Proseminar Theoretische Informatik 15.12.15

Ausgewählte unentscheidbare Sprachen

Marian Sigler, Jakob Köhler *Wolfgang Mulzer*

1 Entscheidbarkeit und Semi-Entscheidbarkeit

Definition 1: L ist entscheidbar $\Leftrightarrow$ es gibt eine TM M, so dass für alle $w \in \Sigma^*$ gilt:

$w \in L \Rightarrow$ M erreicht q_{ja} bei Eingabe w
$w \notin L \Rightarrow$ M erreicht q_{nein} bei Eingabe w

d.h. M hält bei jeder Eingabe

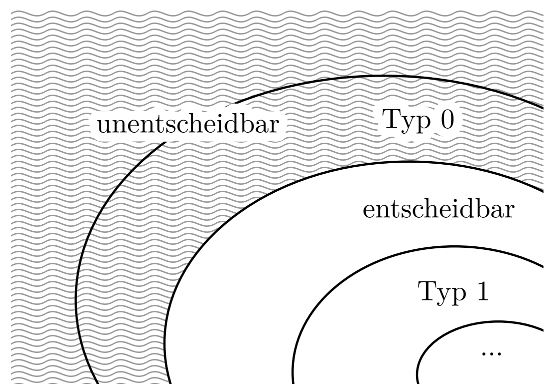
Definition 2: L ist semi-entscheidbar $\Leftrightarrow$ es gibt eine TM M, so dass für alle $w \in \Sigma^*$ gilt:

$w \in L \Rightarrow$ M erreicht q_{ja} bei Eingabe w
$w \notin L \Rightarrow$ M erreicht q_{nein} oder hält nicht bei Eingabe w

d.h. M hält nur zwangsläufig bei Eingaben $w \in$ L

Chomsky-Hierarchie

- Die semi-entscheidbaren Sprachen entsprechen in der Chomsky-Hierarchie den Typ-0 Sprachen
- Entscheidbare Sprachen sind echte Teilmenge der semi-entscheidbaren Sprachen (H, PCP)
- Entscheidbare Sprachen sind echte Obermenge der Typ-1 Sprachen



2 Reduktionen

Ein Reduktionsbeweis dient allgemein dazu, zu beweisen, dass ein Problem mindestens genauso schwer ist wie ein anderes. Im Folgenden soll mit dieser Beweistechnik die Unentscheidbarkeit verschiedener Probleme gezeigt werden, wir definieren es daher hier nur für diesen Spezialfall.

Definition 3: Reduktion

Sei P ein unentscheidbares Problem, Q ein weiteres Problem. Sei f eine (effektiv berechenbare und totale) Funktion, die eine Eingabe p des Problems P in eine Eingabe des Problems Q „umwandelt“: $q := f(p)$.

Wenn nun die Antwort auf eine Eingabe p wahr ist, genau dann wenn die Antwort auf das entsprechende q wahr ist, haben wir eine Möglichkeit gefunden, das Problem P zu lösen, nämlich indem wir seine Eingaben mittels f auf Q abbilden.

Man sagt dann, P *ist reduzierbar auf* Q, kurz: $P \leq Q$.

Lemma 4: Sei P unentscheidbar, und $P \leq Q$. Dann ist auch Q unentscheidbar.

Beweis: Wäre Q entscheidbar, gäbe es eine einfachere Möglichkeit, P zu berechnen, nämlich die obige Abbildung. Damit wäre P entscheidbar, was ein Widerspruch ist.

3 Das Postsche Korrespondenzproblem

Definition 5: PCP

gegeben: Eine endliche Folge von Wortpaaren $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ mit $x_i, y_i \in \Sigma^+$

gefragt: Gibt es eine Folge von Indizes $i_1, i_2, \ldots, i_n \in \{1, 2, \ldots, k\}, n \geq 0$, so dass $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

Dann heißt $i_1, i_2, \ldots, i_n$ Lösung des PCP und $x_{i_1} x_{i_2} \ldots x_{i_n}$ Lösungswort.

Semi-Entscheidbarkeit des PCP

Sei M eine TM, die für immer länger werdende Index-Folgen überprüft, ob sie Lösungen des PCP sind. Dann hält und akzeptiert M nach endlich vielen Schritten, wenn das PCP Lösungen hat. Wenn es keine Lösung gibt, hält M nicht.

Skizze für den Beweis der Unentscheidbarkeit des PCP

Definition 6: Halteproblem: H = {(<M>, w) | M hält bei Eingabe w}

Satz 7: Das Halteproblem H ist semi-entscheidbar, aber nicht entscheidbar.

Satz 8: Das PCP ist nicht entscheidbar

zu zeigen: H $\leq$ MPCP $\leq$ PCP

Aus der Unentscheidbarkeit des Halteproblems folgt dann die Unentscheidbarkeit des PCP

Das Modifizierte Postsche Korrespondenzproblem

Definition 9: MPCP

gegeben: wie beim PCP

gefragt: Gibt es eine Folge von Indizes wie beim PCP, aber mit $i_1 = 1$?

Lemma 10: Es gilt: MPCP $\leq$ PCP *(ohne Beweis)*

Reduktion des Halteproblems auf das MPCP

Lemma 11: Es gilt: H $\leq$ MPCP

Beweis: Sei M eine Turingmaschine mit M $= \{Z, \Sigma, \Gamma, \delta, z_0, \Box, E\}$ und $w \in \Sigma^*$ ein Eingabewort.

Wir konstruieren nun eine Funktion f, die ein solches Paar (M,w) in eine Eingabe $(x_1, y_1), \ldots, (x_k, y_k)$ für das MPCP überführt. Dabei soll gelten:

$$\text{M hält auf } w \quad \Leftrightarrow \quad (x_1, y_1), \ldots, (x_k, y_k) \text{ hat eine Lösung mit } i_1 = 1 \text{ (Reduktion)}$$

Die Funktion f verwendet dabei die folgenden Regeln, um in Abhängigkeit von M und w eine Eingabe für das MPCP zu definieren (Alphabet des MPCP ist $\Gamma \cup Z \cup \{\#\}$):

0. Das erste Wortpaar ist $(\#, \#z_0w\#)$
1. Kopierregeln: (a, a) für alle $a \in \Gamma \cup \{\#\}$
2. Überführungsregeln:
   $(za, z'c)$ falls $\delta(z, a) = (z', c, N)$
   (za, cz') falls $\delta(z, a) = (z', c, R)$
   $(bza, z'bc)$ falls $\delta(z, a) = (z', c, L)$ für alle $b \in \Gamma$
   $(\#za, \#z'\Box c)$ falls $\delta(z, a) = (z', c, L)$
   $(z\#, z'c\#)$ falls $\delta(z, \Box) = (z', c, N)$
   $(z\#, cz'\#)$ falls $\delta(z, \Box) = (z', c, R)$
   $(bz\#, z'bc\#)$ falls $\delta(z, \Box) = (z', c, L)$ für alle $b \in \Gamma$
3. Löschregeln: (az_e, z_e) und (z_ea, z_e) für alle $a \in \Gamma$ und $z_e \in E$
4. Abschlussregeln: $(z_e\#\#, \#)$ für alle $z_e \in E$

Die Eingabe für das MPCP hat genau dann eine Lösung, wenn M auf w hält. Das dazugehörige Lösungswort hat dann die Form

$$\#k_0\#k_1\#\ldots\#k_t\#k_t'\#k_t''\#\ldots\#z_e\#\#$$

wobei k_i Konfigurationen von M sind, k_t eine Endkonfiguration im Zustand z_e und k_t', $k_t''\ldots$ nacheinander aus k_t entstehen, indem die Nachbarsymbole von z_e gelöscht werden. $\Box$

Lemma 12: Das PCP ist auch mit dem Alphabet $\Sigma = \{0, 1\}$ unentscheidbar. *(Ohne Beweis.)*

4 Unentscheidbare Grammatik-Probleme

Ausgehend von der Unentscheidbarkeit des Postschen Korrespondenzproblems können wir beweisen, dass einige Grammatik-Probleme ebenfalls unentscheidbar sind.

Definition 13: Das **Schnittproblem** zweier Sprachen (definiert durch ihre Grammatiken) ist die Frage, ob es ein Wort gibt, das in beiden Sprachen enthalten ist; anders formuliert, dass der Schnitt dieser Sprachen nicht leer ist.

Satz 14: Das Schnittproblem zweier kontextfreier Sprachen ist unentscheidbar.

Beweis: Wir werden im Folgenden das Postsche Korrespondenzproblem (PCP) auf das Schnittproblem reduzieren. Damit ist nach Lemma 4 die Unentscheidbarkeit gezeigt.

Sei ein PCP gegeben: $((x_1, y_1), \ldots, (x_n, y_n))$ über dem Alphabet $0, 1$.

Wir konstruieren ein zu diesem PCP passendes Paar von Grammatiken G_1, G_2. Ziel ist es, dass nur genau die Worte in den Sprachen beider Grammatiken enthalten sind, die einer Lösung des PCP entsprechen.

Beide Grammatiken haben als Terminalsymbole $\Sigma = 0, 1, \$, a_1, \ldots, a_n$.

G_1 hat folgende Regeln:

$$\begin{aligned}
S &\rightarrow A \; \$ \; B \\
A &\rightarrow a_1 \; A \; x_1 \mid \ldots \mid a_n \; A \; x_n \\
A &\rightarrow a_1 \; x_1 \mid \ldots \mid a_n \; x_n \\
B &\rightarrow \widetilde{y_1} \; B \; a_1 \mid \ldots \mid \widetilde{y_n} \; B \; a_n \\
B &\rightarrow \widetilde{y_1} \; a_1 \mid \ldots \mid \widetilde{y_n} \; a_n
\end{aligned}$$

dabei sind die a_i Terminalsymbole, also unverändert Teil der Grammatik, während die x_i und y_i durch die Worte aus dem gegebenen PCP ersetzt werden. $\widetilde{y_i}$ ist y_i rückwärts gelesen.

G_2 hat folgende Regeln:

$$\begin{aligned}
S &\rightarrow a_1 \; S \; a_1 \mid \ldots \mid a_n \; S \; a_n \mid T \\
T &\rightarrow 0 \; T \; 0 \mid 1 \; T \; 1 \mid \$
\end{aligned}$$

alle Wörter in $L(G_2)$ sind also Palindrome (neben weiteren Bedingungen).

Die Wörter aus $L(G_1)$ bestehen vor dem \$ aus a_i und den ihnen entsprechenden x_i aus dem gegebenen PCP; nach dem \$ folgen beliebige a_i und die entsprechenden y_i (jedoch gespiegelt). Der Schnitt mit $L(G_2)$ enthält nur noch solche Wörter, bei denen a) in beiden Seiten die selbe Folge von a_i steht und b) die Konkatenation aller x_i das selbe Wort ist wie die Konkatenation der y_i.

Der Schnitt $L(G_1) \cap L(G_2)$ ist also genau dann nicht leer, wenn es Lösungen für das gegebene PCP gibt. Damit haben wir das PCP auf das Schnittproblem reduziert, das folglich unentscheidbar ist. □

Satz 15: Die Frage, ob die Schnittmenge der Sprachen zweier kontextfreier Grammatiken unendlich groß ist (also $|L(G_1) \cap L(G_2)| = \infty$), ist unentscheidbar.

Beweis: Wir stellen zunächst fest: Ist ein Wort w eine Lösung eines PCP, so sind es auch beliebige Wiederholungen dieses Wortes. Das heißt: hat ein PCP eine Lösung, so hat es unendlich viele. Die Frage, ob ein PCP unendlich viele Lösungen hat, ist also unentscheidbar.

Wir betrachten wieder die Reduktion aus obigem Beweis. Sie ist ebenfalls eine Reduktion der Frage, ob ein PCP unendlich viele Lösungen hat, auf die Frage, ob die Vereinigung beider Sprachen unendlich viele Wörter enthält. Letztere Frage ist also auch unentscheidbar.

Satz 16: Die Frage, ob der Schnitt zweier kontextfreier Sprachen (definiert durch ihre Grammatiken) ebenfalls kontextfrei ist, ist unentscheidbar.

Beweis: Der Schnitt ist, wenn er Worte enthält, nicht kontextfrei (Beweis s. u.). Wir haben schon bewiesen, dass es unentscheidbar ist, ob der Schnitt leer ist (dann wäre er kontextfrei). Folglich ist es unentscheidbar, ob er kontextfrei ist.

Beweis, dass der Schnitt nicht kontextfrei ist: Sei L_S dieser nicht-leere Schnitt. Wir betrachten wieder obige Reduktion, und nehmen an, der Schnitt sei kontextfrei. Dann könnten wir laut dem Pumping-Lemma für kontextfreie Sprachen zwei Teilworte v, x eines Wortes w aus L_S pumpen, und das Ergebnis w' müsste immer noch in L_S sein. Das ist unmöglich, denn:

- enthält eines der Teilworte \$, enthielte w' mehrere \$.
- enthält eines der Teilworte sowohl Zeichen aus dem a_i-Block des Wortes als auch aus dem durch die x_i bzw. y_i erzeugten, nur 0 und 1 enthaltenden Block, hat das gepumpte Wort w' nicht mehr die von G_2 verlangte Form $(\{a_i\}^*\{0,1\}^*\$\{0,1\}^*\{a_i\}^*)$ und ist damit auch nicht in L_S.
- ansonsten besteht eines der Teilworte nur aus dem a_i-Block oder nur aus dem $0,1$-Block. Pumpt man es, passen diese Blöcke nicht mehr zueinander, w' hat also nicht mehr die vom G_1 geforderte Form und ist damit auch nicht in L_S.

(Da wir durch Wiederholung einer Lösung beliebig lange Wörter erzeugen können, lässt sich dieser Widerspruch für jede gegebene Pumping-Zahl zeigen.) $\square$

Literatur

- Uwe Schöning. *Theoretische Informatik – kurz gefasst.*
  Spektrum Akademischer Verlag, 5. Auflage, 2008.
- Katrin Erk, Lutz Priese. *Theoretische Informatik. Eine umfassende Einführung.*
  Springer-Verlag, 3. Auflage, 2008.