9.5 Třída NP

Definice 9.4. Nedeterministický algoritmus *se v některých krocích může libovolně rozhodnout pro některé z několika možných různých pokračování.*

Příklad. Uvažujme problém IND a následující algoritmus.

Vstup: graf G s uzly $u_1, \ldots, u_n$.

1) $X := \emptyset$ (*inicializace*)
2) pro $i = 1, 2, \ldots, n$ proveď:
   - a) buďto $X := X \cup \{u_i\}$,
   - b) nebo $X := X$

   (*nedeterministický krok*)
3) Jestliže $\{u_i, u_j\} \in H(G)$ pro některé $u_i, u_j \in X$, pak REJECT

   (*test nezávislosti*)
4) Je-li $|X| \geq k$ pak ACCEPT, jinak REJECT (*test* $|X| \geq k$)

Vlastnosti:

- v důsledku nedeterminističnosti 2. kroku je 2^n možných různých výpočtů,
- v G existuje nezávislá množina o alespoň k uzlech právě když **existuje** výpočet, končící ACCEPT.

Definice 9.5. Nedeterministický přijímací počítač *má (navíc oproti dříve definovanému přijímacímu počítači) příkaz* CHOOSE L1, L2, *kde L1, L2 jsou návěští.*

Definice 9.6. *Řekneme, že slovo w* je přijímáno *nedeterministickým přijímacím počítačem M, jestliže* existuje *přípustný výpočet počítače M s počáteční konfigurací danou slovem w a končící ACCEPT. V opačném případě řekneme, že nedeterministický přijímací počítač* odmítá *slovo w. Řekneme, že M* přijímá jazyk *J, jestliže pro každé slovo w je $w \in J \Leftrightarrow M$ přijímá w.*

Definice 9.7. *Řekneme, že nedeterministický přijímací počítač M* zpracuje slovo w v čase nejvýše t, *jestliže:*

- *buďto M přijímá w, a* **existuje** *přípustný výpočet určený slovem w a končící ACCEPT po nejvýše t krocích,*
- *nebo M odmítá w a* **každý** *přípustný výpočet končí po nejvýše t krocích.*

Definice 9.8. *Řekneme, že nedeterministický přijímací počítač M* pracuje v polynomiálně omezeném čase, *jestliže existuje polynom p takový, že libovolné slovo délky n zpracuje v čase nejvýše $p(n)$.*

Definice 9.9. *Třída NP je třída všech jazyků J takových, že existuje nedeterministický přijímací počítač, který pracuje v polynomiálně omezeném čase a přijímá jazyk J.*

Věta 9.3. $P \subset NP$.

(i) Je $J \in \mathrm{P}$?

(ii) Je alespoň $J \in \mathrm{NP}$?

(iii) Hledáme přesné řešení postupy typu „hrubá síla“.

(iv) Použijeme heuristiku a spokojíme se s přibližným řešením.

Algoritmus	Deterministický	Polynomiální	Přijímá J
(i) Dokazuje $J \in \mathrm{P}$	ANO	ANO	ANO
(ii) Dokazuje $J \in \mathrm{NP}$	NE	ANO	ANO
(iii) „Hrubá síla“	ANO	NE	ANO
(iv) Heuristika	ANO	ANO	NE

Polynomiální redukce a NP-úplné problémy

Definice 9.10. Překládací počítač *je počítač M s libovolným přístupem takový, že pro každý jeho výpočet (daný libovolnou počáteční konfigurací) platí:*

- každé provedení WRITE zapíše na vstupní pásku číslo 0 *nebo* 1*,*

- po konečném počtu kroků se výpočet zastaví provedením STOP.

Posloupnost nul a jedniček, zapsanou na výstupní pásce při výpočtu určeném slovem w, označíme $M(w)$.

Definice 9.11. *Řekneme, že jazyk J_1 je* redukovatelný *na jazyk J_2 (značíme $J_1 \lhd J_2$), jestliže existuje deterministický překládací počítač M pracující v polynomiálně omezeném čase takový, že pro každé slovo w je $w \in J_1 \Leftrightarrow M(w) \in J_2$.*

Věta 9.4.

1) $J_1 \lhd J_2,\ J_2 \lhd J_3 \ \Rightarrow\ J_1 \lhd J_3$ *(tranzitivita)*

2) $J_1 \lhd J_2,\ J_2 \in NP \ \Rightarrow\ J_1 \in NP$

3) $J_1 \lhd J_2,\ J_2 \in P \ \Rightarrow\ J_1 \in P$

$J_1 \lhd J_2$ znamená, že J_2 je alespoň tak těžký jako J_1.

Definice 9.12. *Jazyk J se nazývá* NP-úplný, *jestliže platí*

1) $J \in NP$,

2) pro každý $J' \in NP$ je $J' \triangleleft J$.

Třídu NP-úplných jazyků označíme NPC („NP-complete“).

Věta 9.5. *Platí: buď $P = NP$, nebo $P \cap NPC = \emptyset$.*

Jazyk J se nazývá *NP-těžký*, jestliže pro každý $J' \in NP$ platí $J' \triangleleft J$.

NP-úplnost problému SAT – Cookova věta

Věta 9.6. (Cook) SAT $\in$ NPC.

1. SAT $\in$ NP.

2. Pro každý $J \in$ NP je $J \triangleleft$ SAT.

Nechť

- $J \in$ NP,
- M je nedeterministický přijímací počítač, přijímající J v čase omezeném polynomem p.

Úkol: pro každé slovo $w \in J$ nalézt formuli f_w v KNF tak, že f_w je splnitelná, právě když M přijímá w.

Označíme:

n - délka slova w,

$m = p(n)$,

q - počet příkazů programu počítače M.

To znamená, že

- pouze prvních n polí vstupní pásky obsahuje čísla $\neq 0$,
- pouze prvních m paměťových buněk obsahuje číslo $\neq 0$,
- v žádné paměťové buňce není číslo větší než m,
- počet kroků počítače je nejvýše m.

K popisu všech konfigurací stačí znát hodnoty logických proměnných:

$a_i, \ i = 1, \ldots, n:$

$a_i = 1 \Leftrightarrow$ v i-tém poli vstupní pásky je číslo 1

$b_{it}, \ i = 1, \ldots, n, \ t = 0 \ldots, m$:

$b_{it} = 1 \Leftrightarrow$ po t-tém kroku je vstupní hlava na i-tém poli pásky

$c_{ijt}, \ i = 0 \ldots m, \ j = 0, \pm 1, \ldots, \pm m, \ t = 0, \ldots m:$

$c_{ijt} = 1 \Leftrightarrow$ v t-té konfiguraci je v i-té paměťové buňce číslo j

$d_{jt}, \ j = 1, \ldots, q, \ t = 0, \ldots, m:$

$d_{jt} = 1 \Leftrightarrow$ v t-té konfiguraci je v programovém registru číslo j

Mají-li proměnné popisovat kroky počítače M, který v m krocích přijme slovo w, musí splňovat následující podmínky:

1) hodnoty a_i se shodují s 0 a 1 ve slově w,
2) *hlava nemůže být současně na dvou místech*:
   pro každé t existuje nejvýše jedno i tak, že $b_{it} = 1$,
3) *v každé paměťové buňce je právě jedno číslo*:
   pro každé i, t existuje právě jedno j tak, že $c_{ijt} = 1$,
4) *totéž pro programový registr*:
   pro každé t existuje právě jedno j tak, že $d_{jt} = 1$,
5) jestliže $d_{jm} = 1$, pak j je návěští některého příkazu ACCEPT,
6) vztah hodnot proměnných mezi $t-1$ a t musí odpovídat prováděnému příkazu počítače.

Položíme

$$f_w = f_{w1} \wedge f_{w2} \wedge f_{w3} \wedge f_{w4} \wedge f_{w5} \wedge f_{w6},$$

kde f_{wi} je formule daná podmínkou i, $i = 1, \dots, 6$.