Unentscheidbarkeit des Eindeutigkeitsproblems für kontextfreie Grammatiken

Niklas Rieken

26. Juni 2018

In der Vorlesung wurde erwähnt, dass es unentscheidbar (d.h. nicht auf algorithmischen Wege lösbar) ist zu einer gegebenen kontextfreien Grammatik (*CFG*) zubestimmen ob diese eindeutig (*unambiguous*) oder mehrdeutig (*ambiguous*) ist. In diesem Dokument liefern wir einen Beweis dazu nach, der zwar bereits einige Konzepte und Sätze aus der Vorlesung *Berechenbarkeit und Komplexität* verwendet, aber konzeptionell trotzdem gut verständlich sein sollte. Wir beginnen mit ein paar Definitionen.

Definition 1. Sei $\mathcal{G} = (N, \Sigma, P, S)$ eine kontextfreie Grammatik. Eine Ableitung in $\mathcal{G}$ heißt *Linksableitung*, wenn in jedem Ableitungsschritt das Nichtterminal ersetzt wird, welches kein Nichtterminal links von sich stehen hat.

Definition 2. Sei $\mathcal{G} = (N, \Sigma, P, S)$ eine kontextfreie Grammatik. $\mathcal{G}$ heißt *eindeutig*, falls für alle $w \in L(\mathcal{G})$ genau eine Linksableitung existiert. Ansonsten heißt $\mathcal{G}$ *mehrdeutig*.

Wir wollen nun zeigen, dass das folgende Problem unentscheidbar ist:
Unambiguous Grammar (UG). Gegeben kontextfreie Grammatik $\mathcal{G}$. Ist $\mathcal{G}$ eindeutig?
Wir verwenden dazu das Hilfsproblem
Ambiguous Grammar (AG). Gegeben kontextfreie Grammatik $\mathcal{G}$. Ist $\mathcal{G}$ mehrdeutig?
Es ist klar, dass wenn AG unentscheidbar ist, dass dann auch UG unentscheidbar ist, denn wenn wir einen einen Algorithmus A für AG hätten so würde ein Algorithmus für UG einfach nur A auf der eingegebenen Grammatik aufrufen und den Output umdrehen.
Wie zeigen wir aber nun, dass AG unentscheidbar ist. Dazu nehmen wir uns das folgende Problem, von dem wir wissen, dass es unentscheidbar ist:
Post's Correspondence Problem (PCP). Gegeben eine Menge von Dominos

$$\mathcal{D} \coloneqq \left\{ \left[\frac{u_1}{v_1}\right], \ldots, \left[\frac{u_n}{v_n}\right] \right\}$$

mit $u_i, v_i \in \Sigma^*$ für ein Alphabet Σ. Gibt es eine Indexfolge $(i_1, \ldots, i_k)$, sodass $u_{i_1} \ldots u_{i_k} = v_{i_1} \ldots v_{i_k}$? (D.h. gesucht ist eine Aneinanderreihung von Dominos (mit Wiederholungen) aus $\mathcal{D}$, sodass oben das selbe Wort wie unten steht).
Dass PCP unentscheidbar ist, ist ein Satz aus Berechenbarkeit und Komplexität, den wir hier nicht beweisen. Das ist vermutlich etwas unbefriedigend, da das Problem eigentlich

nicht sonderlich schwer aussieht und nun doch die Aussage im Raum steht, dass es keinen Algorithmus dafür gibt. Ein Beweis dieser Aussage würde jedoch extrem weit vorgreifen, nur so viel: Man kann zeigen, dass, wenn PCP entscheidbar wäre auch das *Halteproblem für Turingmaschinen*, von dem man in populärwissenschaftlichen Quellen (z.B. Numberphile) gehört haben könnte, ebenfalls entscheidbar wäre.
Wir führen nun das Konzept einer *Reduktion* ein.

Definition 3. Seien $A, B \subseteq \Sigma^*$. A heißt *many-one-reduzierbar* auf B (geschrieben $A \leq_m B$), falls eine berechenbare Funktion $f\colon \Sigma^* \to \Sigma^*$ existiert mit $x \in A$ g.d.w. $f(x) \in B$. Man nennt f dann auch *Reduktionsabbildung*.

Was bedeutet das intuitiv: Zunächst haben wir zwei Sprachen A, B über einem Alphabet Σ. A und B codieren gewissermaßen die Probleme, die wir aufeinander reduzieren wollen. Eine Reduktion ist dann quasi eine Übersetzung von einem Problem ins andere, d.h. wir wollen eine Ja-Instanz des einen Problems (ein Wort aus der Sprache A) in eine Ja-Instanz des anderen Problems (ein Wort aus der Sprache B) überführen, genauso mit Nein-Instanzen.

Lemma. *Seien $A, B \in \Sigma^*$ mit $A \leq_m B$.*

(i) Wenn A unentscheidbar ist, so ist auch B unentscheidbar.

(ii) Wenn B entscheidbar ist, so ist auch A entscheidbar.

Beweis. Nur der erste Fall, anderer ist Kontraposition: Sei A unentscheidbar. Angenommen B wäre entscheibar. Dann berechne für ein $x \in \Sigma^*$ das Wort $f(x)$, wobei f die berechenbare Reduktionsabbildung ist (welche existieren muss wegen $A \leq_m B$). Da B entscheidbar ist, existiert ein Algorithmus für B. Lasse diesen auf $f(x)$ laufen. Die Ausgabe ist auch die korrekte Ausgabe für x bzgl. A. Also ist A entscheidbar. Widerspruch. □

Wir sehen, dass wir oben bereits bei der Aussage, dass UG ist unentscheidbar, wenn AG unentscheidbar ist, bereits eine ähnliche Argumentation wie im Lemma benutzt haben (wir benutzen die Ausgabe des einen Algorithmus um die andere Ausgabe zu erhalten, beachte jedoch, dass das dort keine many-one-Reduktion war).
Wir zeigen nun, dass PCP $\leq_m$ AG. Um den Beweis nicht unnötig zu chiffrieren, werden wir dabei nicht die Domino-Menge $\mathcal{D}$ bzw. die kontextfreie Grammatik $\mathcal{G}$ als Wort über einem Alphabet auffassen. Es ist relativ klar, dass dies geht, zum Beispiel könnten $\mathcal{D}$ codiert werden als $u_1\#v_1\#\#\ldots\#\#u_n\#v_n$ über dem Alphabet $\Sigma \uplus \{\#\}$. Dabei steht $\#\#$ als Trennung zwischen den Dominos und ein ein einzelnes $\#$ für die Trennung zwischen u_i und v_i.

Satz. PCP $\leq_m$ AG.

Beweis. Sei die PCP-Instanz $\mathcal{D} \coloneqq \{[\frac{u_1}{v_1}], \ldots, [\frac{u_n}{v_n}]\}$ codiert als x gegeben. Sollte an der Codierung etwas falsch sein, können wir einfach $f(x)$ auf eine Codierung einer eindeutigen Grammtik setzen, z.B. $(\{S\}, \{a\}, \{S \to \varepsilon\}, S)$. Ansonsten setzen wir $f(x)$ auf die Codierung der Grammatik

$$\mathcal{G}_\mathcal{D} = (\{S, A, B\}, \Sigma \uplus \{1, \ldots, n\}, P, S)$$

mit

$$\begin{aligned} P = &\{S \to A \mid B\} \\ &\cup \{A \to u_i A i \mid u_i i : i \in \{1, \ldots, n\}\} \\ &\cup \{B \to v_i B i \mid v_i i : i \in \{1, \ldots, n\}\}. \end{aligned}$$

Es ist klar, dass f berechenbar ist, denn wir müssen einfach nur Domino für Domino durchgehen und die entsprechenden Regeln hinzufügen. Wir zeigen nun, dass $\mathcal{G}_\mathcal{D}$ genau dann mehrdeutig ist (d.h. zu AG gehört), wenn $\mathcal{D}$ eine Lösung besitzt.

- Sei $\mathcal{D} \in$ PCP. D.h. es ex. Indexfolge $(i_1, \ldots, i_k)$, sodass $u_{i_1} \ldots u_{i_k} = v_{i_1} \ldots v_{i_k}$. Dann existieren für das Wort $u_{i_1} \ldots u_{i_k}$ zwei Linksableitungen in $\mathcal{G}_\mathcal{D}$, nämlich:
  - $S \to A \to u_{i_1} A i_1 \to \ldots \to u_{i_1} \ldots u_{i_k} i_k \ldots i_1$ und
  - $S \to B \to v_{i_1} B i_1 \to \ldots \to v_{i_1} \ldots v_{i_k} i_k \ldots i_1$.

  Beachte dabei, dass $u_{i_1} \ldots u_{i_k} = v_{i_1} \ldots v_{i_k}$. Die Indexfolge steht außerdem in rechts nach links geschrieben rechts neben dem Wort und stimmt in beiden Ableitungen natürlich überein. Also ist $\mathcal{G}_\mathcal{D} \in$ AG.

- Sei $\mathcal{G}_\mathcal{D} \in$ AG. Es gibt also für ein Wort $w \in L(\mathcal{G}_\mathcal{D}$ zwei verschiedene Linksableitungen. Nach Konstruktion hat w die Form $w = u i_k \ldots i_1$ mit $u \in \Sigma^*$ und $i_1, \ldots, i_k \in \{1, \ldots, n\}$. Nun gilt aber, dass wenn w zwei verschiedene Linksableitungen hat, dass eine davon mit der Produktion $S \to A$ beginnt und die andere mit $S \to B$, denn würden beide mit der selben Produktion beginnen müsste auch der Rest der Ableitung übereinstimmen, da sonst der $i_k \ldots i_1$-Teil in den Ableitungen unterschiedlich aussähe. Also ist $\bar{\imath} = (i_1, \ldots, i_k)$ eine Lösung für die PCP-Instanz $\mathcal{D}$, weil die durch $\bar{\imath}$ induzierten Wörter $u_{i_1} \ldots u_{i_k}$ und $v_{i_1} \ldots v_{i_k}$ übereinstimmen. Also ist $\mathcal{D} \in$ PCP. □

Mit dem Lemma oben folgt, dass AG unentscheidbar ist und somit auch UG.