1 rekursive Mengen

1.1 Definition

1.1.1 informal

Eine Menge heißt rekursiv oder entscheidbar, wenn ihre charakteristische Funktion berechenbar ist.

1.1.2 formal

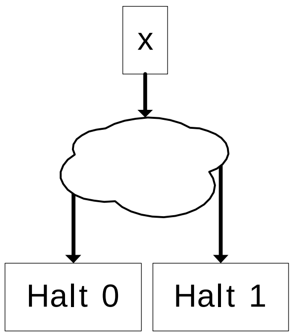
Eine Menge $A \subseteq N^k$ $(k \geq 1)$ heißt rekursiv, gdw. die charakteristische Funktion

$$cf_A(x) := \begin{cases} 1 & \text{falls } x \in A \\ 0 & \text{sonst} \end{cases}$$

berechenbar ist.

1.1.3 Turingmaschine

Es gibt eine Turingmaschine, die in endlicher Zeit jedes beliebige Element aus N^k in seiner Zugehörigkeit zu A entscheiden kann (also cf_A berechnet), also eine Aussage „ja“ oder „nein“ in jedem Fall liefern kann.



1.2 Beispiele

a) $\emptyset \subseteq N^k$ und N^k

b) jede endliche Teilmenge $A \subseteq N^k$

c) $\{ x \subseteq N \,;\, x \text{ ist Primzahl} \}$

d) $\{ (x,y) \in N^2 \,;\, x < y \}$

e) $\{ (i,x,t) \,;\, \Phi_i(x) \leq t \}$

1.3 Eigenschaften

Definition durch die Umkehrkorrespondenz (*): $A \subseteq N^k$ ist rekursiv, gdw. es eine totale berechenbare Funktion $f \in R^{(k)}$ gibt mit $A = f^{-1}(\{0\})$.

Charakterisierung über Aufzählungen: Eine Menge A ist rekursiv, gdw. es eine wachsende total rekursive Funktion f gibt mit A = Bild(f).

Abschlusseigenschaften ($k \in N^{\geq 1}$):

a) Seien $A_1, A_2 \subseteq N^k$ rekursiv.
Dann sind auch $N^k \backslash A_1$, $A_1 \cup A_2$ und $A_1 \cap A_2$ rekursiv.

b) Seien $A \subseteq N$ rekursiv und $f \in R^{(k)}$.
Dann ist $f^{-1}(A)$ rekursiv.

c) Sei $A \subseteq N^k$.
Dann gilt: A rekursiv $\Leftrightarrow$ $\pi^{(k)}(A)$ rekursiv.

d) Seien $A_1, A_2 \subseteq N$ und $A_1, A_2 \neq \emptyset$.
Dann gilt A_1 rekursiv und A_2 rekursiv $\Leftrightarrow$ $A_1 \times A_2$ rekursiv.

2 rekursiv-aufzählbare Mengen

2.1 Definition

2.1.1 informal

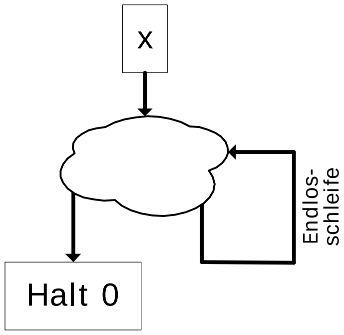
Eine Menge heißt rekursiv-aufzählbar, wenn es eine berechenbare partielle Funktion f gibt, deren Definitionsbereich diese Menge darstellt.

2.1.2 formal

Eine Menge $A \subseteq N^k$ heißt rekursiv-aufzählbar (r.a.), gdw. es eine partielle berechenbare Funktion $f :\subseteq N^k \rightarrow N$ gibt mit $A = Def(f)$

2.1.3 Turingmaschine

Es gibt eine Turingmaschine, welche für jedes $x \in A$ die Antwort „ja“ liefert, im Falle $x \notin A$ keine Antwort gibt.



2.2 Beispiele

u.a. jede rekursive Menge (mehr unter Kapitel 3 „Zusammenhänge“)

2.3 Eigenschaften

Charakterisierung durch Bild: Eine Menge A ist rekursiv-aufzählbar, wenn sie leer oder das Bild einer total rekursiven Funktion f ist.

Projektionssatz: Eine Menge A ist rekursiv-aufzählbar, gdw. sie die Projektion einer rekursiven Menge B ist.

$$[\ A \subseteq N^k \text{ ist r.a.} \Leftrightarrow A = \{\ x \in N^k\ ;\ (\exists t)\ (x,t) \in B\ \} \text{ für ein rekursives } B \subseteq N^{k+1}\ ]$$

Abschlusseigenschaften:

a) Seien $A_1, A_2 \subseteq N^k$ r.a.
Dann sind auch $A_1 \cap A_2$ und $A_1 \cup A_2$ r.a.

b) Sei $A \subseteq N$ r.a. und $f \in P^{(k)}$.
Dann ist $f^{-1}(A)$ r.a.

c) Sei $A \subseteq N^k$.
Dann gilt: A r.a. $\Leftrightarrow \pi^{(k)}(A)$ r.a.

3 Zusammenhänge

3.1 Rekursivität umfasst rekursive Aufzählbarkeit

Eine Menge $A \subseteq N^k$ ist rekursiv, gdw. A und $N^k \backslash A$ rekursiv-aufzählbar sind.

(„Wenn man sowohl eine Menge A, als auch deren Komplement rekursiv aufzählen kann, gibt es auch eine berechenbare charakteristische Funktion cf_A.“)

3.2 rekursiv-aufzählbare, aber nicht rekursive Mengen

Die Mengen

$$K_\varphi := \{\, i \in N \;,\; \varphi_i(i) \text{ existiert} \,\}$$
$$K_\varphi^0 := \{\, (i,x) \in N^2 \;;\; \varphi_i(x) \text{ existiert} \,\}$$

sind rekursiv-aufzählbar, aber nicht rekursiv. Deren Komplemente $N \backslash K_\varphi$ und $N^2 \backslash K_\varphi^0$ sind nicht rekursiv-aufzählbar.

<u>Beweis</u>

Der Umstand, dass K_φ nicht rekursiv ist, kann durch Diagonalisierung nachgewiesen werden:

Sei dazu $g :\subseteq N \rightarrow N$ definiert durch

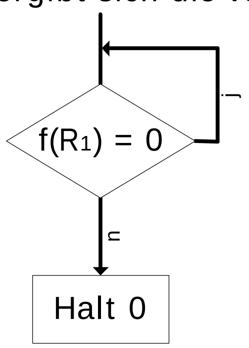
$$g(i) := \begin{cases} \text{div} & \text{falls } \varphi_i(i) \text{ existiert} \\ 0 & \text{sonst} \end{cases}$$

(**) Dieses g ist **nicht** berechenbar — andernfalls gäbe es ein $j \in N$ mit $\varphi_j = g$ und damit für

$j \in K_\varphi : g(j) = \text{div}$ dies steht im Widerspruch zu $\varphi_j(j)$ existiert
$j \notin K_\varphi : g(j) = 0$ dies steht im Widerspruch zu $\varphi_j(j) = \text{div}$

Angenommen K_φ sei rekursiv.
Dann existiert nach *[1] ein $f \in R^{(1)}$ mit $K_\varphi = f^{-1}(\{0\})$.

Dazu ergibt sich die verallgemeinerte Registermaschine M mit Flussdiagramm



[1] siehe Seite 1 zum Thema Umkehrkorrespondenz

und liefert offensichtlich $f_M = g$. Dies bedeutet aber auch Berechenbarkeit von g, welches in **[2] bereits widerlegt wurde.

Die Annahme über K_φ war also falsch; daraus folgt K_φ ist nicht rekursiv.

3.3 „unlösbare Probleme“

Aus dem o.g. Umstand, dass K_φ nicht rekursiv ist, ergeben sich konkrete Anwendungsprobleme:

3.3.1 Selbstanwendbarkeitsproblem, Halteproblem

Da K_φ nicht rekursiv ist, kann man nicht wissen, ob die i-te Bandmaschine bei Eingabe von i hält oder nicht: Das *Selbstanwendbarkeitsproblem für φ ist rekursiv unlösbar*. Genauso ist unentscheidbar, ob die i-te Bandmaschine bei Eingabe von x hält oder nicht: Das *Halteproblem für φ ist rekursiv unlösbar*.

Es lässt sich also kein Algorithmus konstruieren, der ein gegebenes Programm auf Fehler (z.B. Endlosschleifen) kontrolliert...

3.3.2 Satz von Rice, Korrektheitsproblem

Eine Eigenschaft von Funktionen aus $P^{(1)}$ kann man durch eine Teilmenge $F \subseteq P^{(1)}$ beschreiben. Diese Menge ist nicht rekursiv:

> Falls $F \subseteq P^{(1)}$ und $F \neq \emptyset$ und $F \neq P^{(1)}$ ist,
> dann ist die Menge $\varphi^{-1}(F) = \{ i \in N \, ; \, \varphi_i \in F \}$ nicht rekursiv.

Das *Korrektheitsproblem für f ist unentscheidbar*. Man kann also keinen Algorithmus angeben, vorgegebene Programme auf „geforderte Eigenschaften“ entscheidet. Eine automatische, programmgesteuerte Qualitätssicherung kann also immer nur näherungsweise funktionieren...

3.3.3 Postsches Korrespondenzproblem

Sei Σ ein Alphabet mit mindestens 2 Symbolen, es sei $\# \notin \Sigma$. Dann ist die Menge

$$PK := \{ u_1\#v_1\#...\#u_k\#v_k \, ; \, k \geq 1, \, u_i, v_i \in \Sigma^*, \text{ es gibt } n \in N \text{ und es gibt } i_1,...,i_n \in \{1,...,k\} \text{ mit } u_{i1}u_{i2}...u_{in} = v_{i1}v_{i2}...v_{in} \}$$

rekursiv-aufzählbar, aber nicht rekursiv.

Diese Unlösbarkeit für Wortpaare hat praktische Auswirkungen im Bereich des pattern-matching.

[2] siehe Seite 4