Homework 3 - prime

Jiří Klepl

Popište algoritmus (pro Turingův stroj), který ignoruje svůj vstup a na výstup vypisuje postupně všechna prvočísla v rostoucím pořadí.

Formální předpoklady

Ignorujeme vstup na pásce (popř. máme vstupní pásku a výstupní a tu vstupní neuvažujeme) a tedy pro tuto úlohu předpokládáme zjednodušení, že máme TS s pouze jednou páskou, která je ve vstupní konfiguraci prázdná.
Dále předpokládáme, že výstup očekáváme binárně lsb a oddělen heši v sekvenčním pořadí.

Řešení

Postupně bude na pásce čárkami oddělen konečný vzestupní seznam binárních zápisů prvočísel následován dvěma přirozenými čísly, toto také odděleno čárkami.
Postup bude následující:

Kopírovací subrutina

Očekává stroj ve stavu $mem \times copy \times init$ ukazující na první číslici kopírovaného slova.
Zanechá stroj ve stavu $mem \times copy \times end$ ukazující na první číslici kopie.

1. $(mem \times copy \times init, a) \to (mem \times copy, a \times red, N)$
2. $(mem \times copy, a \times red) \to (mem \times copy \times a \times shift, a, R)$
3. $(mem \times copy \times a \times shift, b) \to (mem \times copy \times a, b \times red, R)$
4. $(mem \times copy \times a \times shift, \#) \to (mem \times copy \times a, \#, R)$
5. $(mem \times copy \times a \times shift, \lambda) \to (mem \times copy \times a, \#, R)$
6. $(mem \times copy \times a, b) \to (mem \times copy \times a, b, R)$
7. $(mem \times copy \times a, \lambda) \to (mem \times copy \times L, a, L)$
8. $(mem \times copy \times L, a) \to (mem \times copy \times L, a, L)$
9. $(mem \times copy \times L, \#) \to (mem \times copy, \#, L)$
10. $(mem \times copy, a) \to (mem \times copy, a, L)$
11. $(mem \times copy, \#) \to (mem \times copy \times R, \#, R)$
12. $(mem \times copy, \lambda) \to (mem \times copy \times R, \lambda, R)$
13. $(mem \times copy \times R, \#) \to (mem \times copy \times end, \#, R)$
14. $(mem \times copy \times R, \lambda) \to (mem \times copy \times end, \lambda, R)$

Inkrementovací subrutina

Očekává stroj ve stavu $mem \times inc \times init$ ukazující na první číslici inkrementovaného čísla. Zanechá stroj opět na první číslici ve stavu $mem \times inc \times end$.

1. $(mem \times inc \times init, a) \to (mem \times inc, a, N)$
2. $(mem \times inc, 0) \to (mem \times inc \times L, 1, L)$
3. $(mem \times inc, \lambda) \to (mem \times inc \times L, 1, L)$
4. $(mem \times inc, 1) \to (mem \times inc, 0, R)$
5. $(mem \times inc \times L, a) \to (mem \times inc \times L, a, L)$
6. $(mem \times inc \times L, \lambda) \to (mem \times inc \times end, \lambda, R)$
7. $(mem \times inc \times L, \#) \to (mem \times inc \times end, \#, R)$

Odečítací subrutina

1. $(mem \times sub \times init, a) \to (mem \times sub \times L, a \times red, L)$
2. $(mem \times sub \times L, a) \to (mem \times sub \times L, a, L)$
3. $(mem \times sub \times L, \#) \to (mem \times sub \times L, \#, L)$
4. $(mem \times sub \times L, a \times red) \to (mem \times sub \times a \times shift, a, R)$
5. $(mem \times sub \times a \times shift, b) \to (mem \times sub \times a, b \times red, R)$
6. $(mem \times sub \times a \times shift, \#) \to (mem \times sub \times a, \#, R)$
7. $(mem \times sub \times 0, 0 \times red), carry \notin mem \to (mem \times sub \times L \times shift, 0, R)$
8. $(mem \times sub \times 0, 1 \times red), carry \notin mem \to (mem \times sub \times L \times shift, 1, R)$
9. $(mem \times sub \times 1, 0 \times red), carry \notin mem \to ((mem \cup \{carry\}) \times sub \times L \times shift, 1, R)$
10. $(mem \times sub \times 1, 1 \times red), carry \notin mem \to (mem \times sub \times L \times shift, 0, R)$
11. $(mem \times sub \times 0, 0 \times red), carry \in mem \to (mem \times sub \times L \times shift, 1, R)$
12. $(mem \times sub \times 0, 1 \times red), carry \in mem \to ((mem \setminus \{carry\}) \times sub \times L \times shift, 0, R)$
13. $(mem \times sub \times 1, 0 \times red), carry \in mem \to (mem \times sub \times L \times shift, 0, R)$
14. $(mem \times sub \times 1, 1 \times red), carry \in mem \to (mem \times sub \times L \times shift, 1, R)$
15. $(mem \times sub \times L \times shift, a), \to (mem \times sub \times L, a \times red, L)$
16. $(mem \times sub \times L \times shift, \lambda), \to (mem \times sub \times L, \lambda, L)$
17. $(mem \times sub \times L, \lambda), carry \notin mem \to (mem \times sub \times R, \lambda, R)$
18. $(mem \times sub \times R, a) \to (mem \times sub \times R, a, R)$

19. $(mem \times sub \times R, \#) \to (mem \times sub \times R, \#, R)$
20. $(mem \times sub \times R, a \times red) \to (mem \times sub \times end \times L, a, L)$
21. $(mem \times sub \times R, \lambda) \to (mem \times sub \times end \times L, \lambda, L)$
22. $(mem \times sub \times end \times L, a) \to (mem \times sub \times end \times L, a, L)$
23. $(mem \times sub \times end \times L, \#) \to (mem \times sub \times end, \#, R)$
24. $(mem \times sub \times end \times L, \lambda) \to (mem \times sub \times end, \lambda, R)$
25. $(mem \times sub \times L, \lambda), carry \in mem \to (mem \times sub \times 0, \lambda, R)$
26. $(mem \times sub \times 0, \lambda), carry \in mem \to (((mem \setminus \{carry\}) \cup \{bad\}) \times sub \times end \times L, \lambda, L)$
27. $(mem \times sub \times 1, \lambda) \to (((mem \setminus \{carry\}) \cup \{bad\}) \times sub \times end \times L, \lambda, L)$

Dělící subrutina

Operandy definovány podobně jako u odečítání, v podobném duchu i volání a návrat.

1. Označíme první číslice operandů modře (operandy rozlišujeme jako u odečítání vzájemnou polohou)
2. Pustíme odečtení levého operandu od pravého (nad abecedou, kde každý znak nahradíme modrou a nemodrou variantou)
3. pokud je v pameti bad, vycistime oznaceni operandu a vratime se na prvni cislici vysledku ve stavu $mem \times div \times end$
4. jinak, pokud je vysledek nenulovy, opakujeme krok 2
5. pokud je vysledek nulovy, vycistime oznaceni operandu a vratime se na prvni cislici vysledku ve stavu $mem \times div \times end$.

main rutina

Zde již nebudeme opakovat pomocné obarvování operandů a nebudeme ani uvádět stavy, úplně bychom se opakovali vpodstatě.

1. Na pásku napíšeme 2.
2. označíme si první prvočíslo.
3. Okopírujeme poslední číslo.
4. Inkrementujeme aktuální poslední číslo.
5. Okopírujeme aktuální poslední číslo.
6. Zkusíme poslední číslo vydělit označeným prvočíslem.
   (a) Pokud v paměti je bad, tak posuneme označení na další prvočíslo, smažeme aktuální poslední číslo, vyčistíme si paměť od zlého a

   i. pokud označení je na aktuálním posledním čísle, vyčistíme označení a vrátíme se na krok 2
   ii. jinak se vrátime na krok 5

(b) Jinak smažeme aktuální poslední číslo, označení posuneme na první prvočíslo a vrátíme se na krok 4