

FGI-1 – Formale Grundlagen der Informatik I

Logik, Automaten und Formale Sprachen

Musterlösung 11 : — Resolution, Turingmaschinen

Präsenzaufgabe 11.1 Es sei P ein zweistelliges Prädikatensymbol und x, y, z Variablen. Weiterhin seien folgende Formeln definiert:

$$\begin{aligned}F_1 &= \forall x \forall y (P(x, y) \Rightarrow \neg P(y, x)) \\F_2 &= \forall x \neg P(x, x) \\F_3 &= \forall x \forall y \forall z ((P(x, y) \wedge P(y, z)) \Rightarrow P(x, z))\end{aligned}$$

Zeigen Sie unter Verwendung des prädikatenlogischen Resolutionsverfahrens die folgenden Behauptungen:

1. $F_1 \models F_2$

Hilfestellung: Die Mengendarstellung einer Klauselnormalform von $(F_1 \wedge \neg F_2)$ ist $\{\{\neg P(x_1, y_1), \neg P(y_1, x_1)\}, \{P(a, a)\}\}$.

Erläutern Sie, warum Ihnen diese Information nützlich ist.

Lösung $F_1 \models F_2$ genau dann, wenn $(F_1 \wedge \neg F_2)$ unerfüllbar ist.

Der Vollständigkeit halber:

Umformung von $(F_1 \wedge \neg F_2)$ in Klauselnormalform:

$$(F_1 \wedge \neg F_2) = (\forall x \forall y (P(x, y) \Rightarrow \neg P(y, x)) \wedge \neg \forall x \neg P(x, x))$$

Elimination von \Rightarrow (Äquivalenzumformung):

$$\equiv (\forall x \forall y (\neg P(x, y) \vee \neg P(y, x)) \wedge \neg \forall x \neg P(x, x))$$

Bereinigung (gebundene Umbenennung der Variablen) (Äquivalenzumformung):

$$\equiv (\forall x_1 \forall y_1 (\neg P(x_1, y_1) \vee \neg P(y_1, x_1)) \wedge \neg \forall x_2 \neg P(x_2, x_2))$$

Pränexform (Skopuserweiterung) (Äquivalenzumformung):

$$\equiv \exists x_2 \forall x_1 \forall y_1 ((\neg P(x_1, y_1) \vee \neg P(y_1, x_1)) \wedge \neg \neg P(x_2, x_2))$$

Skolemisierung (Erfüllbarkeitsäquivalenz): x_2 : Skolemkonstante a ;

$$\forall x_1 \forall y_1 ((\neg P(x_1, y_1) \vee \neg P(y_1, x_1)) \wedge \neg \neg P(a, a))$$

Konjunktive Normalform der Matrix (Äquivalenzumformung):

$$\equiv \forall x_1 \forall y_1 ((\neg P(x_1, y_1) \vee \neg P(y_1, x_1)) \wedge P(a, a))$$

Klauselnormalform in Mengendarstellung:

$$\{\{\neg P(x_1, y_1), \neg P(y_1, x_1)\}, \{P(a, a)\}\}$$

Resolution:

$$\begin{array}{ccc}\{\neg P(x_1, y_1), \neg P(y_1, x_1)\} & & \{P(a, a)\} \\ \downarrow [x_1/a, y_1/a] & \swarrow & \\ \square & & \end{array}$$

2. $\{F_2, F_3\} \models F_1$

Hilfestellung: Die Mengendarstellung einer Klauselnormalform von $((F_2 \wedge F_3) \wedge \neg F_1)$ ist $\{\{\neg P(x_1, y_1), \neg P(y_1, z_1), P(x_1, z_1)\}, \{\neg P(x_2, x_2)\}, \{P(a, b)\}, \{P(b, a)\}\}$.

Erläutern Sie, warum Ihnen diese Information nützlich ist.

Lösung $\{F_2, F_3\} \models F_1$ genau dann, wenn $((F_2 \wedge F_3) \wedge \neg F_1)$ unerfüllbar ist.

Der Vollständigkeit halber:

Umformung von $((F_3 \wedge F_2) \wedge \neg F_1)$ in Klauselnormalform:

$$\begin{aligned} ((F_3 \wedge F_2) \wedge \neg F_1) = \\ ((\forall x \forall y \forall z ((P(x, y) \wedge P(y, z)) \Rightarrow P(x, z)) \wedge \forall x \neg P(x, x)) \wedge \neg \forall x \forall y (P(x, y) \Rightarrow \neg P(y, x))) \end{aligned}$$

Elimination von \Rightarrow (Äquivalenzumformung):

$$\equiv ((\forall x \forall y \forall z (\neg(P(x, y) \wedge P(y, z)) \vee P(x, z)) \wedge \forall x \neg P(x, x)) \wedge \neg \forall x \forall y (\neg P(x, y) \vee \neg P(y, x)))$$

Bereinigung (gebundene Umbenennung der Variablen) (Äquivalenzumformung):

$$\equiv ((\forall x_1 \forall y_1 \forall z_1 (\neg(P(x_1, y_1) \wedge P(y_1, z_1)) \vee P(x_1, z_1)) \wedge \forall x_2 \neg P(x_2, x_2)) \wedge \neg \forall x_3 \forall y_2 (\neg P(x_3, y_2) \vee \neg P(y_2, x_3)))$$

Pränexform (Skopuserweiterung) (Äquivalenzumformung):

$$\equiv \exists x_3 \exists y_2 \forall x_1 \forall y_1 \forall z_1 \forall x_2 (((\neg(P(x_1, y_1) \wedge P(y_1, z_1)) \vee P(x_1, z_1)) \wedge \neg P(x_2, x_2)) \wedge \neg(\neg P(x_3, y_2) \vee \neg P(y_2, x_3)))$$

Skolemisierung (Erfüllbarkeitsäquivalenz): x_3 : Skolemkonstante **a**; y_2 : Skolemkonstante **b**;

$$\forall x_1 \forall y_1 \forall z_1 \forall x_2 (((\neg(P(x_1, y_1) \wedge P(y_1, z_1)) \vee P(x_1, z_1)) \wedge \neg P(x_2, x_2)) \wedge \neg(\neg P(a, b) \vee \neg P(b, a)))$$

Konjunktive Normalform der Matrix (Äquivalenzumformung):

$$\equiv \forall x_1 \forall y_1 \forall z_1 \forall x_2 (((\neg P(x_1, y_1) \vee \neg P(y_1, z_1)) \vee P(x_1, z_1)) \wedge \neg P(x_2, x_2)) \wedge (P(a, b) \wedge P(b, a)))$$

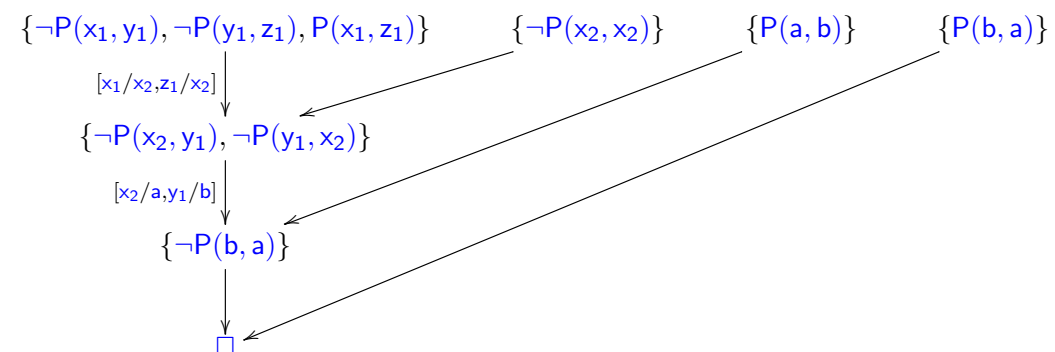
Klammerersparnis, um die Struktur deutlicher zu machen:

$$\equiv \forall x_1 \forall y_1 \forall z_1 \forall x_2 ((\neg P(x_1, y_1) \vee \neg P(y_1, z_1) \vee P(x_1, z_1)) \wedge \neg P(x_2, x_2) \wedge P(a, b) \wedge P(b, a))$$

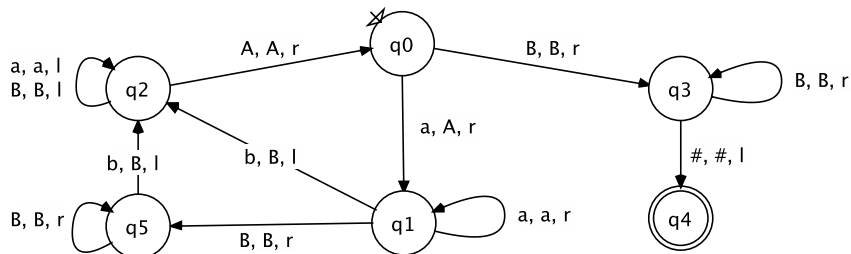
Klauselnormalform in Mengendarstellung:

$$\{\{\neg P(x_1, y_1), \neg P(y_1, z_1), P(x_1, z_1)\}, \{\neg P(x_2, x_2)\}, \{P(a, b)\}, \{P(b, a)\}\}$$

N-Resolution:



Präsenzaufgabe 11.2 Betrachten Sie folgende Turingmaschine A mit $\Sigma = \{a, b\}$ und $\Gamma = \Sigma \cup \{A, B, \#\}$.



1. Geben Sie eine maximale Rechnung von A auf der Eingabe $w = aabb$ an.

Lösung

$$\begin{aligned}
 & q_0 aabb \\
 \vdash & Aq_1abb \\
 \vdash & Aaq_1bb \\
 \vdash & Aq_2aBb \\
 \vdash & q_2AaBb \\
 \vdash & Aq_0aBb \\
 \vdash & AAq_1Bb \\
 \vdash & AABq_5b \\
 \vdash & AAq_2BB \\
 \vdash & Aq_2ABB \\
 \vdash & AAq_0BB \\
 \vdash & AABq_3B \\
 \vdash & AABq_3\# \\
 \vdash & AABq_4B
 \end{aligned}$$

Dies ist eine Erfolgsrechnung.

2. Geben Sie eine maximale Rechnung von A auf der Eingabe $w = abb$ an.

Lösung

$$q_0abb \vdash Aq_1bb \vdash q_2ABb \vdash Aq_0Bb \vdash ABq_3b$$

Keine Erfolgsrechnung, aber Termination.

3. Geben Sie zu jedem Zustand eine inhaltliche Beschreibung an, was dieser leistet.

Lösung

- q_0 : Kopf ist soweit nach links zurückgefahren, dass er jetzt rechts neben einem A steht (oder initial ganz links neben dem $\#$).
- q_1 : Wir haben ein a gelesen und überspringen jetzt alles a nach rechts, bis wir auf ein b oder ein B stoßen.
- q_5 : Wenn wir ein B gelesen haben, dann überspringen wir jetzt alle B , bis wir auf ein b stoßen.
- q_2 : Nachdem wir ein b markiert haben, überspringen wir alle markierten B 's und unmarkierten a 's nach links, bis wir auf ein markiertes A stoßen.

- q_3 : Wir gelangen nach q_3 , wenn wir alle a markiert haben, da dann das erste Zeichen rechts neben einem A ein B ist. Mindestens ein b muss also bereits markiert worden sein. Wir überspringen jetzt alle B nach rechts, um zu überprüfen, ob rechts von den B 's noch etwas steht.
- q_4 : Wir gelangen nach q_4 , wenn wir nach den B 's nichts mehr haben, also nur das $\#$. In diesem Fall akzeptieren wir.

4. Welche Sprache akzeptiert die obige TM?

Lösung Es werden alle Worte der Form $a^n b^n$ akzeptiert.

Ausführliche Begründung: Wir betrachten ein akzeptiertes Wort w , d.h. es gibt u und v , so dass $q_0 w \vdash^* u q_4 v$ gilt.

In der Schleife von q_0 nach q_0 wird ein a und ein b markiert. Hat die Konfiguration nach k Schleifendurchläufen von q_0 nach q_0 die Form:

$$A^k q_0 \alpha B^k \beta \quad \text{mit} \quad \alpha, \beta \in \{a, b\}^*,$$

dann gilt sogar:

$$\alpha \in \{a\}^* \wedge \beta \in \{b\}^* \wedge |\alpha| = |\beta|$$

Induktion über $|\alpha|$:

- Ind.Anfang: Wenn $\alpha = \epsilon$, dann muss auch $\beta = \epsilon$ gelten, denn um in q_4 zu terminieren darf nach den B^k nur noch das $\#$ folgen.
- Ind.Schritt: Wenn $\alpha \neq \epsilon$, dann muss es mit a beginnen, da wir q_0 nur mit a oder B verlassen können. Wenn wir ein a markieren, dann müssen wir auch beim Übergang nach q_2 ein b markieren. Es ist dann $\alpha = a\alpha'$ und $\beta = b\beta'$. Wir erreichen dann die Konfiguration:

$$A^{k+1} q_0 \alpha' B^{k+1} \beta'$$

Da α' jetzt kürzer als α ist, gilt die Induktionsannahme, dass α' nur aus a 's besteht und β nur aus b 's. Dies gilt dann auch für $\alpha = a\alpha'$ und $\beta = b\beta'$. Längengleichheit ist auch offensichtlich.

Da jedes A ein a war und jedes B ein b , hatten wir initial ($k = 0$) also die Konfiguration $q_0 a^n b^n$. Also:

$$L(A) = \{a^n b^n \mid n \geq 1\}$$

5. Was würde sich ändern, wenn auch q_3 Endzustand wäre?

Lösung Dann würde die neue TM A' schon akzeptieren, sobald ein Wort einen Präfix der Form $a^n b^n$ besitzt: $L(A') = L(A)\{a, b\}^*$, denn eine TM akzeptiert ihre Eingabe, sobald sie einen Endzustand durchläuft. Es ist hierbei nicht notwendig (anders als bei NFA oder PDA), dass sie die ganze Eingabe gelesen hätte.

Präsenzaufgabe 11.3 Geben Sie jeweils die Funktionsweise einer DTM an, die folgende Funktionen berechnet:

$$f_1 : \{a\}^* \rightarrow \{a\}^*, \quad f_1(a^n) := a^{n+1}, n > 0$$

und

$$f_2 : \{a\}^* \rightarrow \{b\}^*, \quad f_2(a^n) := b^{2n}, n > 0$$

Lösung f_1 : Wir kopieren vor die Eingabe ein weiteres a .

Imperative Formulierung der Funktionsweise:

1. q_0 : Lese irgendein Zeichen, schreibe es wieder zurück und gehe nach links. Wechsle nach q_1 .
2. q_1 : Lese ein $\#$, schreibe ein a und gehe nach links. Wechsle in den Endzustand q_2 .
3. q_2 : Keine Übergänge.

f_2 : Zu jedem a der Eingabe kopieren wir genausoviele b 's ans Ende an wie die Eingabe lang ist.

Imperative Formulierung der Funktionsweise:

1. Überlaufe ggf. alle A nach rechts. (Wir suchen das erste noch unmarkierte a .)
2. Fallunterscheidung bzgl. des ersten Zeichens $x \neq A$:
 - (a) Wenn $x = \#$, dann war die Eingabe $w = a^0 = \epsilon$. Terminiere im Endzustand.
 - (b) Wenn $x = a$, dann markiere es als A .
 - (c) Erweitere folgendermaßen am rechten Ende um ein b :
 - i. Laufe nach rechts bis zum ersten $\#$.
 - ii. Überschreibe $\#$ mit einem b .
 - iii. Laufe nach links bis zum ersten $\#$.
 - iv. Starte erneut in Schritt 3.
 - (d) Wenn $x = b$, dann überschreibe nach links laufend jedes A mit einem b . Terminiere beim ersten $\#$ im Endzustand. Der LSK steht dann vor der Ausgabe.

Version vom 15. Juni 2012