Gibt es Sprachen, die nicht in P liegen?

Bem.: von den folgenden Sprachen weiss man nicht, ob sie in P liegen:

- HAMILTON = {⟨G⟩ | G ist ungerichteter Graph, in dem es einen Kreis gibt, der jeden Knoten genau 1x benutzt}
- CLIQUE = {⟨G,k⟩ | G ist ungerichteter Graph, in dem es einen vollständigen Untergraphen auf k Knoten gibt}
- GRAPHISO = {⟨G,H⟩ | G und H sind isomorphe Graphen}
- SUBGRAPHISO = {⟨G,H⟩ | G Graph und H ist isomorph zu Untergraphen von G}
- 3-COLOR = {⟨G⟩ | G ist ein 3-färbbarer Graph}

Gibt es Sprachen, die nicht in P liegen?

Bem.: von den folgenden Sprachen weiss man nicht, ob sie in P liegen:

- SAT = {⟨F⟩ | F ist erfüllbare aussagenlogische Formel}
- CNFSAT = {⟨F⟩ | F ist erfüllbare aussagenlogische Formel in konjunktiver Normalform}
- 3-CNFSAT = {⟨F⟩ | F ist erfüllbare aussagenlogische Formel in konjunktiver Normalform mit genau 3 Variablen pro Klausel}
- SUBSETSUM = {⟨S,t⟩ | $S \subset \mathbb{N}$, $t \in \mathbb{N}$ und es gibt $T \subset S$ mit $\sum_{x \in T} x = t$}
- PARTITION = {⟨S⟩ | $S \subset \mathbb{N}$ und es gibt $T \subset S$ mit $\sum_{x \in T} x = \sum_{x \in T^c} x$}

Gibt es Sprachen, die nicht in P liegen?

Bem.:

für L, L'ϵ{SAT,CNFSAT,3-CNFSAT,SUBSETSUM} gilt:

- L ϵ P gdw. L' ϵ P

  → Konzept der *Polynomzeitreduktion*
- L ist leicht (= in polynomieller Zeit) zu verifizieren

  Bsp.:
  - VER-SAT = {<F;B> | B ist erfüllende Belegung der aussagenlogischen Formel F} ϵ P
  - VER-SUBSETSUM = {<S,t;T> | S⊂**N**, tϵ**N**, T⊂S und $\sum_{x\epsilon T} x=t$} ϵ P

  → Komplexitätsklasse NP

Verifizierbare Sprachen

Def.:

- dtm V heisst Verifizierer für eine Sprache L, falls

  $L = \{w \mid \exists z : |z| \leq t_V(|w|) \text{ und } \langle w,z\rangle \epsilon L(V)\}$

- falls wϵL, so heisst z mit <w,z> ϵ L(V)

  Zeuge/Beweis für wϵL

- die Sprache L(V) heisst Verifikationssprache von L

- zu einer Klasse ***V*** von Sprachen ist

  VER-***V*** = {L | ∃Zϵ***V***: Z ist Verifikationssprache von L}

Effizient verifizierbare Sprachen

- VER-SAT = {⟨F;B⟩ | B ist erfüllende Belegung der Formel F}
  ist Verifikationssprache von SAT

- VER-SUBSETSUM = {⟨S,t;T⟩ | S⊂**N**, t∈**N** und $\Sigma_{x \in T} x = t$}
  ist Verifikationssprache von SUBSETSUM

→ SAT, SUBSETSUM ∈ VER-**P**

Eigenschaften verifizierbarer Sprachen

Satz: für jede Klasse *V* von Sprachen gilt *V* ⊂ VER-*V*

Bew.: *(→ Übung)*

Effizient verifizierbare Sprachen

Frage: *Wie sieht die Klasse VER-P aus?*

- Abschlusseigenschaften
- Verhältnis zur Klasse **P**
- Charakterisierung durch ein Maschinenmodell

Eigenschaften der Klasse VER-P

Satz:

- **P** $\subset$ **VER-P**
- $L_1, L_2 \in$ **VER-P** $\rightarrow L_1 \cup L_2, L_1 \cap L_2 \in$ **VER-P**

Bew.: *(→ Übung)*

Nichtdeterministische Turing-Maschine

Def.: Eine nichtdeterministische Turing-Maschine M (*ntm*) ist ein 7-Tupel $(Q,\Sigma,\Gamma,\delta,q_0,\underline{b},F)$ mit

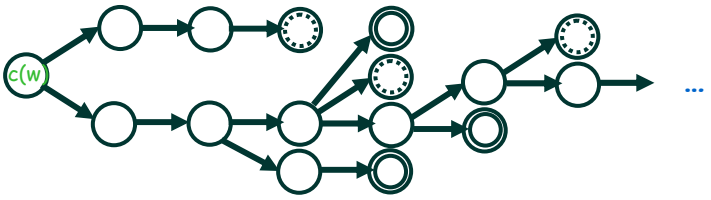
- Q endliche Menge von inneren Zuständen
- Σ endliches Eingabealphabet
- Γ endliches Bandalphabet $\Sigma c \Gamma$
- $\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R,S\}}$
  Zustandsüberführungsfunktion
  (L=links, R=rechts, S=stehenbleiben)
- $q_0 \in Q$ Anfangszustand
- $\underline{b} \in \Gamma \setminus \Sigma$ Blanksymbol
- $F = F_{ja} \cup F_{nein} c Q$ Menge der Endzustände mit
  - F_{ja} akzeptierenden Endzustände
  - F_{nein} nicht akzeptierenden Endzustände

Berechnungspfad einer ntm M

Def.: Ein Berechnungspfad von M auf $w \in \Sigma^*$ ist die Folge der Konfigurationen $B_M(w) = c_0,c_1,c_2,\ldots$ die M bei einer möglichen Abarbeitung von w durchläuft:

- $c_0 = c(w)$
- $c_{i+1} \in N_M(c_i)$ falls $c_i = uqv$ mit $q \in F^c$

"Überlagerung" aller Berechnungspfade von M auf w → Berechnungsbaum $T_M(w)$ von M auf w



Rechenzeit einer ntm M

Def.: Sei M ntm *die auf allen Eingaben hält*

$t_M(w)$:= Rechenzeit von M auf $w \in \Sigma^*$,
= Länge eines kürzesten akzeptierenden Berechnungspfades $B_M(w)$, falls w von M akzeptiert wird
= Länge eines längsten verwerfenden Berechnungspfades $B_M(w)$, falls w von M nicht akzeptiert wird

$t_M(n) := \max_{w \in \Sigma^n} t_M(w)$ heisst Rechenzeit von M

Nichtdeterministische Komplexitätsklassen

Def.: Für $f : \mathbf{N} \to \mathbf{N}$ ist

- $NTIME(f(n)) := \{L \subseteq \{0,1\}^* \mid$ es gibt eine ntm M mit $L=L(M)$ und $t_M(n) \leq f(n)\}$
  die von f definierte nichtdet. Zeitkomplexitätsklasse

- **NP** $:= \{L \subseteq \{0,1\}^* \mid$ es gibt eine ntm M mit $L=L(M)$ und $t_M(n) \leq n^r$ für ein $r>0\}$

ntm und die Klasse VER-P

Satz: NP = VER-P

Bew. …:

- $L \in NP$

  → es gibt eine ntm M mit L=L(M) und

  $t_M(n) \leq n^r$ für ein $r>0$

  → $L \in$ **VER-P** vermöge

  VER-L := {<x;c> | c ist akzeptierender Berechnungs-
  pfad der ntm M auf der Eingabe x}

ntm und die Klasse VER-P

… **Bew.:**

- $L \in$ **VER-P** vermöge VER-L∈P (V sei dtm mit VER-L=L(V))

  betrachte ntm M die bei Eingabe w

  1. nichtdeterministisch z mit $|z| \leq t_V(|w|)$ wählt
  2. V auf Eingabe <w,z> simuliert
  3. w akzeptiert gdw. die Simulation akzeptiert

  → L=L(M) und $t_M(n) = O(t_V(n))$

  → $L \in NP$

Eigenschaften der Klasse NP

Satz: (s.o.)

- $P \subset NP$
- $L_1, L_2 \in NP \rightarrow L_1 \cup L_2, L_1 \cap L_2 \in NP$

Sprachen in NP

- SAT, SUBSETSUM $\in$ NP (s.o.)
- CLIQUE $\in$ NP vermöge
  VER-CLIQUE := {⟨G,k;c⟩ | G Graph mit vollständigem Untergraphen auf der Menge c von k Knoten}
- 3-COLOR $\in$ NP vermöge
  VER-3-COLOR := {⟨G;f⟩ | G Graph und f ist 3-Färbung von G}

Komplemente von Sprachen in NP

Frage:

SAT^c, $SUBSETSUM^c$, $CLIQUE^c$, $3\text{-}COLOR^c \in NP$??

Def.: $co\text{-}NP := \{L^c \Sigma^* \mid L^c \in NP\}$

Zwei zentrale Fragen der Komplexitätstheorie:

- $P = NP$??
- $co\text{-}NP = NP$??