

Úkol 46:

Dokažte, že následující jazyk je PSPACE-úplný: $L = \{ \langle T, w \rangle, T \text{ je TS přijímající } w, \text{ aniž by jeho čtecí hlava opustila úsek pásky, na němž je slovo } w \text{ zapsáno} \}$.

Řešení:

Abych dokázal tvrzení, musel by existovat TS přijímající zadaný jazyk L s polynomičnou paměťovou složitostí a dále by každý PSPACE-úplný jazyk musel být na tento zadaný jazyk redukovatelný (v pol. čase).

Idea:

V prvním kroku dokážu existenci TS D , který bude přijímat v pol. čase zadaný jazyk (A):

TS D rozšíří páskovou abecedu o zarážku (znak #), simuluje činnost takto modifikovaného TS a průběžně kontroluje, zda nebyla zarážka přepsána. Aby byla zachována vlastnost, že TS D bude mít polynomiální paměťovou složitost, musí simulaci provádět simultánně. Tzn, že kontrolu, zda nebyla přepsána zarážka, musí provádět po každém kroku TS T .

V druhém kroku dokážu, že libovolný jazyk z třídy PSPACE lze redukovat na náš problém v polynomičném čase (B):

Nechť L' je lib. jazyk z třídy PSPACE, pak existuje TS I , který tento přijímá L' s polynomičnou paměťovou složitostí. V polynomičném čase sestrojím TS X , který bude přijímat vstupní slovo w právě tehdy, když bude toto vstupní slovo přijímat libovolný TS I . TS X bude splňovat podmínku, aby jeho čtecí hlava neopustila úsek pásky vymezený vstupním slovem w .

A) dvoupáskový TS D pro vstupní slovo $\langle T, w \rangle$ bude vykonávat následující činnost:

1. Rozšíří páskovou abecedu TS T (který je zakódován na první pásce) o znak # (BÚNO předpokládáme, že # není z páskové abecedy TS T).
2. Na druhou pásku zapíše $w\#$.
3. Na druhé pásce simuluje činnost jednoho (dalšího) kroku TS T .
4. Kontroluje, zda není čtecí hlava simulovaného (rozšířeného) TS T nad symbolem #. Pokud ano, znamená to, že se v příštím kroku originální stroj pokusí o čtení mimo vymezený úsek pásky. TS D tedy vstupní slovo $\langle T, w \rangle$ zamítá.
5. Pokud TS T v posledním kroku vstup w zamítá (resp. přijímá), chápeme to tak, jako že TS D zamítá (resp. přijímá) také vstup $\langle T, w \rangle$.
6. Přejde znovu ke kroku 3.

Evidentně TS D přijímá takový TS, který při vstupu w neopustí hranici, kterou TS D vyznačil znakem #. Zbývá tedy ukázat, že paměťové nároky jsou nejvýše polynomiální.

TS D potřebuje ke své činnosti úseky pásky pro:

1. Vstupní slovo $\langle T, w \rangle$ (viz. univerzální TS), které bude rozšířeno o jeden páskový symbol.
2. Pro simulaci TS T , což je rovno $|w| + 1$ (pro symbol #).

Evidentně se v obou případech jedná o lineární paměťovou složitost, je tedy paměťová složitost TS D také lineární, CBD.

B) Idea:

TS C , který bude konstruovat TS splňující podmínku, bude pracovat ve dvou krocích. Nejprve simuluje činnost TS I a zjistí tak, jaký úsek pásky potřebuje. Ve druhém kroku modifikuje původní kód TS I tak, že rozšíří páskovou abecedu o znak - (speciální prázdný symbol) a za vstupní slovo w doplní takový počet těchto speciálních prázdných symbolů, aby se v tomto úseku dal simulovat výpočet původního TS. Náležitě doplní množiny stavů.

Tří-páskový TS C pro vstupní slovo $\langle I, w \rangle$ provádí následující činnost:

1. Na druhou pásku „zkopíruje“ obsah první pásky (zakódování $\langle I, w \rangle$) pro další využití.
2. TS zapsaný na druhé pásce rozšíří o páskový symbol # (BÚNO předpokládáme, že takový

symbol nebyl v páskové abecedě TS).

3. TS zapsaný na druhé pásce rozšíří takto: Pro všechny stavy q : $d(q, _) = (q', x, M)$, kde q, q' náleží Q , x je z páskové abecedy a M je z množiny $\{L, P\}$, vytvoří následující přechody: $\mathbf{d}(q, \#) = (q'', _, \mathbf{P})$, $\mathbf{d}(q'', _) = (q', \#, \mathbf{L})$. BÚNO předpokládáme, že q'' nenáležel Q , a tedy může být množina o tento stav rozšířena. Zjednodušeně tedy takto modifikovaný TS při čtení prázdného symbolu posune zarážku $\#$ o jedno políčko vpravo a vrátí se do původního stavu tak, aby TS „nic nepoznal“.
4. Na třetí pásku zapíše $w\#$.
5. Na třetí pásce simuluje činnost TS z druhé pásky.
6. Hranice $\#$ nyní vyznačila úsek pásky, který je potřebný pro výpočet. TS C nyní na začátek třetí pásky zapíše znovu vstupní slovo w a bude posouvat čtecí hlavu vpravo a přepisovat všechny znaky speciálním symbolem $-$, dokud nenarazí na „zarážku“ $\#$, tu přepíše prázdným symbolem.
7. Na třetí pásce je nyní modifikované vstupní slovo tak, aby vyhovělo podmínce, zbývá změnit původní TS (stále na první pásce), který by s takto modifikovaným vstupním slovem korektně pracoval.
8. TS zapsaný na první pásce (původní TS) rozšíří o páskový symbol $-$ (BÚNO předpokládáme, že takový symbol nebyl v páskové abecedě TS). Jedná se o „speciální prázdný symbol“.
9. TS zapsaný na první pásce (původní TS) rozšíří takto: Pro všechny stavy q : $d(q, _) = (q', x, M)$, kde q, q' náleží Q , x je z páskové abecedy a M je z množiny $\{L, P\}$, vytvoří následující přechody: $\mathbf{d}(q, -) = (q'', _, \mathbf{P})$, $\mathbf{d}(q'', _) = (q', _, \mathbf{L})$. BÚNO předpokládáme, že q'' nenáležel Q , a tedy může být množina o tento stav rozšířena. Zjednodušeně tedy takto modifikovaný TS bude postupně přepisovat „simulované“ prázdné symboly skutečnými prázdnými symboly.
10. Na první pásce máme původní TS, který je rozšířen o symbol $-$ a na třetí pásce pak vstupní slovo, se kterým by takto sestrojený TS měl pracovat, aniž by porušil danou podmínku. TS C tedy zbývá modifikovat TS na první pásce a přesunout vstupní slovo z třetí pásky do zakódování TS na první pásce.
11. Na první pásce se nalézá korektně redukováný TS s modifikovaným vstupním slovem, který vyhovuje naší podmínce.

Zbývá dokázat polynomiální časové nároky:

1. „Kopírování“ lze na TS realizovat s polynomiální časovou složitostí.
2. Rozšíření množiny symbolů má konstantní charakteristiku.
3. TS je v tomto kroku rozšířen o konečný počet přechodů – v nejvýše polynomiálním čase.
4. Zapsání řetězu na pásku má stejnou charakteristiku, jako bod 1.
5. TS I přijímá podle předpokladů slovo w s polynomiální časovou složitostí, takto modifikovaný TS zřejmě také.
6. Kroky 7. a 8. jsou analogií kroků 2. a 3. - mají stejné nároky na časovou složitost.
7. Posledním bodem TS C byla modifikace původního TS na první pásce. Při použití zakódování univerzálního stroje můžeme toto provést s polynomiální časovou složitostí.

Časová složitost je nejvýše polynomiální, CBD.