2.6 Unentscheidbarkeit von Problemen

Oft werden Sprachen auch als (Entscheidungs-)Probleme bezeichnet. Worte werden dann Instanzen genannt. Dann werden sie oft als Gegeben/Gefragt - Paare beschrieben.

Bsp:

Gegeben: zwei DFAs A_1 und A_2

Gefragt: $L(A_1) = L(A_2)$?

$$L_{\ddot{A}} = \{ (A_1, A_2) \mid L(A_1) = L(A_2) \}$$

Ein Problem heißt entscheidbar, falls es ein Verfahren A gibt, so dass

- A stoppt nach endlich vielen Schritten bei Eingabe einer beliebigen Instanz des Problems
- gibt eine korrekte Ja/Nein-Antwort

Ein Problem heißt semi-entscheidbar, falls es ein Verfahren A gibt, so dass

- A stoppt nach endlichen Schritten nach Eingabe einer positiven Instanz
- gibt dann eine korrekte Ja-Antwort
- gibt keine Antwort bei negativen Instanze.

Bsp:

1) SAT_{PL} (Erfüllbarkeitsproblem für Aussagenlog.L)

Gegeben: Eine aussagenlogische Formel φ

Gefragt: Gibt es eine Belegung der Variablen, die φ wahr macht?

$$\begin{array}{cccccccc} (A & \wedge & B) & \vee & (C & \wedge & \neg D) \\ 1 & & 1 & & 0 & & 1 \end{array}$$

2) $UNSAT_{PL}$

Gegeben: Eine aussagenlog. Formel φ

Gefragt: Ist φ nicht erfüllbar?

→ auch entscheidbar!

3) SAT_{FOL}

<u>Gegeben:</u> Eine Formel der Prädikatenlogik 1. Stufe φ

<u>Gefragt:</u> Ist φ erfüllbar?

$\longrightarrow$ ist unentscheidbar

$\longrightarrow$ nicht einmal semi-entscheidbar

<u>Def.</u> Eine Sprache / Problem $L \subseteq \Sigma^*$ heißt <u>entscheidbar</u>,

falls die charakteristische Fkt von L

$$\underline{\chi_L} : \Sigma^* \to \{0,1\}, \quad \chi_L(x) = \begin{cases} 1, & \underline{\text{falls } x \in L} \\ 0, & \underline{\text{falls } x \notin L} \end{cases}$$

berechenbar ist.

<u>Def</u> Eine Sprache / Problem L heißt <u>semi-entscheidbar</u>

falls die "halbe" charakteristische Fkt von L

$$\underline{\chi_L'} : \Sigma^* \to \{0,1\} \quad \chi_L'(x) = \begin{cases} 1, & \text{falls } x \in L \\ \text{undef} & \text{sonst} \end{cases}$$

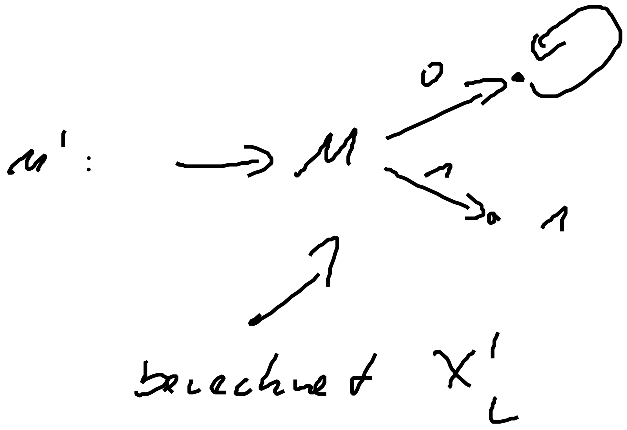
berechenbar ist.

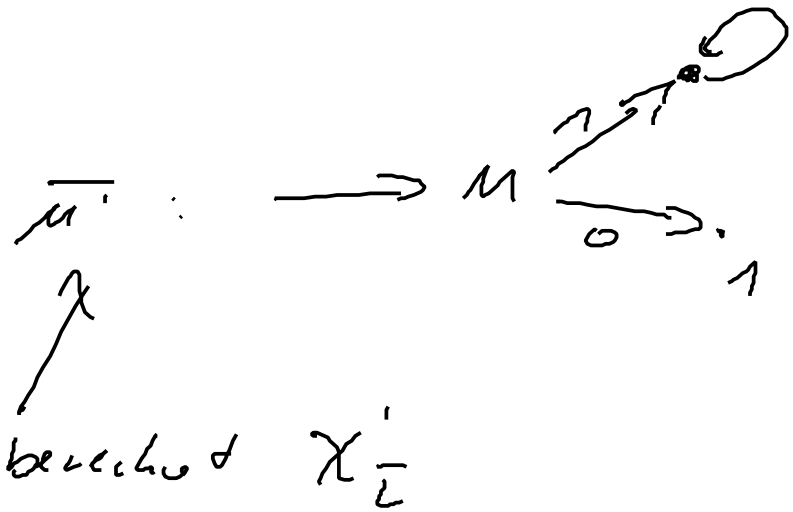
<u>Bem:</u> χ_L' ist eine partielle Funktion!

Satz Eine Sprache $L \subseteq \Sigma^*$ ist genau dann entscheidbar, wenn sowohl L als auch $\overline{L}$ $(= \Sigma^* - L)$ semi-entscheidbar ist.

Bew:

"$\Rightarrow$": Angenommen L ist entscheidbar. Dann gibt es TM M, die χ_L berechnet. Konstruiere M' und $\overline{M'}$:





"$\Leftarrow$": Angenommen L und $\bar{L}$ sind semi-entscheidbar.

D.h. es gibt TM M' und $\bar{M}'$, die χ'_L und $\chi'_{\bar{L}}$ berechnen (OBdA sind dies 1-Band TMs). Konstruiere 2-Band-TM M:

1) M kopiert die Eingabe von Band 1 auf Band 2 und fährt dann beide Köpfe zum linken Rand

2) M arbeitet dann auf Band 1 wie M' und auf Band 2 wie $\bar{M}'$.

3) stoppt M' mit Wert 1 auf Band 1, dann stoppt M. Stoppt $\bar{M}'$ mit Wert 1 auf Band 2, löscht M Band 1 und schreibt 0 auf Band 1 und stoppt.

Da L und $\bar{L}$ semi-entscheidbar sind, stoppt M bei jeder Eingabe nach endlich vielen Schritten mit der korrekten Ausgabe. Also berechnet M tatsächlich χ_L $\square$

Def $L \subseteq \Sigma^*$ heißt $\boxed{\text{rekursiv aufzählbar}}$, falls $L = \emptyset$ oder falls es eine totale und berechenbare Funktion gibt $f : \mathbb{N} \to \Sigma^*$ mit

$$L = f(\mathbb{N}) := \{ f(n) \mid n \in \mathbb{N} \}$$

Bem: f muss nicht injektiv sein.

Bem: Verwechslungsmöglichkeiten:

rekursive Sprache = entscheidbare Sprache

$\boxed{\text{abzählbare}}$ Menge = Aussage über Kardinalität (nicht über Berechenbarkeit)

<u>Satz</u> Eine Sprache $L \in \Sigma^*$ ist genau dann rekursiv aufzählbar, wenn sie semi-entscheidbar ist.

<u>Bew:</u>
"$\Longrightarrow$" Sei OBdA $L \neq \emptyset$ rekursiv aufzählbar, wobei $f: \mathbb{N} \to \Sigma^*$ eine totale und berechenbare Fkt mit $f(\mathbb{N}) = L$.

<u>Programm:</u>

```
Input(w);
x := 1;
WHILE TRUE DO
    IF f(x) = w THEN RETURN 1;
    x := x + 1;
END
```

"$\Leftarrow$": Sei $L \subseteq \Sigma^*$ semi-entscheidbar, d.h.

χ'_L ist berechenbar mit TM M'.

<u>Zu zeigen</u>: Es ex. eine totale und berechenbare Fkt $f : \mathbb{N} \to \Sigma^*$ und $f(\mathbb{N}) = L$.

<u>Programm</u>:

Input (n);

$k := e(n)$;

$l := f(n)$;

IF M' stoppt nach Eingabe

der $k$ten Wortes w_k nach

<u>l Schritten</u> mit 1 als Ausgabe

THEN RETURN w_k

ELSE RETURN w^*

END

$n = c(k, n)$

$e, f : \mathbb{N} \to \mathbb{N}$ sind Umkehrfkt:

$e(c(k,n)) = k$

$f(c(k,n)) = n$

wir ordnen alle unsere Worte lexikalisch über Σ^*

Dabei $w^* \in L$ ist ein fixes Wort.

<u>Bem</u>: Beweistechnik heißt "dove tailing"

Das Verfahren für $f(n)$ stoppt für jedes n und gibt nur Wörter aus L zurück.

Umgekehrt, falls $w \in L$, dann gibt es k mit $w = w_k$ und es ex. $\ell \in \mathbb{N}$, so dass M' nach ℓ Schritte auf w hält und 1 ausgibt. D.h. aber, dass unser Verfahren für $n = c(k, \ell)$ gibt w zurück. D.h. für jedes Wort w ex. ein n mit $f(n) = w$. $\square$

Zusammenfassend sind folgende Aussagen äquivalent:

(a) L ist semi-entscheidbar

Def

(b) χ'_L ist berechenbar

Def

einfach

(c) L ist Definitionsbereich einer berechenbaren Fkt.

oben gezeigt

(d) L ist rekursiv aufzählbar

Def

einfach

(e) L ist Wertebereich einer berechenbaren Fkt.

FS 16 (2.3)

(f) $L = T(M)$ für ein TM M

FS 15 (1.8)

(g) L ist vom Typ 0

Das Halteproblem

Gegeben: Eine Kodierung einer TM M code(M) und ein Eingabewert w.

Gefragt: Hält M auf w?

$$H = \{ (\text{code}(M), w) \mid M \text{ hält auf } w \}$$

Kodieren von TMs:

Sei $\Gamma = \{ a_0, \dots, a_k \}$

$Z = \{ z_0, \dots, z_m \}$

Jede δ-Regel $\boxed{\delta(z_i, a_j) = (z_{i'}, a_{j'}, y)}$ kann kodiert werden

$$\boxed{\#\#\, \text{bin}(i) \# \text{bin}(j) \# \text{bin}(i') \# \text{bin}(j') \# \text{bin}(m)}$$

mit $m = \begin{cases} 0 ; & y = L \\ 1 ; & y = R \\ 2 ; & y = N \end{cases}$

Jede TM kann über $\{0, 1, \#\}$ kodiert werden.

Aber auch über $\{0,1\}$:

$$0 \mapsto 00$$
$$1 \mapsto 01$$
$$\# \mapsto 11.$$

D.h. jede TM kann über $\{0,1\}$ kodiert werden.

Sei $\hat{M}$ irgendeine TM (aber fix).

$$M_w = \begin{cases} \hat{M}, & \text{falls } w \text{ keine Kodierung einer TM ist} \\ M, & \text{falls } w = \text{code}(M). \end{cases}$$

$$E = \{ w \mid M_w \text{ hält auf dem leeren Band} \}$$

$\rightarrow$ ist unentscheidbar!
