Technische Universität Dresden

Formale Systeme

25. Vorlesung: Unentscheidbare Probleme formaler Sprachen

Markus Krötzsch

Professur für Wissensbasierte Systeme

TU Dresden, 26. Januar 2026

Rückblick

Unentscheidbare Probleme formaler Sprachen

Wiederholung

Wir schreiben $\mathbf{L}(G)$ für die Sprache, welche durch die Grammatik G erzeugt wird.

Satz (Vorlesung 5): Das Schnittproblem regulärer Grammatiken ist entscheidbar.
Gegeben: Reguläre Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) \cap \mathbf{L}(G_2) \neq \emptyset$?

Wiederholung

Wir schreiben $\mathbf{L}(G)$ für die Sprache, welche durch die Grammatik G erzeugt wird.

Satz (Vorlesung 5): Das Schnittproblem regulärer Grammatiken ist entscheidbar.
Gegeben: Reguläre Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) \cap \mathbf{L}(G_2) \neq \emptyset$?

Beweisskizze: Für reguläre Grammatiken G_1 und G_2 kann man $\mathbf{L}(G_1) \cap \mathbf{L}(G_2)$ durch einen Automaten darstellen (Produktkonstruktion). Automaten kann man leicht auf Leerheit testen. $\square$

Wiederholung

Wir schreiben $\mathbf{L}(G)$ für die Sprache, welche durch die Grammatik G erzeugt wird.

Satz (Vorlesung 5): Das Schnittproblem regulärer Grammatiken ist entscheidbar.
Gegeben: Reguläre Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) \cap \mathbf{L}(G_2) \neq \emptyset$?

Beweisskizze: Für reguläre Grammatiken G_1 und G_2 kann man $\mathbf{L}(G_1) \cap \mathbf{L}(G_2)$ durch einen Automaten darstellen (Produktkonstruktion). Automaten kann man leicht auf Leerheit testen. □

Satz (Vorlesung 14): Das Schnittproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) \cap \mathbf{L}(G_2) \neq \emptyset$?

CFG-Schnittproblem unentscheidbar (1)

Satz: Das Schnittproblem kontextfreier Grammatiken ist unentscheidbar.

Gegeben: Kontextfreie Grammatiken G_1 und G_2

Frage: Ist $\mathbf{L}(G_1) \cap \mathbf{L}(G_2) \neq \emptyset$?

CFG-Schnittproblem unentscheidbar (1)

Satz: Das Schnittproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) \cap \mathbf{L}(G_2) \neq \emptyset$?

Beweis: Durch Many-One-Reduktion vom PCP:

CFG-Schnittproblem unentscheidbar (1)

Satz: Das Schnittproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) \cap \mathbf{L}(G_2) \neq \emptyset$?

Beweis: Durch Many-One-Reduktion vom PCP:

- Für eine gegebene PCP-Instanz P
- konstruieren wir kontextfreie Grammatiken G_x und G_y,

  so dass gilt:
- P hat eine Lösung genau dann wenn $\mathbf{L}(G_x) \cap \mathbf{L}(G_y) \neq \emptyset$.

CFG-Schnittproblem unentscheidbar (2)

Beweis: Sei $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad \ldots \quad \begin{bmatrix} x_k \\ y_k \end{bmatrix}$ eine PCP-Instanz mit Alphabet Σ.

CFG-Schnittproblem unentscheidbar (2)

Beweis: Sei $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad \ldots \quad \begin{bmatrix} x_k \\ y_k \end{bmatrix}$ eine PCP-Instanz mit Alphabet Σ.

Die Grammatik G_x wird definiert als $\langle V, \Sigma_k, P_x, \mathsf{S}\rangle$ mit

- $V = \{\mathsf{S}\}$
- $\Sigma_k = \Sigma \cup \{1, \ldots, k\}$ (o.B.d.A. sei dies eine disjunkte Vereinigung)
- P_x ist die Menge aller Regeln

$$\mathsf{S} \to i\mathsf{S}x_i \quad \text{und} \quad \mathsf{S} \to ix_i \qquad \text{für alle } 1 \leq i \leq k$$

Damit ist G_x leicht berechenbar.

$G_y = \langle V, \Sigma_k, P_y, \mathsf{S}\rangle$ wird analog definiert.

CFG-Schnittproblem unentscheidbar (2)

Beweis: Sei $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad \ldots \quad \begin{bmatrix} x_k \\ y_k \end{bmatrix}$ eine PCP-Instanz mit Alphabet Σ.

Die Grammatik G_x wird definiert als $\langle V, \Sigma_k, P_x, \mathsf{S} \rangle$ mit

- $V = \{\mathsf{S}\}$
- $\Sigma_k = \Sigma \cup \{1, \ldots, k\}$ (o.B.d.A. sei dies eine disjunkte Vereinigung)
- P_x ist die Menge aller Regeln

$$\mathsf{S} \to i\mathsf{S}x_i \quad \text{und} \quad \mathsf{S} \to ix_i \qquad \text{für alle } 1 \leq i \leq k$$

Damit ist G_x leicht berechenbar.
$G_y = \langle V, \Sigma_k, P_y, \mathsf{S} \rangle$ wird analog definiert.

Damit ergibt sich:

- $\mathbf{L}(G_x) = \{i_\ell \cdots i_1 x_{i_1} \cdots x_{i_\ell} \mid \ell \geq 1 \text{ und } i_1, \ldots, i_\ell \in \{1, \ldots, k\}\}$ und
- $\mathbf{L}(G_y) = \{i_\ell \cdots i_1 y_{i_1} \cdots y_{i_\ell} \mid \ell \geq 1 \text{ und } i_1, \ldots, i_\ell \in \{1, \ldots, k\}\}$

CFG-Schnittproblem unentscheidbar (3)

Beweis: Wie soeben erkannt:

- $\mathbf{L}(G_x) = \{i_\ell \cdots i_1 x_{i_1} \cdots x_{i_\ell} \mid \ell \geq 1$ und $i_1, \ldots, i_\ell \in \{1, \ldots, k\}\}$ und
- $\mathbf{L}(G_y) = \{i_\ell \cdots i_1 y_{i_1} \cdots y_{i_\ell} \mid \ell \geq 1$ und $i_1, \ldots, i_\ell \in \{1, \ldots, k\}\}$

CFG-Schnittproblem unentscheidbar (3)

Beweis: Wie soeben erkannt:

- $\mathbf{L}(G_x) = \{i_\ell \cdots i_1 x_{i_1} \cdots x_{i_\ell} \mid \ell \geq 1$ und $i_1, \ldots, i_\ell \in \{1, \ldots, k\}\}$ und
- $\mathbf{L}(G_y) = \{i_\ell \cdots i_1 y_{i_1} \cdots y_{i_\ell} \mid \ell \geq 1$ und $i_1, \ldots, i_\ell \in \{1, \ldots, k\}\}$

Damit folgt:

$$\mathbf{L}(G_x) \cap \mathbf{L}(G_y) \neq \emptyset$$

CFG-Schnittproblem unentscheidbar (3)

Beweis: Wie soeben erkannt:

- $\mathbf{L}(G_x) = \{i_\ell \cdots i_1 x_{i_1} \cdots x_{i_\ell} \mid \ell \geq 1 \text{ und } i_1, \ldots, i_\ell \in \{1, \ldots, k\}\}$ und
- $\mathbf{L}(G_y) = \{i_\ell \cdots i_1 y_{i_1} \cdots y_{i_\ell} \mid \ell \geq 1 \text{ und } i_1, \ldots, i_\ell \in \{1, \ldots, k\}\}$

Damit folgt:

$$\mathbf{L}(G_x) \cap \mathbf{L}(G_y) \neq \emptyset$$

gdw. es gibt eine Sequenz $i_1, \ldots, i_\ell \in \{1, \ldots, k\}$ mit $\ell \geq 1$, so dass:

$$i_\ell \cdots i_1 x_{i_1} \cdots x_{i_\ell} = i_\ell \cdots i_1 y_{i_1} \cdots y_{i_\ell}$$

CFG-Schnittproblem unentscheidbar (3)

Beweis: Wie soeben erkannt:

- $\mathbf{L}(G_x) = \{i_\ell \cdots i_1 x_{i_1} \cdots x_{i_\ell} \mid \ell \geq 1$ und $i_1, \ldots, i_\ell \in \{1, \ldots, k\}\}$ und
- $\mathbf{L}(G_y) = \{i_\ell \cdots i_1 y_{i_1} \cdots y_{i_\ell} \mid \ell \geq 1$ und $i_1, \ldots, i_\ell \in \{1, \ldots, k\}\}$

Damit folgt:

$\mathbf{L}(G_x) \cap \mathbf{L}(G_y) \neq \emptyset$

gdw. es gibt eine Sequenz $i_1, \ldots, i_\ell \in \{1, \ldots, k\}$ mit $\ell \geq 1$, so dass:

$i_\ell \cdots i_1 x_{i_1} \cdots x_{i_\ell} = i_\ell \cdots i_1 y_{i_1} \cdots y_{i_\ell}$

gdw. es gibt eine Sequenz $i_1, \ldots, i_\ell \in \{1, \ldots, k\}$ mit $\ell \geq 1$, so dass:

$x_{i_1} \cdots x_{i_\ell} = y_{i_1} \cdots y_{i_\ell}$

CFG-Schnittproblem unentscheidbar (3)

Beweis: Wie soeben erkannt:

- $\mathbf{L}(G_x) = \{i_\ell \cdots i_1 x_{i_1} \cdots x_{i_\ell} \mid \ell \geq 1 \text{ und } i_1, \ldots, i_\ell \in \{1, \ldots, k\}\}$ und
- $\mathbf{L}(G_y) = \{i_\ell \cdots i_1 y_{i_1} \cdots y_{i_\ell} \mid \ell \geq 1 \text{ und } i_1, \ldots, i_\ell \in \{1, \ldots, k\}\}$

Damit folgt:

$$\mathbf{L}(G_x) \cap \mathbf{L}(G_y) \neq \emptyset$$

gdw. es gibt eine Sequenz $i_1, \ldots, i_\ell \in \{1, \ldots, k\}$ mit $\ell \geq 1$, so dass:

$$i_\ell \cdots i_1 x_{i_1} \cdots x_{i_\ell} = i_\ell \cdots i_1 y_{i_1} \cdots y_{i_\ell}$$

gdw. es gibt eine Sequenz $i_1, \ldots, i_\ell \in \{1, \ldots, k\}$ mit $\ell \geq 1$, so dass:

$$x_{i_1} \cdots x_{i_\ell} = y_{i_1} \cdots y_{i_\ell}$$

gdw. Die PCP-Instanz P hat eine Lösung □

Wiederholung

Wir wissen:

- Kontextfreie Grammatiken kann man als Kellerautomaten darstellen und umgekehrt (diese Umformung ist berechenbar)
- Deterministische kontextfreie Sprachen kann man als deterministische Kellerautomaten darstellen

Satz (Wiederholung):

- Das Leerheitsproblem für kontextfreie Grammatiken ist entscheidbar
- Kontextfreie Sprachen sind unter Vereinigung abgeschlossen
- Deterministische kontextfreie Sprachen sind unter Komplement abgeschlossen

Eine einfache Beobachtung

Die Grammatiken G_x und G_y aus dem vorigen Beweis kann man leicht als deterministische Kellerautomaten darstellen

Eine einfache Beobachtung

Die Grammatiken G_x und G_y aus dem vorigen Beweis kann man leicht als deterministische Kellerautomaten darstellen:

- Die Indices $i_\ell \cdots i_1$ lassen sich deterministisch einlesen und auf dem Stack ablegen
- Sobald der Wortteil $x_{i_1} \cdots x_{i_\ell}$ beginnt, wird der Stack abgearbeitet und jeweils nur das Wort für den aktuellen Index akzeptiert

Eine einfache Beobachtung

Die Grammatiken G_x und G_y aus dem vorigen Beweis kann man leicht als deterministische Kellerautomaten darstellen:

- Die Indices $i_\ell \cdots i_1$ lassen sich deterministisch einlesen und auf dem Stack ablegen
- Sobald der Wortteil $x_{i_1} \cdots x_{i_\ell}$ beginnt, wird der Stack abgearbeitet und jeweils nur das Wort für den aktuellen Index akzeptiert

Wir haben also auch schon gezeigt:

Korollar: Das Schnittproblem deterministischer Kellerautomaten ist unentscheidbar.
Gegeben: Deterministische Kellerautomaten $\mathcal{M}_1$ und $\mathcal{M}_2$
Frage: Ist $\mathbf{L}(\mathcal{M}_1) \cap \mathbf{L}(\mathcal{M}_2) \neq \emptyset$?

CFG-Äquivalenz (1)

Satz: Das Äquivalenzproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) = \mathbf{L}(G_2)$?

Beweis: Durch Many-One-Reduktion vom Komplement des Schnittproblems.

CFG-Äquivalenz (1)

Satz: Das Äquivalenzproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) = \mathbf{L}(G_2)$?

Beweis: Durch Many-One-Reduktion vom Komplement des Schnittproblems.

- Wir verwenden G_x und G_y aus dem vorigen Beweis

CFG-Äquivalenz (1)

Satz: Das Äquivalenzproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) = \mathbf{L}(G_2)$?

Beweis: Durch Many-One-Reduktion vom Komplement des Schnittproblems.

- Wir verwenden G_x und G_y aus dem vorigen Beweis
- Wir wissen, wie man einen det. Kellerautomaten $\mathcal{M}_x$ für G_x konstruiert

CFG-Äquivalenz (1)

Satz: Das Äquivalenzproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) = \mathbf{L}(G_2)$?

Beweis: Durch Many-One-Reduktion vom Komplement des Schnittproblems.

- Wir verwenden G_x und G_y aus dem vorigen Beweis
- Wir wissen, wie man einen det. Kellerautomaten $\mathcal{M}_x$ für G_x konstruiert
- $\mathcal{M}_x$ kann man komplementieren: sei $\overline{\mathcal{M}_x}$ der Automat für die Sprache $\overline{\mathbf{L}(\mathcal{M}_x)}$

CFG-Äquivalenz (1)

Satz: Das Äquivalenzproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) = \mathbf{L}(G_2)$?

Beweis: Durch Many-One-Reduktion vom Komplement des Schnittproblems.

- Wir verwenden G_x und G_y aus dem vorigen Beweis
- Wir wissen, wie man einen det. Kellerautomaten $\mathcal{M}_x$ für G_x konstruiert
- $\mathcal{M}_x$ kann man komplementieren: sei $\overline{\mathcal{M}}_x$ der Automat für die Sprache $\overline{\mathbf{L}(\mathcal{M}_x)}$
- Für $\overline{\mathcal{M}}_x$ kann man eine Grammatik berechnen: sei $\overline{G}_x$ die Grammatik für die Sprache $\mathbf{L}(\overline{\mathcal{M}}_x)$

CFG-Äquivalenz (1)

Satz: Das Äquivalenzproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) = \mathbf{L}(G_2)$?

Beweis: Durch Many-One-Reduktion vom Komplement des Schnittproblems.

- Wir verwenden G_x und G_y aus dem vorigen Beweis
- Wir wissen, wie man einen det. Kellerautomaten $\mathcal{M}_x$ für G_x konstruiert
- $\mathcal{M}_x$ kann man komplementieren: sei $\overline{\mathcal{M}}_x$ der Automat für die Sprache $\overline{\mathbf{L}(\mathcal{M}_x)}$
- Für $\overline{\mathcal{M}}_x$ kann man eine Grammatik berechnen: sei $\overline{G}_x$ die Grammatik für die Sprache $\mathbf{L}(\overline{\mathcal{M}}_x)$
- Kontextfreie Grammatiken kann man vereinigen: sei $G_{\overline{x}y}$ die Grammatik mit $\mathbf{L}(G_{\overline{x}y}) = \mathbf{L}(\overline{G}_x) \cup \mathbf{L}(G_y)$

CFG-Äquivalenz (2)

Satz: Das Äquivalenzproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) = \mathbf{L}(G_2)$?

Beweis: Wir behaupten:

$$„\mathbf{L}(G_x) \cap \mathbf{L}(G_y) \stackrel{?}{=} \emptyset“ \mapsto „\mathbf{L}(G_{\overline{x}y}) \stackrel{?}{=} \mathbf{L}(\overline{G}_x)“$$

ist die gesuchte Reduktion.

CFG-Äquivalenz (2)

Satz: Das Äquivalenzproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) = \mathbf{L}(G_2)$?

Beweis: Wir behaupten:

$$„\mathbf{L}(G_x) \cap \mathbf{L}(G_y) \stackrel{?}{=} \emptyset" \mapsto „\mathbf{L}(G_{\overline{x}y}) \stackrel{?}{=} \mathbf{L}(\overline{G}_x)"$$

ist die gesuchte Reduktion.

$$\mathbf{L}(G_x) \cap \mathbf{L}(G_y) = \emptyset$$

CFG-Äquivalenz (2)

Satz: Das Äquivalenzproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) = \mathbf{L}(G_2)$?

Beweis: Wir behaupten:

$$„\mathbf{L}(G_x) \cap \mathbf{L}(G_y) \stackrel{?}{=} \emptyset“ \mapsto „\mathbf{L}(G_{\overline{x}y}) \stackrel{?}{=} \mathbf{L}(\overline{G}_x)“$$

ist die gesuchte Reduktion.

$$\mathbf{L}(G_x) \cap \mathbf{L}(G_y) = \emptyset \quad \text{gdw.} \quad \mathbf{L}(G_y) \subseteq \mathbf{L}(\overline{G}_x)$$

CFG-Äquivalenz (2)

Satz: Das Äquivalenzproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) = \mathbf{L}(G_2)$?

Beweis: Wir behaupten:

$$„\mathbf{L}(G_x) \cap \mathbf{L}(G_y) \stackrel{?}{=} \emptyset\text{“} \mapsto „\mathbf{L}(G_{\overline{x}y}) \stackrel{?}{=} \mathbf{L}(\overline{G}_x)\text{“}$$

ist die gesuchte Reduktion.

$$\begin{aligned} \mathbf{L}(G_x) \cap \mathbf{L}(G_y) = \emptyset \quad &\text{gdw.} \quad \mathbf{L}(G_y) \subseteq \mathbf{L}(\overline{G}_x) \\ &\text{gdw.} \quad \mathbf{L}(G_y) \cup \mathbf{L}(\overline{G}_x) = \mathbf{L}(\overline{G}_x) \end{aligned}$$

CFG-Äquivalenz (2)

Satz: Das Äquivalenzproblem kontextfreier Grammatiken ist unentscheidbar.
Gegeben: Kontextfreie Grammatiken G_1 und G_2
Frage: Ist $\mathbf{L}(G_1) = \mathbf{L}(G_2)$?

Beweis: Wir behaupten:

$$„\mathbf{L}(G_x) \cap \mathbf{L}(G_y) \stackrel{?}{=} \emptyset“ \mapsto „\mathbf{L}(G_{\overline{x}y}) \stackrel{?}{=} \mathbf{L}(\overline{G}_x)“$$

ist die gesuchte Reduktion.

$$\begin{aligned} \mathbf{L}(G_x) \cap \mathbf{L}(G_y) = \emptyset \quad & \text{gdw.} \quad \mathbf{L}(G_y) \subseteq \mathbf{L}(\overline{G}_x) \\ & \text{gdw.} \quad \mathbf{L}(G_y) \cup \mathbf{L}(\overline{G}_x) = \mathbf{L}(\overline{G}_x) \\ & \text{gdw.} \quad \mathbf{L}(G_{\overline{x}y}) = \mathbf{L}(\overline{G}_x) \end{aligned}$$

Die Behauptung folgt, da das Komplement des Schnittproblems unentscheidbar ist. □

Diskussion

Anmerkung 1: $G_{\overline{x}y}$ ist nicht unbedingt deterministisch. Der Beweis gilt also nicht für deterministische CFGs. In der Tat ist Äquivalenz dort (mit viel Aufwand) entscheidbar.

(Sénizergues: L(A)=L(B)? decidability results from complete formal systems, 2001; der komplexe Beweis zeigt Semi-Entscheidbarkeit des Problems und seines Komplements, also keine Zeitgrenzen!)

Diskussion

Anmerkung 1: $G_{\overline{x}y}$ ist nicht unbedingt deterministisch. Der Beweis gilt also nicht für deterministische CFGs. In der Tat ist Äquivalenz dort (mit viel Aufwand) entscheidbar.

(Sénizergues: L(A)=L(B)? decidability results from complete formal systems, 2001; der komplexe Beweis zeigt Semi-Entscheidbarkeit des Problems und seines Komplements, also keine Zeitgrenzen!)

Anmerkung 2: Aus der Unentscheidbarkeit der CFG-Äquivalenz folgt – durch einfache Many-One-Reduktion – die Unentscheidbarkeit der Äquivalenz aller Formalismen, in die man CFGs leicht übersetzen kann:

- Kellerautomaten
- kontextsensitive Grammatiken/LBAs
- LOOP-Programme
- Typ-0-Grammatiken/Turingmaschinen/WHILE-Programme
- ...

Unentscheidbare Probleme für Typ 1

Wir halten noch einmal fest:

Satz: Für kontextsensitive Grammatiken G_1 und G_2 sind die folgenden Fragen unentscheidbar:

(1) Äquivalenz: $\mathbf{L}(G_1) = \mathbf{L}(G_2)$?

(2) Schnitt: $\mathbf{L}(G_1) \cap \mathbf{L}(G_2) = \emptyset$?

(3) Leerheit: $\mathbf{L}(G_1) = \emptyset$?

Beweis: (1) und (2) gelten, weil alle kontextfreien Grammatiken auch kontextsensitive Grammatiken sind (offensichtliche Many-One-Reduktion).

(3) gilt, da kontextsensitive Grammatiken unter Schnitten abgeschlossen sind (siehe Vorlesung 24), so dass man Schnitt auf Leerheit reduzieren kann. □

Semi-Entscheidbarkeit

Beobachtung 1: Das Schnittproblem ist semi-entscheidbar: zähle alle Wörter von $\mathbf{L}(G_1)$ auf und teste jeweils, ob sie in $\mathbf{L}(G_2)$ liegen.

Semi-Entscheidbarkeit

Beobachtung 1: Das Schnittproblem ist semi-entscheidbar: zähle alle Wörter von $\mathbf{L}(G_1)$ auf und teste jeweils, ob sie in $\mathbf{L}(G_2)$ liegen.

Beobachtung 2: Das Komplement des Schnittproblems ist demnach nicht semi-entscheidbar. Ebenso ist also das Äquivalenzproblem nicht semi-entscheidbar (wegen Many-One-Reduktion).

Semi-Entscheidbarkeit

Beobachtung 1: Das Schnittproblem ist semi-entscheidbar: zähle alle Wörter von $\mathbf{L}(G_1)$ auf und teste jeweils, ob sie in $\mathbf{L}(G_2)$ liegen.

Beobachtung 2: Das Komplement des Schnittproblems ist demnach nicht semi-entscheidbar. Ebenso ist also das Äquivalenzproblem nicht semi-entscheidbar (wegen Many-One-Reduktion).

Beobachtung 3: Das Komplement des Äquivalenzproblems ist semi-entscheidbar: zähle abwechselnd Wörter von $\mathbf{L}(G_1)$ und $\mathbf{L}(G_2)$ auf und teste jeweils, ob sie nicht in $\mathbf{L}(G_2)$ bzw. $\mathbf{L}(G_1)$ liegen.

Unentscheidbarkeiten

Das schwerste unentscheidbare Problem?

Wir haben gesehen (Übung):

Satz: Jedes semi-entscheidbare Problem kann auf das Halteproblem many-one-reduziert werden.

Das schwerste unentscheidbare Problem?

Wir haben gesehen (Übung):

Satz: Jedes semi-entscheidbare Problem kann auf das Halteproblem many-one-reduziert werden.

Demnach kann man außerdem Komplemente semi-entscheidbarer Probleme („co-semi-entscheidbare“ Probleme) auf das Halteproblem Turing-reduzieren.

Mit anderen Worten: Wenn man das Halteproblem lösen könnte, dann könnte man jedes (co-)semi-entscheidbare Problem lösen.

Das schwerste unentscheidbare Problem?

Wir haben gesehen (Übung):

Satz: Jedes semi-entscheidbare Problem kann auf das Halteproblem many-one-reduziert werden.

Demnach kann man außerdem Komplemente semi-entscheidbarer Probleme („co-semi-entscheidbare“ Probleme) auf das Halteproblem Turing-reduzieren.

Mit anderen Worten: Wenn man das Halteproblem lösen könnte, dann könnte man jedes (co-)semi-entscheidbare Problem lösen.

Ist das Halteproblem das schwerste unentscheidbare Problem?

(Sind alle unentscheidbaren Probleme auf das Halteproblem Turing-reduzierbar?)

Das schwerste unentscheidbare Problem?

Ist das Halteproblem das schwerste unentscheidbare Problem?

(Sind alle unentscheidbaren Probleme auf das Halteproblem Turing-reduzierbar?)

Das schwerste unentscheidbare Problem?

Ist das Halteproblem das schwerste unentscheidbare Problem?

(Sind alle unentscheidbaren Probleme auf das Halteproblem Turing-reduzierbar?)

Nein, sicher nicht.

Das schwerste unentscheidbare Problem?

Ist das Halteproblem das schwerste unentscheidbare Problem?

(Sind alle unentscheidbaren Probleme auf das Halteproblem Turing-reduzierbar?)

Nein, sicher nicht.

Beweisskizze: Wir können uns Turing-Reduktionen als TMs vorstellen, die Subroutinen aufrufen dürfen.

- Selbst ohne die Details der formalen Definition ist klar: Solche TMs müssen weiterhin endlich beschreibbar sein.
- Daher gibt es nur abzählbar viele solcher TMs.
- Es gibt aber überabzählbar viele Probleme.

Also sind die meisten Probleme nicht durch Turing-Reduktionen auf das Halteproblem lösbar. □

Noch unentscheidbarere Probleme

Gibt es auch konkrete unentscheidbare Probleme, die nicht mithilfe von $\mathbf{P}_{\text{halt}}$ lösbar sind?

Noch unentscheidbarere Probleme

Gibt es auch konkrete unentscheidbare Probleme, die nicht mithilfe von $\mathbf{P}_{\text{halt}}$ lösbar sind?

Ja, zum Beispiel folgendes:

Wir betrachten folgendes Problem $\mathbf{P}^2_{\text{halt}}$:
Gegeben: ein Wort w und eine DTM $\mathcal{M}$, welche $\mathbf{P}_{\text{halt}}$ als Subroutine verwenden darf
Frage: Hält $\mathcal{M}$ auf w?

Dies ist sozusagen ein Haltproblem höherer Ordnung.

Noch unentscheidbarere Probleme

Gibt es auch konkrete unentscheidbare Probleme, die nicht mithilfe von $\mathbf{P}_{\text{halt}}$ lösbar sind?

Ja, zum Beispiel folgendes:

Wir betrachten folgendes Problem $\mathbf{P}^2_{\text{halt}}$:
Gegeben: ein Wort w und eine DTM $\mathcal{M}$, welche $\mathbf{P}_{\text{halt}}$ als Subroutine verwenden darf
Frage: Hält $\mathcal{M}$ auf w?

Dies ist sozusagen ein Haltproblem höherer Ordnung.

Ein noch schwereres Problem $\mathbf{P}^3_{\text{halt}}$ ist das Halteproblem für TMs, die $\mathbf{P}^2_{\text{halt}}$ als Subroutine verwenden dürfen
$\leadsto$ eine unendliche Hierarchie unentscheidbarer Probleme

Noch unentscheidbarere Probleme

Gibt es auch konkrete unentscheidbare Probleme, die nicht mithilfe von $\mathbf{P}_{\text{halt}}$ lösbar sind?

Ja, zum Beispiel folgendes:

Wir betrachten folgendes Problem $\mathbf{P}^2_{\text{halt}}$:
Gegeben: ein Wort w und eine DTM $\mathcal{M}$, welche $\mathbf{P}_{\text{halt}}$ als Subroutine verwenden darf
Frage: Hält $\mathcal{M}$ auf w?

Dies ist sozusagen ein Haltproblem höherer Ordnung.

Ein noch schwereres Problem $\mathbf{P}^3_{\text{halt}}$ ist das Halteproblem für TMs, die $\mathbf{P}^2_{\text{halt}}$ als Subroutine verwenden dürfen
$\leadsto$ eine unendliche Hierarchie unentscheidbarer Probleme

Und selbst all diese Probleme sind nur abzählbar viele ...

Das leichteste unentscheidbare Problem?

Ist das Halteproblem das leichteste unentscheidbare Problem?

(Ist das Halteproblem auf alle unentscheidbaren Probleme Turing-reduzierbar?)

Das leichteste unentscheidbare Problem?

Ist das Halteproblem das leichteste unentscheidbare Problem?

(Ist das Halteproblem auf alle unentscheidbaren Probleme Turing-reduzierbar?)

Nein, auch das gilt nicht.

Die Situation ist ziemlich kompliziert:

- Es gibt unentscheibare Probleme **A** und **B**, so dass
- **A** $\leq_T$ $\mathbf{P}_{\mathsf{halt}}$ und **B** $\leq_T$ $\mathbf{P}_{\mathsf{halt}}$, aber
- **A** $\nleq_T$ **B** und **B** $\nleq_T$ **A**

Man kann also mit $\leq_T$ nicht einmal alle Klassen unentscheidbarer Probleme in eine totale Ordnung bringen.

Bewiesen in 1956 (unabhängig!) von Friedberg (USA) und Muchnik (USSR)

Allerdings sind diese Probleme sehr künstlich.

Wozu das alles?

Die Untersuchung der Struktur des Unentscheidbaren hat sehr viele Fragen betrachtet und beantwortet.

$\leadsto$ Forschungsgegenstand der Berechenbarkeitstheorie

Wozu das alles?

Die Untersuchung der Struktur des Unentscheidbaren hat sehr viele Fragen betrachtet und beantwortet.

$\leadsto$ Forschungsgegenstand der Berechenbarkeitstheorie

Offensichtliche Frage: Bringt uns das praktische Einsichten?

Wozu das alles?

Die Untersuchung der Struktur des Unentscheidbaren hat sehr viele Fragen betrachtet und beantwortet.
$\leadsto$ Forschungsgegenstand der Berechenbarkeitstheorie

Offensichtliche Frage: Bringt uns das praktische Einsichten?

„Jain“:

- Einerseits sind alle unentscheidbaren Probleme praktisch unlösbar
- Andererseits kann der Grad der Unentscheidbarkeit ein Hinweis auf die Schwere entscheidbarer Teilprobleme sein

Beispiel: Noch ein Problem

Das Universalitätsproblem von TMs fragt, ob eine TM alle Eingaben akzeptiert:

Gegeben: Turingmaschine $\mathcal{M}$ über Eingabealphabet Σ

Frage: Ist $\mathbf{L}(\mathcal{M}) = \Sigma^*$?

Beispiel: Noch ein Problem

Das Universalitätsproblem von TMs fragt, ob eine TM alle Eingaben akzeptiert:
Gegeben: Turingmaschine $\mathcal{M}$ über Eingabealphabet Σ
Frage: Ist $\mathbf{L}(\mathcal{M}) = \Sigma^*$?

Das Universalitätsproblem vom TMs ist schwerer als das Halteproblem (aber Turing-reduzierbar auf $\mathbf{P}^2_{\text{halt}}$). Das zeigt sich auch bei Sonderfällen:

- Kontextfreie Grammatiken: Wortproblem und Leerheitsproblem entscheidbar; Universalität unentscheidbar
- Endliche Automaten: Wortproblem und Leerheitsproblem effizient lösbar (polynomiell); Universalität PSpace-schwer (nur exponentielle Algorithmen bekannt)

Zusammenfassung und Ausblick

Die Unentscheidbarkeit vieler Probleme der Sprachtheorie lässt sich gut durch Reduktion vom Postschen Korrespondenzproblem zeigen

Es gibt mehr als eine Art von Unentscheidbarkeit

Was erwartet uns als nächstes?

- Euklid als Informatiker
- Abschließende Bemerkungen und Zusammenfassung
- Prüfungsvorbereitung und Prüfung