4 Komplexität der elementaren Theorie

Satz 2.14 liefert also einen Algorithmus mit:
Eingabe: automatische Präsentation P
FO-Satz φ

Ausgabe: $\mathcal{A}(P) \models \varphi$ bzw. $\mathcal{A}(P) \models \neg\varphi$

Definition 4.1. Sei $\mathcal{P}$ eine Menge von automatischen Präsentationen und $\mathcal{F} \subseteq \mathrm{FO}$ eine Menge von Formeln der Logik erster Stufe. Das *Auswertungsproblem für $\mathcal{P}$ und $\mathcal{F}$* ist die folgende Menge:

$$\mathrm{MC}(\mathcal{P}, \mathcal{F}) = \left\{(P, \varphi) \in \mathcal{P} \times \mathcal{F} \mid \varphi \text{ Satz mit } \mathcal{A}(P) \models \varphi\right\}.$$

Bemerkung. MC steht für „model checking problem“, die im Englischen übliche Bezeichnung für das Auswertungsproblem.

Wir untersuchen die Komplexität dieses Problems in verschiedenen Ausprägungen:

- eingeschränkte Präsentationen:
  - $|\mathcal{P}| = 1$, d.h. nur eine feste Präsentation - „Ausdruckskomplexität“
  - beliebige Präsentation mit „beschränktem Grad“
- eingeschränkte Sätze
  - $|\mathcal{F}| = 1$, d.h. ein fester Satz - „Datenkomplexität“
  - beliebige Sätze mit „beschränkter Quantorenalternierung“

Notationen $\mathrm{Poly} = \mathrm{Exp}_0 = \bigcup_{k\in\mathbb{N}} O(n^k)$ ist die Menge der polynomiell beschränkten Funktionen $\mathbb{N} \to \mathbb{N}$, $\mathrm{Exp}_{n+1} = \bigcup_{f\in\mathrm{Exp}_n} 2^{O(f)}$ ist die Menge der $(n+1)$-fach exponentiellen Funktionen

$n-\mathsf{EXPSPACE}$ ist die Menge der durch eine deterministische TM mit Platzbeschränkung $p \in \mathrm{Exp}_n$ entscheidbaren Probleme, insbesondere $\mathsf{PSPACE} = 0-\mathsf{EXPSPACE}$.

$\Sigma_0 \subseteq \mathrm{FO}$ ist die Menge der quantorenfreien Formeln (d.h. Boolesche Kombination von Formeln $x = y$ und $R(x_1, \ldots, x_k)$).

$B\Sigma_n$ ist die Menge von Booleschen Kombinationen von Formeln aus Σ_n (insbesondere $B\Sigma_0 = \Sigma_0$)

Σ_{n+1} ist der Abschluß der Klasse $B\Sigma_n$ unter existentieller Quantifikation, Konjunktion und Disjunktion

AutPr ist die Klasse aller automatischen Präsentationen.

Der Beweis von Satz 2.14 liefert

Korollar 4.2. *Für jedes* $n \geq 1$ *ist das Auswertungsproblem* $\mathrm{MC}(\mathrm{AutPr}, \Sigma_n)$ *in* $(n-1)-\mathsf{EXPSPACE}$.

Wir werden untersuchen, ob bzw. in welchen Situationen sich dieses Ergebnis verbessern läßt.

4.1 Datenkomplexität

Wir untersuchen also die Komplexität von $\mathrm{MC}(\mathcal{P}, \mathcal{F})$ mit $\mathcal{F} = \{\varphi\}$. Für $\varphi \in \Sigma_{n+1}$ ergibt Korollar 4.2 dann $\mathrm{MC}\big(\mathrm{AutPr}, \{\varphi\}\big) \in n{-}\mathsf{EXPSPACE}$. Im allgemeinen läßt sich dieses Ergebnis nicht verbessern:

Satz 4.3. *Für alle $n \in \mathbb{N}$ existiert Satz $\varphi_n \in \Sigma_{n+1}$, so daß die Menge $\mathrm{MC}\big(\mathrm{AutPr}, \{\varphi_n\}\big)$ der automatischen Präsentationen P mit $\mathcal{A}(P) \models \varphi_n$ vollständig für die Klasse $n{-}\mathsf{EXPSPACE}$ ist.*

Beweisstrategie für Satz 4.3 zunächst definieren wir Funktionen $F_n \colon \mathbb{N} \to \mathbb{N}$ in Exp_n induktiv:

$$F_0(m) = m \text{ und } F_{n+1}(m) = F_n(m) \cdot 2^{F_n(m)}$$

Ab jetzt sei $n \in \mathbb{N}$ fest.

Wir fixieren eine $\big(F_n(m){-}2\big)$-platzbeschränkte TM M_n, die ein $n{-}\mathsf{EXPSPACE}$-vollständiges Problem $L(M_n)$ entscheidet. (ohne Beweis: eine solche TM existiert tatsächlich)

Sei w Eingabewort für M_n. Wir konstruieren in polynomieller Zeit automatische Präsentation P_w und Satz $\varphi_n \in \Sigma_{n+1}$ mit $\mathcal{A}(P_w) \models \varphi_n \iff w \in L(M_n)$.

Da Satz φ_n *nicht* von w abhängt, ist dies eine Polynomialzeitreduktion von $L(M_n)$ auf $\mathrm{MC}(\mathrm{AutPr}, \{\varphi_n\})$.

Notationen Q Zustandsmenge, q_0 Initialzustand, q_f akzeptierender Zustand, Γ Bandalphabet und $\diamond$ Leerzeichen von M, $\$, 0, 1 \notin Q \cup \Gamma$, $\Delta = \{\$\} \cup Q \cup \Gamma$

Die Turingmaschine akzeptiert, wenn sie den Zustand q_f erreicht hat.

Konfiguration: Wort $c = upv \in \Gamma^* Q \Gamma^+$ (Idee: auf Band steht uv, Zustand ist p, Kopf steht auf erster Position von $v \in \Gamma^+$)

m-Konfiguration (für $m \in \mathbb{N}$): Konfiguration $upv \in \Gamma^* Q \Gamma^+$ mit $|uv| = F_n(m) - 2$, d.h. $|upv| = F_n(m) - 1$

Sei $w \in \Sigma^*$ ein Eingabewort für die TM M_n und sei m seine Länge. Wir beschreiben zunächst eine automatische Struktur $\mathcal{A}_w$ mit Universum $\big(Q \cup \Gamma \cup \{0, 1, \$\}\big)^* = \big(\Delta \cup \{0, 1\}\big)^*$:

Sie enthält die folgenden synchron-rationalen Relationen:

1. StepInBlocks umfaßt alle Paare (c, c') mit
   - $c = \$c_0\$c_1\$\dots\$c_k\$$ und $c' = \$c'_0\$c'_1\$\dots\$c'_k\$$ für Konfigurationen $c_i, c'_i \in \Gamma^*Q\Gamma^+$,
   - $c_i \vdash c'_i$ und $|c_i| = |c'_i|$ für alle $0 \leq i \leq k$,
   - $c_0 \in q_0w\diamond^*$ (diese Relation hängt also vom Eingabewort w ab),
   - $c'_k \in \Gamma^*q_f\Gamma^+$
2. EqLet umfaßt alle Tripel $(x, y, z) \in \{0, 1\}^* \times \Delta^* \times \Delta^*$
   - x enthält genau zwei Einsen an den Stellen $k_1 < k_2$
   - Buchstabe an Stelle k_1 in y und Buchstabe an Stelle k_2 in z stimmen überein

3. $\text{Replace}^{\otimes} = (\$, 1)\big((Q \cup \Gamma) \times \{0\}\big)^*(\$, 1)\big(\Delta \times \{0\}\big)^*$, d.h. Replace ist die Menge aller Paare (c, x), wobei c mit \$ beginnt und wenigstens ein weiteres \$ enthält und x aus c entsteht, indem die ersten beiden \$ durch 1 und die anderen Buchstaben durch 0 ersetzt werden

4. $L_n = 0^*10^{F_n(m)-1}10^*$, d.h., die Menge aller Wörter mit genau zwei Einsen an den Stellen k_1 und $k_1 + F_n(m)$ (da m die Länge von w ist, hängt auch diese unäre Relation vom Eingabewort w ab)

5. der Längenvergleich $|u| \leq |v|$

Betrachte den folgenden Satz, der über die Struktur $\mathcal{A}_w$ spricht:

$$\begin{aligned}\psi = \exists c, c' \colon &\mathrm{StepInBlocks}(c, c') \\ &\wedge \exists x \colon \mathrm{Replace}(c, x) \wedge L_n(x) \\ &\wedge \forall x' \colon \big((L_n(x') \wedge |x'| \leq |c'|) \to \mathrm{EqLet}(x', c', c)\big)\end{aligned}$$

Erläuterung:

- 1. Zeile besagt, daß es $k \in \mathbb{N}$ und Konfigurationen c_i und c_i' gibt mit
  - $c = \$c_0\$\dots\$c_k\$$ und $c' = \$c_0'\$\dots\$c_k'\$$,
  - $c_i \vdash c_i'$ und $|c_i| = |c_i'|$ für alle $0 \leq i \leq k$ und
  - $c_0 \in q_0 w \diamond^*$ und $c_k' \in \Gamma^* q_f \Gamma^+$
- 2. Zeile besagt: $|c_0| = F_n(m) - 1$
- 3. Zeile: Buchstabe Nummer i in c' und Nummer $i + F_n(m)$ in c stimmen überein (für alle $0 \leq i \leq |c'| - F_n(m)$),

Gilt dieser Satz in $\mathcal{A}_w$, so gelten (wobei $c'[i,j]$ der Faktor von c' ist, der an der Stelle i beginnt und an der Stelle j endet):

$$\begin{aligned}
\$c_0'\$ &= c'[0, F_n(m)] && \text{denn } F_n(m) - 1 = |c_0| = |c_0'| \\
&= c[F_n(m), 2F_n(m)] && \text{nach 3. Zeile von } \psi \\
&= \$c_1\$ && \text{da } c[F_n(m), 2F_n(m)] \text{ Präfix von } \$c_1\$c_2 \cdots \$c_k\$ \text{ ist}
\end{aligned}$$

Also haben wir $c_0' = c_1$.

Insbesondere folgt $F_n(m) - 1 = |c_0| = |c_0'| = |c_1|$. Daher können wir induktiv fortfahren und erhalten $c_i' = c_{i+1}$ für alle $0 \leq i < k$.

Damit haben wir aber $c_0 \vdash c_0' = c_1 \vdash c_1' = c_2 \vdash \cdots c_{k-1}' = c_k \vdash c_k'$, woraus folgt, daß w von M_n akzeptiert wird. Wir haben also die Implikation

$$\mathcal{A}_w \models \psi \implies w \in L(M_n)$$

gezeigt.

Gelte umgekehrt $w \in L(M_n)$. Dann existieren m-Konfigurationen $c_0 \vdash c_1 \vdash \cdots \vdash c_k \vdash c'_k$ mit $c_0 \in q_0 w \diamond^*$ und $c'_k \in \Gamma^* q_f \Gamma^+$. Mit $c = \$c_0\$c_1\$\ldots\$c_k\$$ und $c' = \$c_1\$\ldots\$c_k\$c'_k\$$ erhält man Zeugen für die Formel ψ in $\mathcal{A}_w$. Wir haben also auch die umgekehrte Implikation, womit die Äquivalenz

$$\mathcal{A}_w \models \psi \iff w \in L(M_n)$$

gezeigt ist. Außerdem hängt der Satz ψ nicht vom Eingabewort w ab.

Es ist leicht zu sehen, daß aus dem Wort w eine automatische Präsentation P'_w der automatischen Struktur $\mathcal{A}_w$ berechnet werden kann. Also ist die Abbildung $w \mapsto P'_w$ eine Reduktion des $n-\mathsf{EXPSPACE}$-vollständigen Problems $L(M_n)$ auf $\mathrm{MC}(\mathrm{AutPr}, \{\psi\})$.

Allerdings ist der Automat für L_n mehrfach exponentiell in $m = |w|$, unsere Reduktion ist also keine Polynomialzeitreduktion, es bleibt also noch Arbeit für die nächste Vorlesung!