Examen CC

Problema 1

$Y - 1$ $\quad E - 1$

$X_1 - 2$ $\quad A_1 - 2$

$Z_1 - 3$ $\quad A_2 - 3$

$X_2 - 4$

$Z_2 - 5$

$X_3 - 6$

IF $X_3 \neq$ GOTO A_2 $= \langle 0, \langle 3+2, 6-1 \rangle\rangle = \langle 0, 351 \rangle = 702$

$Y \leftarrow Y + 1$ $= \langle 0, \langle 1, 0 \rangle\rangle - \langle 0, 1 \rangle = 2$

$A_1 : X_1 \leftarrow X_1 - 1$ $= \langle 2, \langle 2, 1 \rangle\rangle - \langle 2, 11 \rangle = 91$

Am folosit formula din curs $\langle a, b \rangle = 2^a(2b+1) - 1$.

Vrem să Godelizăm $[702, 2, 91]$: $2^{702} \cdot 3^2 \cdot 5^{91}$

$\cancel{2^{702} \cdot 3^2 \cdot 5^{91} =}$

Așadar, encodarea instrucțiunilor este $2^{702} \cdot 3^2 \cdot 5^{91} - 1$.

Calcule:

IF $X_3 \neq 0$ GOTO A_2

Label $= 0$

variabila $= \#(X_3) = 6$

tip $= \#(A_2) + 2 = 3 + 2 = 5$

$\Rightarrow \langle 0, \langle 5, 6-1 \rangle\rangle = \langle 0, 2^5(2\cdot 5+1)-1 \rangle = \langle 0, 351 \rangle$

$= 702$

$Y \leftarrow Y + 1$

Label 0

variabila: 1

tip: 1

$\Rightarrow \langle 0, \langle 1, 1-1 \rangle\rangle = \langle 0, \langle 1, 0 \rangle\rangle = \langle 0, 1 \rangle = 2$

A_1: $X_1 \leftarrow X_1 - 1$

Label: $\#(A_1) = 2$

variabila: $\# X_1 = 2$

tip: 2

$\Rightarrow \langle 2, \langle 2, 2-1 \rangle\rangle = \langle 2, \langle 2, 1 \rangle\rangle = \langle 2, 11 \rangle = 91$

Examen CC

Problema 2

Idee:

Folosim o MT deterministă pe două benzi.
Pe prima bandă citim inputul, fără să scriem niciodată pe ea, având așadar o MT off-line.

Pe a doua bandă ținem un contor în baza 2, pe care îl incrementăm cu câte 1 când ne deplasăm la un cuvânt mai în dreapta.
I.E. A doua bandă va conține bin(1) când îl verificăm pe w_1, bin(2) când îl verificăm pe w_2, ..., bin (n) când îl verificăm pe w_n.

Pentru comoditate, ținem numărul de pe banda 2 în reverse. De ex, la w_{10} dorim a doua bandă să arate <u>|0|1|0|1|B ...</u>.

Pașii MT:

B_1: | # w_1 # w_2 --- # $w_{\cancel{x}}$ --- # w_n

↑

B_2 | 1 | 0 | --- |

↑

$B_2 = bin(x)$ reversed

Pas 1: adăugam 1 la banda B_2, și sărim peste '#'. Dacă nu avem ~~niciun~~ '#' pe banda 1, atunci inputul nu este corect și refuzăm.

Pas 2: Dacă pe banda B_1 am dat de Blank, atunci refuzăm inputul.

Dacă cuvântul w_i pe care ne aflăm pe B_1 este egal cu numărul de pe B_2 (inversat), atunci acceptăm inputul.

Pas 3: ~~Sărim~~ Ne deplasăm pe B_1 până la '#' sau Blank în dreapta, și sărim la Pas 1, dacă avem '#', sau refuzăm dacă avem Blank.

Explicații suplimentare:

- Creșterea lui B_2 cu 1:
  - setăm cursorul la începutul benzii
  - cât timp cursorul citește 1, scriem 0 și ne deplasăm la dreapta.
  - scriem 1.

Deși poate face până la log(valoare) pași, complexitatea amortizată de incrementare de n ori este:

$$Inc = \underset{\substack{\uparrow \\ \text{bit 0 este 0}}}{\frac{1}{2} \cdot 1} + \underset{\substack{\uparrow \\ \text{bit 0 este 1} \\ \text{bit 1 este 0}}}{\frac{1}{4} \cdot 2} + \dots + \underset{\substack{\uparrow \\ \text{bit 0 este 1} \\ \vdots \\ \text{bit n-1 este 1} \\ \text{bit n este 0}}}{\frac{1}{2^n} \cdot n} + \dots$$

Prin calcul, $Inc = 2$, adică este constant.

Așadar, incrementarea cu 1 poate fi considerată $O(1)$.

- verificarea că w_i = B_2 reversed

B_1 L --- # w_{i0} w_{i1} --- $w_{i|w_i|}$ # ---

B_2 | 1 | 1 0 1 | 1 1 0 1 | --- | 1 | 1 1 0 | B ---

Dorim să efectuăm verificarea în $O(|w_i|)$.

Pentru asta, mutăm acul de citire de la începutul lui w_i la sfârșitul lui w_i. Acul de citire pe B_2 rămâne la începutul benzii.

B_1 : L --- # w_{i0} --- $w_{i|w_i|}$ ---

↑

B_2 | 0 | --- --- | B ---

↑

Verificăm în $|w_i|$ pași dacă cele două șiruri sunt egale, crescând acul pe B_2 și scăzând pe B_1, oprindu-ne la prima diferență.

Indiferent dacă acceptăm sau nu, readucem acele de pe B_1 la sfârșitul cuvântului, și de pe B_2 la începutul benzii.

Complexitate:

Fie w inputul. Așadar, $|w| \geq 2 \cdot n > n$.

Timp: - incrementarea de pe B_2 se efectuează de n ori, în $O(1)$ deci în total în $O(n)$
- parcurgerea și verificarea lui w_i se efectuează prin 2 parcurgeri al fiecărui cuvânt, deci în timp liniar în $|w|$.

Așadar, complexitatea finală este $O(|w| + n) = O(|w|)$, adică timp liniar în lungimea inputului.

Spațiu: - MT este off-line.
- pe B_2 vom avea maxim $\lceil \log_2(n) \rceil$ elemente.

Așadar, complexitatea spațiu este $O(\log(n)) \subset O(\log(|w|))$

Avem asadar $O(input)$ timp (polinomial)

$O(\log(input))$ spatiu (sub-polinomial)

De asemenea, observăm că MT nu poate cicla la infinit (se oprește pe fiecare intrare)