiefe 3
  - φ hat Alternierungstiefe 2
  - φ verwendet 2 Variablen

Fragmente (2/3)

- Fragmente entstehen oft durch Beschränkung von Ressourcen
- Typische Ressourcen:
  - erlaubte Modalitäten / Prädikate
  - Quantorentiefe
  - Alternierungstiefe
  - Anzahl der Variablen
- Beispiel:
  - $$\begin{aligned}\varphi \;=\; \exists x\big(\, & a(x) \wedge \\ & \exists y(b(y) \wedge x < y \wedge \forall x(x \leq y \vee c(x))) \wedge \\ & \forall y(y \geq x)\,\big)\end{aligned}$$
  - φ hat Quantorentiefe 3
  - φ hat Alternierungstiefe 2
  - φ verwendet 2 Variablen
  - $L(\varphi) = a\{a, b, c\}^* bc^*$

Fragmente (3/3)

- $y = x + 1 \;\equiv\; x < y \wedge \forall z (z \leq x \vee z \geq y)$

Fragmente (3/3)

- $y = x + 1 \;\equiv\; x < y \wedge \forall z (z \leq x \vee z \geq y)$
- Makro erfordert Quantor (Tiefe/Alternierung) und neue Variable z

Fragmente (3/3)

- $y = x + 1 \;\equiv\; x < y \wedge \forall z(z \leq x \vee z \geq y)$
- Makro erfordert Quantor (Tiefe/Alternierung) und neue Variable z
- $\mathrm{FO}[<] = \mathrm{FO}[<, +1]$

Fragmente (3/3)

- $y = x + 1 \;\equiv\; x < y \wedge \forall z (z \leq x \vee z \geq y)$
- Makro erfordert Quantor (Tiefe/Alternierung) und neue Variable z
- $\mathrm{FO}[<] = \mathrm{FO}[<, +1]$
- $\mathrm{FO}^3[<] = \mathrm{FO}^3[<, +1]$

Fragmente (3/3)

- $y = x + 1 \;\equiv\; x < y \wedge \forall z (z \leq x \vee z \geq y)$
- Makro erfordert Quantor (Tiefe/Alternierung) und neue Variable z
- $\mathrm{FO}[<] = \mathrm{FO}[<, +1]$
- $\mathrm{FO}^3[<] = \mathrm{FO}^3[<, +1]$
- $\mathrm{FO}^2[<] \subsetneq \mathrm{FO}^2[<, +1]$

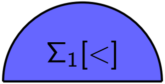
Fragmente (3/3)

- $y = x + 1 \;\equiv\; x < y \wedge \forall z(z \leq x \vee z \geq y)$
- Makro erfordert Quantor (Tiefe/Alternierung) und neue Variable z
- $\mathrm{FO}[<] = \mathrm{FO}[<, +1]$
- $\mathrm{FO}^3[<] = \mathrm{FO}^3[<, +1]$
- $\mathrm{FO}^2[<] \subsetneq \mathrm{FO}^2[<, +1]$
- Σ_m = Alternierungstiefe m, erster Block ist $\exists$

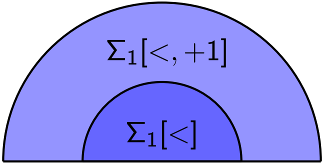
Fragmente (3/3)

- $y = x + 1 \ \equiv\ x < y \wedge \forall z (z \leq x \vee z \geq y)$
- Makro erfordert Quantor (Tiefe/Alternierung) und neue Variable z
- $\mathsf{FO}[<] = \mathsf{FO}[<, +1]$
- $\mathsf{FO}^3[<] = \mathsf{FO}^3[<, +1]$
- $\mathsf{FO}^2[<] \subsetneq \mathsf{FO}^2[<, +1]$
- Σ_m = Alternierungstiefe m, erster Block ist $\exists$
- $\Sigma_m[<] \subsetneq \Sigma_m[<, +1] \subsetneq \Sigma_{m+1}[<]$

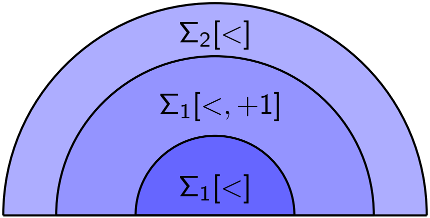
Quantorenalternierung



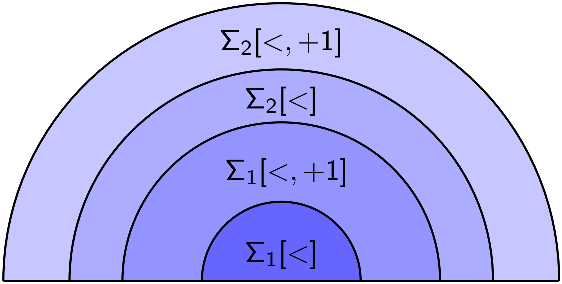
Quantorenalternierung



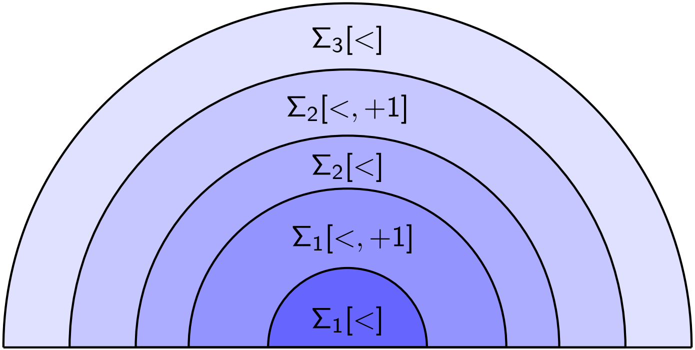
Quantorenalternierung



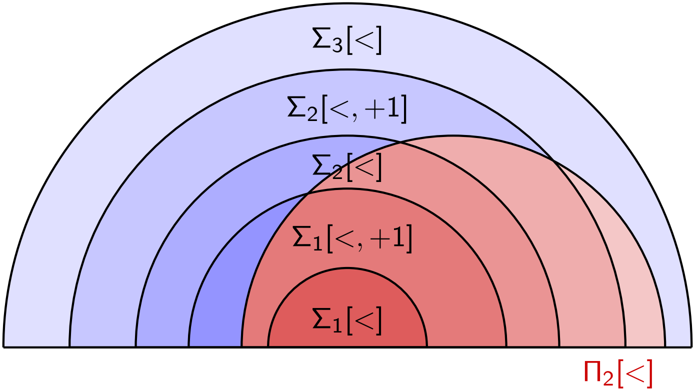
Quantorenalternierung



Quantorenalternierung



Quantorenalternierung



Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$.

[Schützenberger 1976]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$.
  [Schützenberger 1976]
- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

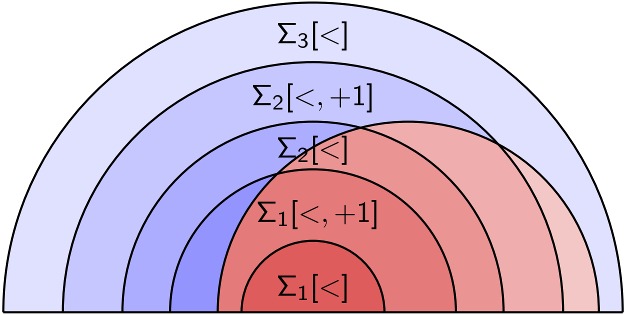
- Das syntaktische Monoid von L ist in **DA**.
- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$.
  [Schützenberger 1976]
- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$.
  [Etessami, Vardi, Wilke 1997]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- L ist Vereinigung eindeutiger Monome über $\{\, B^* \mid B \subseteq A \,\}$. [Schützenberger 1976]
- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$. [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathrm{FO}^2[<]$. [Thérien, Wilke 1998]

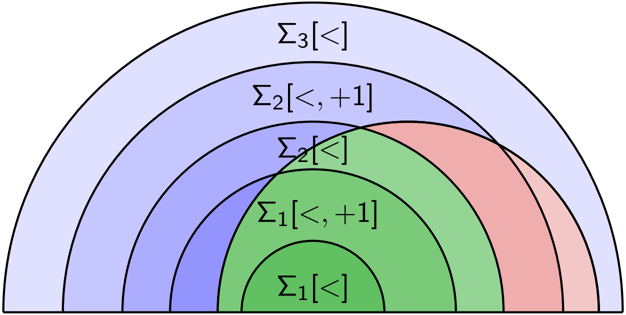
Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$.
  [Schützenberger 1976]
- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$.
  [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $FO^2[<]$. [Thérien, Wilke 1998]



Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$.
  [Schützenberger 1976]
- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$.
  [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathrm{FO}^2[<]$. [Thérien, Wilke 1998]



Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- L ist Vereinigung eindeutiger Monome über $\{\, B^* \mid B \subseteq A \,\}$. [Schützenberger 1976]
- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$. [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathrm{FO}^2[<]$. [Thérien, Wilke 1998]
- L wird von po2dfa akzeptiert. [Schwentick, Thérien, Vollmer 2001]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$. [Schützenberger 1976]
- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$. [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathrm{FO}^2[<]$. [Thérien, Wilke 1998]
- L wird von po2dfa akzeptiert. [Schwentick, Thérien, Vollmer 2001]
- L ist definierbar in eindeutiger Temporallogik $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$. [K. 2006]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$.
  [Schützenberger 1976]
- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$.
  [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $FO^2[<]$. [Thérien, Wilke 1998]
- L wird von po2dfa akzeptiert. [Schwentick, Thérien, Vollmer 2001]
- L ist definierbar in eindeutiger Temporallogik $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$.
  [K. 2006]
- L ist definierbar in eindeutiger Intervall-Logik $\mathrm{ITL}[\mathsf{F}_a, \mathsf{L}_a]$.
  [Lodaya, Pandya, Shah 2008]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$. [Schützenberger 1976]
- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$. [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathrm{FO}^2[<]$. [Thérien, Wilke 1998]
- L wird von po2dfa akzeptiert. [Schwentick, Thérien, Vollmer 2001]
- L ist definierbar in eindeutiger Temporallogik $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$. [K. 2006]
- L ist definierbar in eindeutiger Intervall-Logik $\mathrm{ITL}[\mathsf{F}_a, \mathsf{L}_a]$. [Lodaya, Pandya, Shah 2008]
- L ist im Abschluss von $\{ B^* \mid B \subseteq A \}$ unter Vereinigung und (co)deterministischen Produkten. [K., Weil 2010]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$.
  [Schützenberger 1976]
- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$.
  [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathrm{FO}^2[<]$. [Thérien, Wilke 1998]
- L wird von po2dfa akzeptiert. [Schwentick, Thérien, Vollmer 2001]
- L ist definierbar in eindeutiger Temporallogik $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$.
  [K. 2006]
- L ist definierbar in eindeutiger Intervall-Logik $\mathrm{ITL}[\mathsf{F}_a, \mathsf{L}_a]$.
  [Lodaya, Pandya, Shah 2008]
- L ist im Abschluss von $\{ B^* \mid B \subseteq A \}$ unter Vereinigung und (co)deterministischen Produkten. [K., Weil 2010]
- L ist definierbar in azyklischem $\Sigma_2[<, \leq]$. [K., Lauser 2012]

Die FO^2-Alternierungs-Hierarchie

$FO^2_m[<]$ = Alternierungstiefe m in $FO^2[<]$

Die FO^2-Alternierungs-Hierarchie

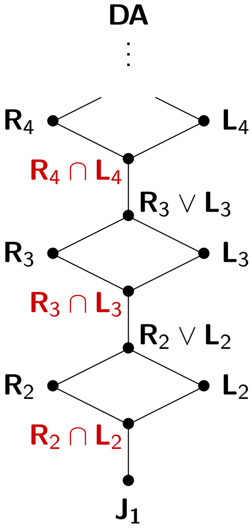
$\mathrm{FO}^2_m[<]$ = Alternierungstiefe m in $\mathrm{FO}^2[<]$

Satz [K., Weil 2012]: L ist genau dann in $\mathrm{FO}^2_m[<]$ definierbar, wenn das syntaktische Monoid von L in $\mathbf{R}_{m+1} \cap \mathbf{L}_{m+1}$ ist.

Die FO^2-Alternierungs-Hierarchie

$FO^2_m[<]$ = Alternierungstiefe m in $FO^2[<]$

Satz [K., Weil 2012]: L ist genau dann in $FO^2_m[<]$ definierbar, wenn das syntaktische Monoid von L in $\mathbf{R}_{m+1} \cap \mathbf{L}_{m+1}$ ist.

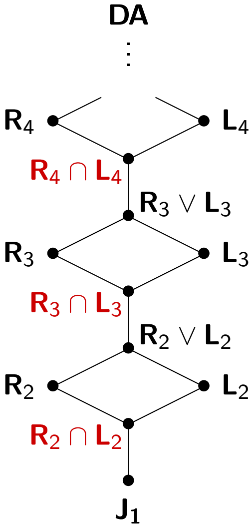


Die FO^2-Alternierungs-Hierarchie

$FO^2_m[<]$ = Alternierungstiefe m in $FO^2[<]$

Satz [K., Weil 2012]: L ist genau dann in $FO^2_m[<]$ definierbar, wenn das syntaktische Monoid von L in $\mathbf{R}_{m+1} \cap \mathbf{L}_{m+1}$ ist.

Korollar: Für jedes m ist es entscheidbar, ob eine reguläre Sprache $FO^2_m[<]$-definierbar ist.



Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$. [Schützenberger 1976]
- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$. [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathrm{FO}^2[<]$. [Thérien, Wilke 1998]
- L wird von po2dfa akzeptiert. [Schwentick, Thérien, Vollmer 2001]
- L ist definierbar in eindeutiger Temporallogik $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$. [K. 2006]
- L ist definierbar in eindeutiger Intervall-Logik $\mathrm{ITL}[\mathsf{F}_a, \mathsf{L}_a]$. [Lodaya, Pandya, Shah 2008]
- L ist im Abschluss von $\{ B^* \mid B \subseteq A \}$ unter Vereinigung und (co)deterministischen Produkten. [K., Weil 2010]
- L ist definierbar in azyklischem $\Sigma_2[<, \leq]$. [K., Lauser 2012]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.

kein

- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$. [Schützenberger 1976]
- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$. [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathrm{FO}^2[<]$. [Thérien, Wilke 1998]
- L wird von po2dfa akzeptiert. [Schwentick, Thérien, Vollmer 2001]
- L ist definierbar in eindeutiger Temporallogik $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$. [K. 2006]
- L ist definierbar in eindeutiger Intervall-Logik $\mathrm{ITL}[\mathsf{F}_a, \mathsf{L}_a]$. [Lodaya, Pandya, Shah 2008]
- L ist im Abschluss von $\{ B^* \mid B \subseteq A \}$ unter Vereinigung und (co)deterministischen Produkten. [K., Weil 2010]
- L ist definierbar in azyklischem $\Sigma_2[<, \leq]$. [K., Lauser 2012]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- kein $\left\{ \right.$ L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$. [Schützenberger 1976]
- ? L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$. [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathrm{FO}^2[<]$. [Thérien, Wilke 1998]
- L wird von po2dfa akzeptiert. [Schwentick, Thérien, Vollmer 2001]
- L ist definierbar in eindeutiger Temporallogik $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$. [K. 2006]
- L ist definierbar in eindeutiger Intervall-Logik $\mathrm{ITL}[\mathsf{F}_a, \mathsf{L}_a]$. [Lodaya, Pandya, Shah 2008]
- L ist im Abschluss von $\{ B^* \mid B \subseteq A \}$ unter Vereinigung und (co)deterministischen Produkten. [K., Weil 2010]
- L ist definierbar in azyklischem $\Sigma_2[<, \leq]$. [K., Lauser 2012]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.

kein

- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$. [Schützenberger 1976]

$\mathbf{R}_{m+1} \cap \mathbf{L}_{m+1}$

- **?** L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$. [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathsf{FO}^2[<]$. [Thérien, Wilke 1998]
- L wird von po2dfa akzeptiert. [Schwentick, Thérien, Vollmer 2001]
- L ist definierbar in eindeutiger Temporallogik $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$. [K. 2006]
- L ist definierbar in eindeutiger Intervall-Logik $\mathrm{ITL}[\mathsf{F}_a, \mathsf{L}_a]$. [Lodaya, Pandya, Shah 2008]
- L ist im Abschluss von $\{ B^* \mid B \subseteq A \}$ unter Vereinigung und (co)deterministischen Produkten. [K., Weil 2010]
- L ist definierbar in azyklischem $\Sigma_2[<, \leq]$. [K., Lauser 2012]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.

kein

- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$. [Schützenberger 1976]

?

- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]

$\mathbf{R}_{m+1} \cap \mathbf{L}_{m+1}$

- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$. [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathsf{FO}^2[<]$. [Thérien, Wilke 1998]

fuzzy

- L wird von po2dfa akzeptiert. [Schwentick, Thérien, Vollmer 2001]
- L ist definierbar in eindeutiger Temporallogik $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$. [K. 2006]
- L ist definierbar in eindeutiger Intervall-Logik $\mathrm{ITL}[\mathsf{F}_a, \mathsf{L}_a]$. [Lodaya, Pandya, Shah 2008]
- L ist im Abschluss von $\{ B^* \mid B \subseteq A \}$ unter Vereinigung und (co)deterministischen Produkten. [K., Weil 2010]
- L ist definierbar in azyklischem $\Sigma_2[<, \leq]$. [K., Lauser 2012]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.
- kein: L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$. [Schützenberger 1976]
- ? L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]
- $\mathbf{R}_{m+1} \cap \mathbf{L}_{m+1}$: L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$. [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathrm{FO}^2[<]$. [Thérien, Wilke 1998]
- fuzzy: L wird von po2dfa akzeptiert. [Schwentick, Thérien, Vollmer 2001]
- L ist definierbar in eindeutiger Temporallogik $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$. [K. 2006]
- $\mathbf{R}_m \vee \mathbf{L}_m$: L ist definierbar in eindeutiger Intervall-Logik $\mathrm{ITL}[\mathsf{F}_a, \mathsf{L}_a]$. [Lodaya, Pandya, Shah 2008]
- L ist im Abschluss von $\{ B^* \mid B \subseteq A \}$ unter Vereinigung und (co)deterministischen Produkten. [K., Weil 2010]
- L ist definierbar in azyklischem $\Sigma_2[<, \leq]$. [K., Lauser 2012]

Theorem [FO^2]: Für jede Sprache $L \subseteq A^*$ sind äquivalent:

- Das syntaktische Monoid von L ist in **DA**.

kein

- L ist Vereinigung eindeutiger Monome über $\{ B^* \mid B \subseteq A \}$. [Schützenberger 1976]

?

- L ist sowohl $\Sigma_2[<]$- als auch $\Pi_2[<]$-definierbar. [Pin, Weil 1997]

$\mathbf{R}_{m+1} \cap \mathbf{L}_{m+1}$

- L ist definierbar in unärer Temporallogik $\mathrm{TL}[\mathsf{XF}, \mathsf{YP}]$. [Etessami, Vardi, Wilke 1997]
- L ist definierbar in $\mathrm{FO}^2[<]$. [Thérien, Wilke 1998]

fuzzy

- L wird von po2dfa akzeptiert. [Schwentick, Thérien, Vollmer 2001]
- L ist definierbar in eindeutiger Temporallogik $\mathrm{TL}[\mathsf{X}_a, \mathsf{Y}_a]$. [K. 2006]

$\mathbf{R}_m \vee \mathbf{L}_m$

- L ist definierbar in eindeutiger Intervall-Logik $\mathrm{ITL}[\mathsf{F}_a, \mathsf{L}_a]$. [Lodaya, Pandya, Shah 2008]
- L ist im Abschluss von $\{ B^* \mid B \subseteq A \}$ unter Vereinigung und (co)deterministischen Produkten. [K., Weil 2010]

?

- L ist definierbar in azyklischem $\Sigma_2[<, \leq]$. [K., Lauser 2012]

Weitere Resultate

- Identitäten für $\mathrm{FO}^2_m[<]$ [Krebs, Straubing 2012]

Weitere Resultate

- Identitäten für $\mathsf{FO}^2_m[<]$ [Krebs, Straubing 2012]
- Identitäten für $\mathsf{FO}^2_m[<, +1]$ [K., Lauser 2013]

Weitere Resultate

- Identitäten für $\mathsf{FO}^2_m[<]$ [Krebs, Straubing 2012]
- Identitäten für $\mathsf{FO}^2_m[<,+1]$ [K., Lauser 2013]
- Identitäten für $\Sigma^2_m[<]$ [K., Lauser 2013]

Weitere Resultate

- Identitäten für $\mathrm{FO}^2_m[<]$ [Krebs, Straubing 2012]
- Identitäten für $\mathrm{FO}^2_m[<,+1]$ [K., Lauser 2013]
- Identitäten für $\Sigma^2_m[<]$ [K., Lauser 2013]
- $\mathrm{FO}^2[<]$ über unendlichen Wörtern [K., Diekert 2009]

Weitere Resultate

- Identitäten für $\mathsf{FO}^2_m[<]$ [Krebs, Straubing 2012]
- Identitäten für $\mathsf{FO}^2_m[<,+1]$ [K., Lauser 2013]
- Identitäten für $\Sigma^2_m[<]$ [K., Lauser 2013]
- $\mathsf{FO}^2[<]$ über unendlichen Wörtern [K., Diekert 2009]
- $\mathsf{FO}^2[<,+1]$ über unendlichen Wörtern [Kallas, K., Lauser 2011]

Weitere Resultate

- Identitäten für $\mathrm{FO}^2_m[<]$ [Krebs, Straubing 2012]
- Identitäten für $\mathrm{FO}^2_m[<,+1]$ [K., Lauser 2013]
- Identitäten für $\Sigma^2_m[<]$ [K., Lauser 2013]
- $\mathrm{FO}^2[<]$ über unendlichen Wörtern [K., Diekert 2009]
- $\mathrm{FO}^2[<,+1]$ über unendlichen Wörtern [Kallas, K., Lauser 2011]
- Abstrakter Fragmente-Begriff [K., Lauser 2012]

Weitere Resultate

- Identitäten für $\mathrm{FO}^2_m[<]$ [Krebs, Straubing 2012]
- Identitäten für $\mathrm{FO}^2_m[<,+1]$ [K., Lauser 2013]
- Identitäten für $\Sigma^2_m[<]$ [K., Lauser 2013]
- $\mathrm{FO}^2[<]$ über unendlichen Wörtern [K., Diekert 2009]
- $\mathrm{FO}^2[<,+1]$ über unendlichen Wörtern [Kallas, K., Lauser 2011]
- Abstrakter Fragmente-Begriff [K., Lauser 2012]
- Ehrenfeucht-Fraïssé-Spiele auf ω-Termen [Huschenbett, K.]

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable | st | s^ω

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable | st | s^ω
- Intuition: s^ω = „viele“ Iterationen von s, pumpbar

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable $|$ st $|$ s^ω
- Intuition: s^ω = „viele" Iterationen von s, pumpbar
- $L \models s = t$, falls für fast alle n: $p\,s[n!]\,q \in L \Leftrightarrow p\,t[n!]\,q \in L$

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable | st | s^ω
- Intuition: s^ω = „viele“ Iterationen von s, pumpbar
- $L \models s = t$, falls für fast alle n: $p\,s[n!]\,q \in L \Leftrightarrow p\,t[n!]\,q \in L$
- $\varphi \in \mathrm{FO}^2[<]$ impliziert $L(\varphi) \models (st)^\omega s (st)^\omega = (st)^\omega (st)^\omega$

Die Inklusion von $FO^2[<]$ in **DA**

- ω-Term ::= Variable | st | s^ω
- Intuition: s^ω = „viele“ Iterationen von s, pumpbar
- $L \models s = t$, falls für fast alle n: $p\,s[n!]\,q \in L \Leftrightarrow p\,t[n!]\,q \in L$
- $\varphi \in FO^2[<]$ impliziert $L(\varphi) \models (st)^\omega s (st)^\omega = (st)^\omega (st)^\omega$
- Sei $n >$ Quantorentiefe von φ, $u = (st)^n s (st)^n$, $v = (st)^n (st)^n$

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable $\mid st \mid s^\omega$
- Intuition: s^ω = „viele“ Iterationen von s, pumpbar
- $L \models s = t$, falls für fast alle n: $p\,s[n!]\,q \in L \Leftrightarrow p\,t[n!]\,q \in L$
- $\varphi \in \mathrm{FO}^2[<]$ impliziert $L(\varphi) \models (st)^\omega s (st)^\omega = (st)^\omega (st)^\omega$
- Sei $n >$ Quantorentiefe von φ, $u = (st)^n s (st)^n$, $v = (st)^n (st)^n$
- k-Rand: $(st)^k \boxed{} (st)^k$

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable | st | s^ω
- Intuition: s^ω = „viele“ Iterationen von s, pumpbar
- $L \models s = t$, falls für fast alle n: $ps[n!]q \in L \Leftrightarrow pt[n!]q \in L$
- $\varphi \in \mathrm{FO}^2[<]$ impliziert $L(\varphi) \models (st)^\omega s (st)^\omega = (st)^\omega (st)^\omega$
- Sei $n >$ Quantorentiefe von φ, $u = (st)^n s (st)^n$, $v = (st)^n (st)^n$
- k-Rand: $(st)^k \boxed{} (st)^k$
- Invariante: Bei verbleibender Tiefe $k < n$:
  exakt auf dem k-Rand, beliebig aber isomorph innen

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable | st | s^ω
- Intuition: s^ω = „viele“ Iterationen von s, pumpbar
- $L \models s = t$, falls für fast alle n: $p\,s[n!]\,q \in L \Leftrightarrow p\,t[n!]\,q \in L$
- $\varphi \in \mathrm{FO}^2[<]$ impliziert $L(\varphi) \models (st)^\omega s (st)^\omega = (st)^\omega (st)^\omega$
- Sei $n >$ Quantorentiefe von φ, $u = (st)^n s (st)^n$, $v = (st)^n (st)^n$
- k-Rand: $(st)^k \boxed{} (st)^k$
- Invariante: Bei verbleibender Tiefe $k < n$:
  exakt auf dem k-Rand, beliebig aber isomorph innen

- Spiel: $u = (st)^k (st) \boxed{} (st)(st)^k$

  $v = (st)^k (st) \boxed{} (st)(st)^k$

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable | st | s^ω
- Intuition: s^ω = „viele“ Iterationen von s, pumpbar
- $L \models s = t$, falls für fast alle n: $p\,s[n!]\,q \in L \Leftrightarrow p\,t[n!]\,q \in L$
- $\varphi \in \mathrm{FO}^2[<]$ impliziert $L(\varphi) \models (st)^\omega s (st)^\omega = (st)^\omega (st)^\omega$
- Sei $n >$ Quantorentiefe von φ, $u = (st)^n s (st)^n$, $v = (st)^n (st)^n$
- k-Rand: $(st)^k \boxed{\qquad\qquad\qquad} (st)^k$
- Invariante: Bei verbleibender Tiefe $k < n$:
  exakt auf dem k-Rand, beliebig aber isomorph innen
- Spiel: $u = (st)^k (st) \boxed{\qquad\overset{x}{\downarrow}\qquad\overset{y}{\downarrow}\qquad} (st)(st)^k$

  $v = (st)^k (st) \boxed{\qquad\overset{x}{\downarrow}\qquad\qquad\overset{y}{\downarrow}} (st)(st)^k$

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable | st | s^ω
- Intuition: s^ω = „viele“ Iterationen von s, pumpbar
- $L \models s = t$, falls für fast alle n: $p\,s[n!]\,q \in L \Leftrightarrow p\,t[n!]\,q \in L$
- $\varphi \in \mathrm{FO}^2[<]$ impliziert $L(\varphi) \models (st)^\omega s (st)^\omega = (st)^\omega (st)^\omega$
- Sei $n >$ Quantorentiefe von φ, $u = (st)^n s (st)^n$, $v = (st)^n (st)^n$
- k-Rand: $(st)^k \boxed{\qquad\qquad\qquad\qquad} (st)^k$
- Invariante: Bei verbleibender Tiefe $k < n$:
  exakt auf dem k-Rand, beliebig aber isomorph innen

- Spiel: $u = (st)^k (st) \boxed{\qquad\qquad\qquad\overset{y}{\downarrow}\qquad} (st)(st)^k$

  $v = (st)^k (st) \boxed{\qquad\overset{x}{\downarrow}\qquad\qquad\overset{y}{\downarrow}} (st)(st)^k$

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable | st | s^ω
- Intuition: s^ω = „viele“ Iterationen von s, pumpbar
- $L \models s = t$, falls für fast alle n: $ps[n!]q \in L \Leftrightarrow pt[n!]q \in L$
- $\varphi \in \mathrm{FO}^2[<]$ impliziert $L(\varphi) \models (st)^\omega s (st)^\omega = (st)^\omega (st)^\omega$
- Sei $n >$ Quantorentiefe von φ, $u = (st)^n s (st)^n$, $v = (st)^n (st)^n$
- k-Rand: $(st)^k \boxed{} (st)^k$
- Invariante: Bei verbleibender Tiefe $k < n$:
  exakt auf dem k-Rand, beliebig aber isomorph innen
- Spiel: $u = (st)^k (st) \boxed{\overset{y}{\downarrow}} (st)(st)^k$

  $v = (st)^k (st) \boxed{\overset{y}{\downarrow}} (st)(st)^k$

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable | st | s^ω
- Intuition: s^ω = „viele“ Iterationen von s, pumpbar
- $L \models s = t$, falls für fast alle n: $p\,s[n!]\,q \in L \Leftrightarrow p\,t[n!]\,q \in L$
- $\varphi \in \mathrm{FO}^2[<]$ impliziert $L(\varphi) \models (st)^\omega s (st)^\omega = (st)^\omega (st)^\omega$
- Sei $n >$ Quantorentiefe von φ, $u = (st)^n s (st)^n$, $v = (st)^n (st)^n$
- k-Rand: $(st)^k \boxed{\qquad\qquad\qquad\qquad} (st)^k$
- Invariante: Bei verbleibender Tiefe $k < n$:
  exakt auf dem k-Rand, beliebig aber isomorph innen
- Spiel: $u = (st)^k (st) \boxed{\qquad\qquad\qquad\overset{y}{\downarrow}\;\overset{x}{\downarrow}} (st)(st)^k$

  $v = (st)^k (st) \boxed{\qquad\qquad\qquad\qquad\overset{y}{\downarrow}} (st)(st)^k$

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable | st | s^ω
- Intuition: s^ω = „viele“ Iterationen von s, pumpbar
- $L \models s = t$, falls für fast alle n: $p\,s[n!]\,q \in L \Leftrightarrow p\,t[n!]\,q \in L$
- $\varphi \in \mathrm{FO}^2[<]$ impliziert $L(\varphi) \models (st)^\omega s (st)^\omega = (st)^\omega (st)^\omega$
- Sei $n >$ Quantorentiefe von φ, $u = (st)^n s (st)^n$, $v = (st)^n (st)^n$
- k-Rand: $(st)^k \boxed{\qquad\qquad\qquad\qquad} (st)^k$
- Invariante: Bei verbleibender Tiefe $k < n$:
  exakt auf dem k-Rand, beliebig aber isomorph innen
- Spiel: $u = (st)^k (st) \boxed{\qquad\qquad\qquad\quad \overset{y}{\downarrow}\;\overset{x}{\downarrow}} (st)(st)^k$

  $v = (st)^k (st) \boxed{\qquad\qquad\qquad\quad \overset{y}{\downarrow}} (\overset{x}{\downarrow}\!st)(st)^k$

Die Inklusion von $\mathrm{FO}^2[<]$ in **DA**

- ω-Term ::= Variable | st | s^ω
- Intuition: s^ω = „viele“ Iterationen von s, pumpbar
- $L \models s = t$, falls für fast alle n: $ps[n!]q \in L \Leftrightarrow pt[n!]q \in L$
- $\varphi \in \mathrm{FO}^2[<]$ impliziert $L(\varphi) \models (st)^\omega s (st)^\omega = (st)^\omega (st)^\omega$
- Sei $n >$ Quantorentiefe von φ, $u = (st)^n s (st)^n$, $v = (st)^n (st)^n$
- k-Rand: $(st)^k \boxed{} (st)^k$
- Invariante: Bei verbleibender Tiefe $k < n$:
  exakt auf dem k-Rand, beliebig aber isomorph innen
- Spiel: $u = (st)^k (st) \boxed{\overset{y}{\downarrow}\;\overset{x}{\downarrow}} (st)(st)^k$

  $v = (st)^k (st) \boxed{\overset{y}{\downarrow}} (\overset{x}{\downarrow}st)(st)^k$
- Idee: Bei $(st)^\omega s (st)^\omega$ und $(st)^\omega (st)^\omega$ werden die Züge am Rand kopiert, in der Mitte ist es egal.

Ausblick

- Effiziente Algorithmen durch Verbotsmuster

Ausblick

- Effiziente Algorithmen durch Verbotsmuster
- Block-Produkte und Trotter-Weil-Hierarchie

Ausblick

- Effiziente Algorithmen durch Verbotsmuster
- Block-Produkte und Trotter-Weil-Hierarchie
- Fortschritt bei FO-Alternierungs-Hierarchie??

Ausblick

- Effiziente Algorithmen durch Verbotsmuster
- Block-Produkte und Trotter-Weil-Hierarchie
- Fortschritt bei FO-Alternierungs-Hierarchie??
- Unendliche Wörter

Ausblick

- Effiziente Algorithmen durch Verbotsmuster
- Block-Produkte und Trotter-Weil-Hierarchie
- Fortschritt bei FO-Alternierungs-Hierarchie??
- Unendliche Wörter
- Eigenschaften abstrakter Fragmente

Ausblick

- Effiziente Algorithmen durch Verbotsmuster
- Block-Produkte und Trotter-Weil-Hierarchie
- Fortschritt bei FO-Alternierungs-Hierarchie??
- Unendliche Wörter
- Eigenschaften abstrakter Fragmente
- Abstrakte temporallogische Fragmente

Danke für die Aufmerksamkeit!