Prof. Javier Esparza
Philipp Czerner, Kevin Kappelmann, Valentin Krasotin
Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2024 – Hausaufgabenblatt 10

- Die Aufgaben werden in folgender Reihenfolge korrigiert: H10.1, H10.2.
- Die Knobelaufgabe bitte separat auf Moodle abgeben. Sie wird korrigiert.

Aufgabe H10.1. (*Falsch/Wahr*) 4 Punkte

Entscheiden Sie, ob die folgenden Aussagen wahr oder falsch sind. Begründen Sie ihre Behauptung. Ein ausführlicher formaler Beweis ist nicht gefordert, aber eine umfassende Begründung.

(a) Sei $L \subseteq \Sigma^*$ eine unentscheidbare Sprache, $w \in \Sigma^*$. L lässt sich mittels $f : \Sigma^* \to \Sigma^*$ mit

$$f(x) = \begin{cases} w & \text{falls } x \in L \text{ oder } x = w \\ x & \text{sonst} \end{cases}$$

auf $L \cup \{w\}$ reduzieren.

(b) Sei $L \subseteq \Sigma^*$ eine unentscheidbare Sprache, sei $w \in \Sigma^*$. Dann ist $L \cup \{w\}$ unentscheidbar.

(c) Sei $\Sigma = \{a\}$. Alle Sprachen über Σ sind entscheidbar.

(d) Sei $L \subseteq \Sigma^*$. Falls L^* entscheidbar ist, ist auch L entscheidbar.

Lösungsskizze.

(a) Falsch, die Reduktion hat mehrere Probleme:

- Falls $w \notin L$ gilt $f(w) \in L \cup \{w\}$
- Aus der Definition von f ist die Berechenbarkeit von f nicht ersichtlich, da $x \in L$ nicht berechenbar ist.

(b) Wahr, wenn $w \in L$, ist $L \cup \{w\} = L$ und daher unentscheidbar. Wenn $w \notin L$, zeigen wir die Unentscheibarkeit per Reduktion. Hierfür benutzen wir die folgende Reduktionsfunktion

$$f(x) = \begin{cases} v & \text{falls } x = w \\ x & \text{sonst} \end{cases}$$

wobei $v \neq w$ ein Wort nicht in L ist. Ein solches Wort v existiert. Beweis per Widerspruch: Nehme an, es gäbe kein solches v. Dann wäre $L \cup \{w\} = \Sigma^*$. Dann wäre aber L entscheidbar (Zur Berechnung der charakteristischen Funktion prüfe ob die Eingabe w ist, wenn ja gib 0, sonst 1, zurück). Widerspruch.

(c) Falsch, es gibt überabzählbar viele Sprachen über Σ, aber nur abzählbar viele entscheidbare Sprachen (da es nur abzählbar viele TMs gibt).

(d) Falsch. Gegenbeispiel: Wähle $\Sigma = \{a\}$, $L^* = \Sigma^*$. Sei L' irgendeine unentscheidbare Menge über Σ (siehe (c)). Setze $L = L' \cup \{a\}$. Nach (b) ist L unentscheidbar und es gilt $L^* = \Sigma^*$.

Aufgabe H10.2. (*Alle Jahre wieder*) 7+4 Punkte

Heute ist „Tag der offenen Tür“ in Theos Schule und Dora besucht ihn. Zu diesem Anlass gibt es einen Test über die Geschichte des Sommerfestes, das jedes Jahr entweder in Estlingen oder in Oberfeld ausgetragen wird. Dies folgt einem ausgeklügeltem Schema, das Theo leider nicht kennt, da er zu sehr damit beschäftigt war, die Sandkörner in seinem Sandkasten zu zählen. Zum Glück weiß Dora genau, in welchen Jahren das Sommerfest in Estlingen stattfindet, und kann Theo helfen.

Allerdings ist Doras Kindergarten in Oberfeld – in Theos Estlinger Schule werden nur Jahreszahlen in der traditionellen Estlinger Form verwendet: an ungerade Jahreszahlen wird eine 3 angehängt. Statt dem Jahr 517 schreibt man z.B. 5173, aber 1024 schreibt man immer noch als 1024.

Sei $S \subseteq \mathbb{N}$ die Menge der Jahre, in denen das Sommerfest in Estlingen stattfindet, und $S' \subseteq \mathbb{N}$ die Menge der gleichen Jahre in Estlinger Form. Es gilt also $x \in S$ genau dann, wenn x gerade ist und $x \in S'$, oder wenn x ungerade ist und $10x + 3 \in S'$. Außerdem gilt $0 \notin S$.

(a) Theo will den Test lösen. Dafür möchte er sich ein Verfahren ausdenken, mit dem er berechnen kann, in welchen Jahren das Sommerfest in Estlingen ausgetragen wurde. Schreiben Sie also ein `WHILE`-Programm, das die charakteristische Funktion $\chi_{S'}$ berechnet (siehe Definition 5.28). Verwenden Sie in Ihrem `WHILE`-Programm den zusätzlichen Befehl dora; nach Ausführung von dora wird x_0 auf 1 gesetzt, falls $x_1 \in S$, sonst auf 0. (dora berechnet also χ_S.) Erklären Sie ihren Ansatz in natürlicher Sprache.

Sie dürfen alle Makros aus der Vorlesung verwenden (z.B. `MOD`, `DIV`) sowie eigene definieren.

(b) Zeigen Sie, dass S' reduzierbar auf S ist, indem sie eine berechenbare, totale Funktion $f : \mathbb{N} \to \mathbb{N}$ angeben, sodass $x \in S' \Leftrightarrow f(x) \in S$ für jedes $x \in \mathbb{N}$ gilt.

Sie müssen die Funktion nicht in geschlossener Form angeben, eine präzise Beschreibung genügt.

Lösungsskizze. **(a)** Jede Zahl, die auf 3 endet, ist ungerade. Wenn wir ein Jahr in Estlinger Form konvertieren, bleiben ungerade Jahre also ungerade, und gerade bleiben gerade. Wenn uns eine Jahreszahl in Estlinger Form gegeben wird, können wir diese somit zur normalen Form übersetzen, indem wir (1) sie durch 10 teilen (ohne Rest) wenn sie ungerade ist und (2) sie gleich lassen, wenn sie gerade ist.

Wenn wir die normale Jahreszahl haben, fragen wir Dora.

```
s := x_1 MOD 2
if s ≠ 0 then
    s := x_1 MOD 10
    x_1 := x_1 DIV 10
    t := x_1 MOD 2
    if s ≠ 3 OR t ≠ 1 then
        x_1 := 0
dora
```

Hierfür müssen wir zusätzlich beachten, was wir mit Eingaben machen, die nicht in Estlinger Form sind (z.B. 12345). Da diese Zahlen auch nicht in S' enthalten sein können, geben wir da 0 aus, indem wir $x_1 := 0 \notin S$ setzen.

(b) Eine einfache Lösung ist, f als die Funktion zu definieren, die von dem `WHILE`-Programm aus (a) ausgerechnet wird, wenn man die letzte Zeile (mit dem dora Befehl), ersetzt durch $x_0 := x_1$. Dann ist f offensichtlich berechenbar. Außerdem ist f total, da das Programm immer terminiert. Da das ursprüngliche Programm (mit dem dora-Befehl) die Funktion $\chi_{S'}$ berechnet und dora die Funktion χ_S berechnet, muss das modifizierte Programm eine Funktion f berechnen, die $x \in S' \Leftrightarrow f(x) \in S$ erfüllt.

Alternativ können wir die Funktion auch direkt angeben:

$$f(x) := \begin{cases} x & \text{für } x \text{ gerade} \\ \frac{x-3}{10} & \text{für } x \equiv 3 \pmod{10} \text{ und } \frac{x-3}{10} \text{ ungerade} \\ 0 & \text{sonst} \end{cases}$$

Quizaufgabe H10.3. *(Auf- und Ab)* unkorrigiert (8 Punkte)

Wir betrachten folgende Eigenschaften: (1) abzählbar, (2) rekursiv aufzählbar, (3) semi-entscheidbar, und (4) entscheidbar. Geben Sie für jede der folgenden Mengen an, welche der vier Eigenschaften erfüllt sind. Bitte beachten Sie, dass (2-4) nur auf Sprachen definiert sind.

(a) $\emptyset$

(b) $\{w \in \mathbb{R} : w < \frac{1}{\sqrt{2}}\}$

(c) $\{w \in \{0,...,9\}^* : (0.w)_{10} < \frac{1}{\sqrt{2}}\}$

(d) $\{w \in \{0,1\}^* : \{1, 10, 111\} \cap L(M_w) \neq \emptyset\}$

(e) $\{w \in \{0,1\}^* \mid \forall v \in L(M_w) : (7(v)_2 - 31)^2 > 8\}$

(f) $\{w \in \{0,1\}^* : M_w \text{ hält für ein } v \in \{0,1\}^* \text{ in } |v| \text{ Schritten}\}$

Für (c) verwenden wir die Notation aus H5.7.

Lösungsskizze.

	(1)	(2)	(3)	(4)
(a)	×	×	×	×
(b)	·	·	·	·
(c)	×	×	×	×
(d)	×	×	×	·
(e)	×	×	×	×
(f)	×	×	×	·

Anmerkungen: (2) und (3) sind äquivalent. (c) Die Sprache kann man entscheiden, indem man die Eingabe quadriert und überprüft, ob sie kleiner als $\frac{1}{2}$ ist. (d) Für ein gegebenes w kann man M_w auf 1, 10 und 111 gleichzeitig laufen lassen – wenn eine dieser Eingaben terminiert, akzeptieren wir. (e) Die Ungleichung $(7x - 31)^2 \leq 8$ hat keine ganzzahlige Lösung, da das Polynom sein Minimum bei $4 < \frac{31}{7} < 5$ annimmt, aber $(7 \cdot 4 - 31)^2 = 9$ und $(7 \cdot 5 - 31)^2 = 16$ beide größer als 8 sind. Die Eigenschaft ist also immer wahr. (f) Wir können über alle Wörter v iterieren und M_w jeweils für $|v|$ Schritte simulieren, die Sprache ist also semi-entscheidbar. Aber nicht entscheidbar (man kann von $\mathcal{H}_0$ reduzieren, indem man die Eingabe ignoriert).

Knobelaufgabe H10.4. (*Quine*)

Wir betrachten die Programmiersprache `STRING`. Ähnlich zu `WHILE` und `GOTO` verwendet ein Programm Register $x_0, ..., x_k$, diese haben nun aber Werte in Σ^*, mit Alphabet $\Sigma := \{\texttt{a}, ..., \texttt{z}, \texttt{0}, ..., \texttt{9}, \texttt{:}, \texttt{=}\}$. Seien $i, j, k \in \mathbb{N}$, $w \in \Sigma^*$, und P_1, P_2 `STRING`-Programme. Die gültigen `STRING`-Programme sind:

- $x_i := |w| : w$ setzt Register x_i auf w (z.B. `x1:=4:theo`, die Länge von w gefolgt von einem Doppelpunkt und w selbst – dies erlaubt, Sonderzeichen in w zu verwenden)
- $x_i := x_j x_k$ setzt x_i auf die Konkatenation der Werte von x_j und x_k
- $x_i :=$ `len` x_j setzt x_i auf die Länge des Wertes von x_j, als Dezimalzahl kodiert
- $x_i :=$ `pop` x_j entfernt das letzte Zeichen von x_j und schreibt es in x_i (wenn $x_j \neq \varepsilon$)
- `while` x_i `do` P_1 `end` führt P_1 aus, solange $x_i \neq \varepsilon$
- $P_1 P_2$ führt zunächst P_1 aus und dann P_2.

Diese Programme kodieren wir als Wort über Σ^*, z.B.

```
x2 := 5:olleh
x1 := x1 x2
while x1 do
    x2 := pop x1
    x0 := x0 x2
end
```

(Leerzeichen und Zeilenumbrüche sind nur zur besseren Lesbarkeit eingefügt worden, und nicht Teil des Programmtextes.) `STRING`-Programme berechnen partielle Funktionen $f : (\Sigma^*)^k \to \Sigma^*$, wobei die Eingabe in Registern $x_1, ..., x_k$ steht, alle anderen Register auf ε initialisiert werden, und das Ergebnis in x_0 ausgegeben wird. Obiges Beispiel berechnet also die Funktion $f(w) = \texttt{hello}\, w^R$, etwa $f(\texttt{:dlrow}) = \texttt{hello:world}$.

Sei $w \in \Sigma^*$ und P_w das `STRING`-Programm, das über w kodiert wird. Konstruieren Sie ein w, sodass P_w eine Funktion f mit $f(\varepsilon) = w$ berechnet.

Hinweis: Nicht alle Anweisungen von `STRING`-Programmen sind zum Lösen dieser Aufgabe notwendig; obige Definition stellt sicher, dass `STRING` Turing-vollständig ist. Sie finden einen Interpreter für `STRING` unter diesem Link.

Lösungsskizze. Sei $f_w : \Sigma^* \to \Sigma^*$ die Funktion, die von P_w berechnet wird, für $w \in \Sigma^*$.

Wir konstruieren zunächst ein Programm P_u, sodass für alle $w \in \Sigma^*$ gilt, dass $f_u(w) = vw$, für ein $v \in \Sigma^*$, das ein Programm kodiert, das x_1 auf w setzt. Intuitiv berechnet P_u also auf Eingabe w den Code eines Programms, dass zunächst w in x_1 schreibt, und dann das von w kodierte Programm ausführt.

```
x0 := 4:x1:=
x2 := len x1
x0 := x0 x2
x2 := 1::
x0 := x0 x2
x0 := x0 x1
x0 := x0 x1
```

Unser gesuchtest Wort w mit $f_w(\varepsilon) = w$ ist nun einfach $w := f_u(u)$. Nach Wahl von P_u gilt $f_u(u) = vu$, wobei P_v Register x_1 auf u setzt. Es folgt $f_w(\varepsilon) = f_{vu}(\varepsilon) = f_u(u) = w$. Der Vollständigkeit halber geben wir w auch direkt an:

x1:=58:x0:=4:x1:=x2:=lenx1x0:=x0x2x2:=1::x0:=x0x2x0:=x0x1x0:=x0x1
x0:=4:x1:=x2:=lenx1x0:=x0x2x2:=1::x0:=x0x2x0:=x0x1x0:=x0x1