СЕМЕСТРАЛНИ РАД ИЗ СЛОЖЕНОСТИ АЛГОРИТАМА И ОДАБРАНИХ МЕТОДА ОПТИМИЗАЦИЈЕ И ЕЛЕМЕНАТА ДИСКРЕТНЕ МАТЕМАТИКЕ У ТЕЛЕКОМУНИКАЦИЈАМА

ТЈУРИНГОВА МАШИНА

Јелена Златановић

2018/0364

Ђорђе Антић

2018/0044

Реља Маринковић

2018/0120

14.5.2020.

Садржај

1 АПСТРАКТ

Тјурингова машина представља апстрактну машину која извршава низ инструкција на основу предефинисаног скупа стања. Она служи као модел за симулацију алгоритама које једна физичка машина (рачунар) може да изведе.

У овом раду представићемо неколико алгоритама које је могуће извести на бесконачној траци Тјурингове машине са једном главом. Алгоритми ће исписивати различите низове, конкретно:

- Фибоначијев низ
- Геометријски низ
- Низ природних бројева
- Низ простих бројева (Ератостеново сито)
- Аритметички низ
- `THUE MORSE`ов низ

Неки од ових низова изведени су са ограниченим бројем итерација, док су други неограничени, с обзиром на бесконачну траку.

Сви бројеви у низовима представљени су у унарној репрезентацији, изузев `THUE MORSE`овог низа који је по својој природи бинаран, те је и остављен у бинарној репрезентацији.

2 ЗАДАЦИ

2.1 Аритметички и геометријски низ бројева

Аритметички и геометријски низ су у својој суштини низови задати рекурентним везама - оба су одређена понављањем првог члана и одређене везе између два узастопна члана. Те везе су облика

$a_{n+1} = a_n + d$

за аритметички низ, односно

$a_{n+1} = qa_n$

за геометријски, где је n из скупа природних бројева, док је $q \neq 0$

Решење једначине аритметичког низа је:

$$a_2 - a_1 = d$$

$$...$$

$$a_n - a_{n-1} = d$$

Сабирањем претходних једначина добијамо следећу:

$$a_n = a_1 + (n-1)d$$

$$a_n = a_1 + nd - d$$

$$a_n = (a_1 - d) + nd$$

то јест, можемо писати $a_n = C + nd$ за неку константу $C = a_1 - d$

Решење једначине геометријског низа је:

$$a_n = a_1 q^{n-1}$$

$$a_n = a_1 q^{-1} q^n$$

$$a_n = \frac{a_1}{q} q^n$$

то јест, можемо писати $a_n = Kq^n$ за неку константу $K = \frac{a_1}{q}$

Посматрајмо сада општу диференцну једначину првог реда са константним коефицијентима:

$$a_{n+1} = qa_n + d \tag{1}$$

Одавде видимо да ова једначина представља аритметички низ за $q = 1$, тј. геометријски за $d = 0$.

Формираћемо нови аритметички или геометријски низ, који ће се од датог низа разликовати за константу. Покажимо то. За нови низ b_n ће важити следеће једначине:

$$a_n = b_n + k$$

$$b_{n+1} = qb_n$$

Из једначине 1 следи

$$b_{n+1} + k = q(b_n + k) + d$$
$$qb_n + k = qb_n + qk + d$$
$$k = \frac{d}{1-q}$$

тада, видимо да је $b_n = a_n - \dfrac{d}{1-q}$ геометријски низ, чије опште решење знамо да је $b_n = Kq^n$.

Одавде имамо

$$Kq^n = a_n - \frac{d}{1-q}, q \neq 1$$

Одакле следи

$$a_n = Kq^n + \frac{d}{1-q}, q \neq 1$$

уз

$$a_n = K + nd$$

што представља аритметички низ.

2.1.1 Аритметички низ

Задатак: Направити програм за Тјурингову машину који исписује аритметички низ.

Аритметички низ представља низ бројева такав да је разлика свака два суседна члана константна, и то по формули:
$a_{n+1} = a_n + d$

Збир првих n чланова низа рачуна се по формули:

$$S_n = \sum_{1}^{n} a_i = \frac{n}{2}(2 \cdot a_1 + (n-1) \cdot d) = \frac{n}{2} \cdot (a_1 + a_n)$$

Име је добио по особини да је сваки члан низа (осим првог) аритметичка средина њему претходног и следећег члана у низу:

$$a_n = \frac{a_{n-1} + a_{n+1}}{2}$$

У задатку је узето да је почетни члан 1, а да је разлика суседних чланова једнака 2. Пример низа је:

1 3 5 7 9 11 13 15 ...

Идеја код овог програма је да корисник сам унесе колико чланова низа жели да буде исписано, уношењем одговарајућег броја посебних симбола, како би се извршавање ограничило на коначан број итерација. У суштини, почетни део траке понаша се као бројач. Ово је могло и да се изведе на другачије начине, на пример да корисник унесе број итерација у другом формату, попут децималне или бинарне представе броја, и да се потом одбројава, али би ово много компликовало програм за мало побољшање у интерфејсу. Још један начин да се ограничи низ био би да се на почетној траци после извесног броја бланко знакова постави посебан знак који служи као сигнал да се програм заврши када глава наиђе на њега. Међутим, у овој верзији генератора аритметичког низа коришћен је првобитно описани метод, не толико због оптимизације рада програма, већ у покушају да се поједностави сам код.

Опет једноставности ради, бројеви биће приказани у својој унарној представи, и одвојени бланко знаком.

Азбука: Машина ће користити азбуку $S = \{0, 1, g, x, t, b\}$. Њихова функција је следећа:

- 1 - Бројеви на траци (чланови низа) ће бити представљени унарним записом, односно, број симбола $'1'$ одговараће броју који је представљен.
- g - Знак g користиће се за ограничавање низа, тј број знакова g означаваће број итерација програма, а самим тим и број чланова низа који ће бити исписани на траци по завршетку извршавања програма.
- b - Бланко знак. Чланови низа ће бити одвојени бланко знаком (b).
- Остали знаци су помоћни, и њихова функција је описана у одељку *Поступак.*

Почетна и крајња трака: Као што је већ речено, у почетној траци корисник дефинише број чланова низа, тако да је исправан формат почетне траке за, на пример, четири итерације следећи:

```
gggg
^
```

где се лево и десно од скупа симбола g налазе бланко карактери. Позиција главе означена је знаком ^.

По извршавању програма, трака ће имати следећи изглед:

```
bbbbbb1b111b11111b1111111b
                         ^
```

што се чита као 1 3 5 7.

Поступак и имплементација: Код овакве имплементације, при свакој итерацији се по један симбол g мења помоћним симболом 0, као ознака да је једна итерација одрађена.

Уписивање првог члана

Прво што прогам уради је уписивање симбола 0 уместо првог симбола g, и уписивање првог броја (1) десно од групе симбола g и једног бланко знака. Уз то, десно од првог знака 1 уписује се симбол t као ознака да је ту крај последње уписаног члана.

Провера

Програм се враћа до групе симбола g, и, ако не наиђе на знак 0 непосредно лево од бланко знака, уместо другог знака g уписује 0. У супротном, програм се завршава.

Уписивање сваког наредног члана

Глава се помера до симбола t,и иза њега уписује два сибмола 1.

Затим, глава се помера у улево до симбола t, и програм копира све симболе 1 између t и првог бланко знака лево од t на места десно од t (како би знао докле је стигао са копирањем, уместо сваког ископираног знака 1 привремено се уписује x).

Потом, када су све јединице замењене са x, глава се помера у десно, мењајући успут све привремене симболе x у 1, и додаје се један симбол t десно од свих ископираних симбола 1 (тј на уместо првог бланко знака).

Најзад, глава се помера улево, успут мењајући стари симбол t у b све док не наиђе или на симбол 0, када се завршава програм, или на симбол g, када се још једно g мења на 0, и процес додавања још једног члана започиње.

Завршавање програма

На самом крају, пре одласка у завршно стање програм помера главу у лево, мењајући све 0 у b, а затим у десно, мењајући последње t у b

Изглед траке кроз неколико корака

```
ggg
0gg
0ggb1t
00gb1t
00gb1t11
00gbxt11
00gbxt111
00gb1t111t
00gb1b111t
000b1b111t11
000b1b11xt111
000b1b1xxt1111
000b1bxxxt11111
000b1b111t11111t
000b1b111b11111t
bbbb1b111b11111b
```

Кратак опис стања

q0 - почетно стање. Одмах мења једно g у 0, и мења на стање 1.
q1 - оставља g док не наиђе на b, и мења на стање 2.
q2 - уписује 1 после првог b и мења на стање 3.
q3 - уписује t десно од уписаног 1 (обележава крај последњег места уписивања) и мења на стање 4.
q4 - иде лево док не наиђе на g,успут исправљајући t у b. Ако наиђе на 0, то значи да није више остало g, и одлази у стање 20, а ако наиђе на g, одлази у стање 5.
q5 - иде лево док не наиђе на 0, оставља је, помера једно место у десно и иде у стање 6.
q6 - мења g на 0 (да обележи нову итерацију), и мења на стање 7.
q7 - иде десно док не наиђе на t, и мења на стање 8.
q8 - уписује 1 после t и иде у стање 9.
q9 - уписује 1 поново и иде у стање 10.
q10 - помера улево док не наиђе на t, и мења на стање 11.
q11 - иде даље лево док не наиђе на 1 (прескаче све x). Ако наиђе на 1 уместо њега уписује x и мења на стање 12. Ако наиђе на b, то значи да су сви знаци 1 ископирани, и мења на стање 13.
q12 - иде десно док не наиђе на b, уписује 1 и враћа на стање 10.
q13 - иде десно, исправља све x у 1. Када наиђе на t, њега мења у b, и мења у стање 4, чиме се једна итерација завршава.
q20,q21 - завршни кораци за форматирање, уклањају привремене симболе $x, t, 0$

Инструкције

$f(q_0, g) = (q_1, 0, +1)$

$f(q_1, g) = (q_1, g, +1)$
$f(q_1, b) = (q_2, b, +1)$

$f(q_2, b) = (q_3, 1, +1)$

$f(q_3, b) = (q_4, t, -1)$

$f(q_4, 1) = (q_4, 1, -1)$
$f(q_4, b) = (q_4, b, -1)$
$f(q_4, t) = (q_4, b, -1)$
$f(q_4, g) = (q_5, g, -1)$
$f(q_4, 0) = (q_{20}, 0, -1)$

$f(q_5, g) = (q_5, g, -1)$
$f(q_5, 0) = (q_6, 0, +1)$

$f(q_6, g) = (q_7, 0, +1)$

$f(q_7, 1) = (q_7, 1, +1)$
$f(q_7, g) = (q_7, g, +1)$
$f(q_7, x) = (q_7, x, +1)$
$f(q_7, b) = (q_7, b, +1)$
$f(q_7, t) = (q_8, t, +1)$

$f(q_8, b) = (q_9, 1, +1)$

$f(q_9, b) = (q_{10}, 1, -1)$

$f(q_{10}, 1) = (q_{10}, 1, -1)$
$f(q_{10}, t) = (q_{11}, t, -1)$

$f(q_{11}, x) = (q_{11}, x, -1)$
$f(q_{11}, 1) = (q_{12}, x, +1)$
$f(q_{11}, b) = (q_{13}, b, +1)$

$f(q_{12}, t) = (q_{12}, t, +1)$
$f(q_{12}, x) = (q_{12}, x, +1)$
$f(q_{12}, 1) = (q_{12}, 1, +1)$
$f(q_{12}, b) = (q_{10}, 1, -1)$

$f(q_{13}, x) = (q_{13}, 1, +1)$
$f(q_{13}, t) = (q_{13}, t, +1)$
$f(q_{13}, 1) = (q_{13}, 1, +1)$
$f(q_{13}, b) = (q_4, t, -1)$

$f(q_{20}, 0) = (q_{20}, b, -1)$
$f(q_{20}, b) = (q_{21}, b, +1)$

$f(q_{21}, b) = (q_{21}, b, +1)$
$f(q_{21}, 1) = (q_{21}, 1, +1)$
$f(q_{21}, t) = (q_+, b, -1)$

2.1.2 Геометријски низ

Геометријска прогресија је математички израз за принцип по коме се формира низ код кога је однос једног члана са његовим претходним чланом константан.

Овакав низ се зове геометријски низ и може се дефинисати својим првим чланом a_1 и поменутим односом q.

$$q = \frac{a_n}{a_{n-1}}$$

Притом низ за одређене вредности q има следећа својства:

- $q > 1$ — тежи бесконачности
- $q = 1$ — низ у коме су сви елементи међусобно једнаки
- $0 < |q| < 1$ — низ који тежи нули
- $q \leq -1$ — неодређено дивергентан низ

Било који члан низа тражи се по формули:

$$a_n = a_1 \cdot q^{n-1}$$

Збир првих n чланова низа тражи се по формули:

$$S_n = \sum_{1}^{n} a_i = a_i \cdot \frac{q^{n-1}}{q-1}$$

А ако је |q|<1 тада је могуће одредити збир бесконачно много чланова низа:

$$S_\infty = \sum_{1}^{\infty} a_i = \frac{a_1}{1-q}$$

Име је добио по особини да је сваки члан низа (осим првог) геометријска средина њему претходног и следећег члана у низу:

$$a_n = \sqrt{a_{n-1} \cdot a_{n+1}}$$

Задатак: Написати програм за Тјурингову машину која исписује геометријски низ за задати почетни члан и количник.

Почетна трака и азбука: За решавање ћемо користити азбуку $S = \{b, 1, x\}$. Бројеве ћемо представљати унарним записом, b је празан симбол, док је x помоћни симбол. На траци се налази бесконачно много симбола b.

Поступак и код: За почетни члан бирамо број 2 који ће на траци бити записан као 11, а за количник број 3, дакле $a_1 = 2$, $q = 3$.

$f(q_0, b) = (q_1, 1, +1)$
$f(q_1, b) = (q_2, 1, +1)$
$f(q_2, b) = (q_3, b, -1)$

Претходна 3 корака исписују 11 на траци и трака изгледа овако:

$\dots bbb11bbbbbb\dots$

Уколико желимо да неки други број буде почетни члан исписујемо га на исти начин само повећавамо број стања.

$f(q_3, 1) = (q_4, x, +1)$

Овим кораком последњу јединицу замењујемо помоћним симболом x.

$f(q_3, x) = (q_3, x, -1)$
$f(q_3, b) = (q_9, b, +1)$
$f(q_4, b) = (q_5, b, +1)$
$f(q_4, x) = (q_4, x, +1)$
$f(q_5, b) = (q_6, 1, +1)$
$f(q_5, 1) = (q_5, 1, +1)$
$f(q_6, b) = (q_7, 1, +1)$
$f(q_7, b) = (q_8, 1, -1)$

Идеја је да сваку јединицу која је замењена симболом x 3 пута „копирамо" тако што након првог празног симбола b допишемо 3 јединице. На тај начин за било који количник правимо следећи члан у низу, тј. уколико желимо да количник буде било који други број, рецимо 5, за сваку јединицу почетног члана или претходног члана у низу пишемо 5 јединица. То проузрокује већи број стања.

$f(q_8, b) = (q_3, b, -1)$

Овим кораком се враћамо на претходни број и настављамо „копирање" уколико имамо слободних јединица тако што их заменимо са x.

$f(q_8, 1) = (q_8, 1, -1)$
$f(q_9, x) = (q_9, 1, +1)$
$f(q_9, b) = (q_{10}, b, +1)$
$f(q_{10}, 1) = (q_{10}, 1, +1)$
$f(q_{10}, b) = (q_3, b, -1)$

Уколико више нема јединица за „копирање" , знамо да смо направили нови члан низа и кораком $f(q_3, b) = (q_9, b, +1)$ *идемо даље и прештампавамо све помоћне симболе x јединицама (* $f(q_9, x) = (q_9, 1, +1)$ *).*

Након тога прелазимо на прављење следећег члана у низу.

2.2 Фибоначијев низ

Фибоначијев низ представља низ бројева у коме збир претходна два броја у низу даје вредност наредног члана низа. Индексирање чланова овог низа почиње од нуле а прва два члана су му 0 и 1.

$f_0 = 0$

$f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$

Можемо приметити да је Фибоначијев низ врста рекурзивног низа. Дакле за n=0, 1, 2, 3,... низ би изгледао овако: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233... Понекад се узима да почиње са $F_1 = 1$.

До овог открића дошао је италијански математичар из Пизе, Леонардо Фибоначи (Leonardo Fibonacci, 1170-1250), посматрајући размножавање зечева у природи. Овај проблем је изложио у својој књизи Liber Abaci (Књига о абакусу или Књига рачуна) која је написана 1202, а ревидирана и значајно допуњена 1228. године.

Питање је гласило: *"Колико парова зечева ће репродуковати један пар за годину дана ако се претпостави да сваког месеца један пар роди нови пар који за два месеца постане репродуктиван?"*

Проучавајући размножавање зечева добио је низ бројева на следећи начин: првог месеца експеримент почиње једним паром зечева, у другом месецу ће постојати само тај један пар, у трећем месецу ће их бити 2, у четвртом 3, у петом 5, у шестом 8, у седмом 13, у осмом 21, у деветом 34, у десетом 55 у јеаданаестом 89 и у дванаестом 144.

Међутим делећи сваки број у низу са претходним, дошао је до следећег односа:

$1/1 = 1$
$2/1 = 2$
$3/2 = 1,5$
$5/3 = 1,66...$
$8/5 = 1,6$
$13/8 = 1,625$
$21/13 = 1,615384$
$34/21 = 1,619047$
$55/34 = 1,617647$
$89/55 = 1,618181$
...

Као што видимо однос 2 сукцесивна броја у Фибоначијевом низу тежи броју φ („фи“) познатом као „златни пресек (златни однос)“ како n тежи бесконачно.

Две величине су у златном односу, ако је њихов однос једнак односу суме те две вредности наспрам веће вредности.
За $a > b > 0$:

$\frac{a+b}{a} = \frac{a}{b} = \varphi$

$\frac{a+b}{a} = 1 + \frac{b}{a} = 1 + \frac{1}{\varphi} = \varphi$

$\varphi + 1 = \varphi^2$

$\varphi^2 - \varphi - 1 = 0$

Израчунавањем квадратне једначине добијамо:

$$\varphi = \frac{1+\sqrt{5}}{2} = 1.6180339887...$$

и

$$\varphi = \frac{1-\sqrt{5}}{2} = -0.6180339887...$$

Зато што је φ однос између две позитивне вредности, φ је увек позитивна вредност:

$$\varphi = \frac{1+\sqrt{5}}{2} = 1.6180339887...$$

Такође можемо посматрати једначину $f_n = f_{n-1} + f_{n-2}$ као хомогену линеарну диференцну једначину другог реда и њено решење добити решавањем одговарајуће карактеристичне једначине $\varphi^2 - \varphi - 1 = 0$. Почетни услови су $f(0) = 0$ и $f(1) = 1$. Корени карактеристичне једначине су $\varphi_{1,2} = \frac{1\pm\sqrt{5}}{2}$. Заменом почетних услова $f(0) = C_1 + C_2 = 0$ и $f(1) = \frac{1}{2}(C_1 + C_2 + (C_1 - C_2)\sqrt{5}) = 1$ добијамо систем од 2 једначине са 2 непознате:

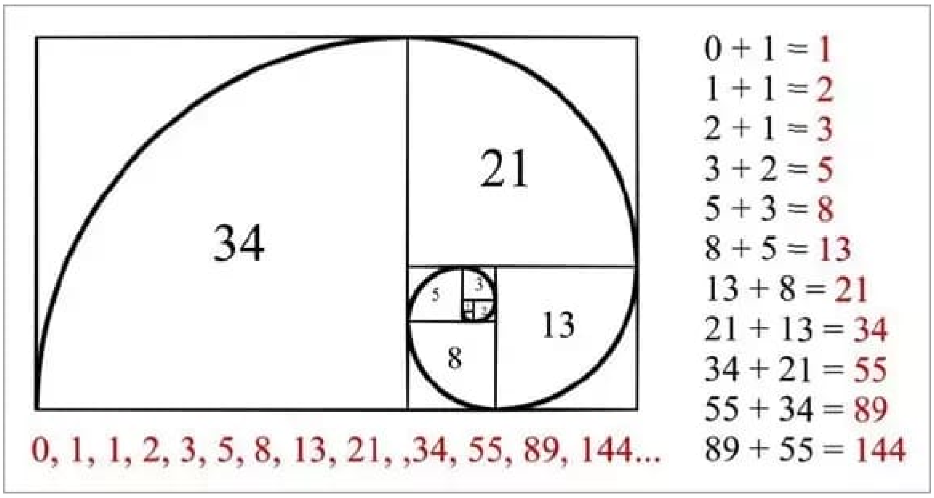
$C_1 + C_2 = 0$ и $C_1 - C_2 = 2\frac{\sqrt{5}}{5}$

Одатле следи $C_1 = \frac{1}{\sqrt{5}}$ и $C_2 = -\frac{1}{\sqrt{5}}$

Коначно, заменом свих вредности добијамо формулу за израчунавање n-тог члана низа.

$$f_n = \frac{\frac{(1+\sqrt{5})^n - (1-\sqrt{5})^n}{}}{2^n\sqrt{5}}$$

Фибоначијев низ можемо представити графички почевши са два квадрата дужине странице 1 постављеним један до другог. Затим, изнад њих квадрат дужине странице 2 (1+1=2), а поред тако добијеног правоугаоника (1+2) квадрат дужине странице 3. Ако наставимо са додавањем нових квадрата на слици, тако да је страница новог квадрата једнака збиру дужина страница претходна два, добићемо квадрате чије су дужине страница једнаке бројевима Фибоначијевог низа. А када у тако добијене квадрате уцртамо кружне лукове, добићемо спиралу која се у том обику најчешће појављује у природи, као и у уметности.



У природи се златни пресек јавља у цветовима сунцокрета, распоређивању листова око стабла, броју спирала на шишаркама, односу броја пчела и трутова у кошници, пропорцијама човековог тела, броју латица цветова којих најчешће има 3, 5, 8, 13, 21, 34, 55 или 89. Шкољка пужа Наутилус је један од најсавршенијих облика у природи. Она је у облику спирале чији су саставни делови квадрати, сви дужине неког од чланова Фибоначијевог низа.

Задатак: Написати програм за Тјурингову машину којим се исписује Фибоначијев низ.

Почетна трака и азбука: За решавање ћемо користити азбуку $S = \{b, 1, x, y\}$. Бројеве ћемо представљати унарним записом, b је празан символ, док су x и y помоћни симболи. На траци се налази бесконачно много симбола b.

Поступак и код: Желимо да испишемо низ бројева 1, 1, 2, 3, 5, 8, 13... у облику $1b1b11b111b11111b11111111b1111111111111b\ldots$ Идеја је да наредни члан добијемо сабирањем претходна 2 тако што крећемо од последњег броја, дођемо до празног места између последња 2 броја и прештампамо га са x, затим на крај другог броја и редом јединицу прештампамо са y и „копирамо“ је на место нашег новог броја, и тако уназад, прештампамо јединицу са x па је додамо на наш нови број, све док у прештампавању не наиђемо на b, што је знак да смо прекопирали све јединице са оба броја, тачније сабрали претходна 2 броја. Након тога прештампавамо све y са 1 и x са b. Исти процес понављамо за следећи број у низу.

$f(q_0, b) = (q_1, 1, +1)$
$f(q_1, b) = (q_2, b, +1)$
$f(q_2, b) = (q_3, 1, +1)$
$f(q_3, b) = (q_4, b, -1)$

Претходна 4 корака исписују прва 2 члана низа и трака изгледа овако:

$\ldots b1b1bbbbbb\ldots$

$f(q_4, b) = (q_5, x, +1)$

Празан симбол b прештампамо са x како бисмо означили место између 2 броја која же-

лимо да саберемо.

$$f(q_4, 1) = (q_4, 1, -1)$$
$$f(q_5, b) = (q_6, b, -1)$$

Крај другог броја.

$$f(q_5, 1) = (q_5, 1, +1)$$
$$f(q_6, y) = (q_6, y, -1)$$
$$f(q_6, 1) = (q_7, y, +1)$$

Прештампамо јединицу са y.

$$f(q_6, x) = (q_6, x, -1)$$

Прелазак на прештампавање и копирање јединица првог броја.

$$f(q_6, b) = (q_{10}, b, +1)$$

Наилазак на b значи да немамо више јединица за копирање, тј. да смо добили број који је збир претходна 2.

$$f(q_7, b) = (q_8, b, +1)$$

Прелазак на формирање новог броја.

$$f(q_7, y) = (q_7, y, +1)$$
$$f(q_7, x) = (q_7, x, +1)$$
$$f(q_8, b) = (q_9, 1, -1)$$

Формирање новог броја „копирањем“ јединице коју смо претходно прештампали са y.

$$f(q_8, 1) = (q_8, 1, +1)$$
$$f(q_9, b) = (q_6, b, -1)$$

Понављање претходних корака како бисмо копирали све јединице претходна 2 броја, до наилажења на b.

$$f(q_9, 1) = (q_9, 1, -1)$$
$$f(q_{10}, x) = (q_{10}, b, +1)$$
$$f(q_{10}, y) = (q_{10}, 1, +1)$$

Претходна 2 корака прештампавају y са 1 и x са b.

$$f(q_{10}, b) = (q_{11}, b, +1)$$
$$f(q_{11}, 1) = (q_{11}, 1, +1)$$
$$f(q_{11}, b) = (q_4, b, -1)$$

Прелазак на формирање следећег члана у низу.

2.3 Низ природних бројева

Задатак: Направити програм за Тјурингову машину који исписује низ природних бројева.

Почетна трака и азбука: За решавање ћемо користити азбуку $S = \{b, 1, x, u\}$. Бројеве ћемо представљати унарним записом, b је празан симбол, док су x и u помоћни симболи. Почетна бесконачна трака је попуњена знаковима b.

Поступак: Прво уписујемо јединицу, након тога улазимо у петљу која преписује претходни број, након чега на новонастали број додаје још једну јединицу, чиме се добија број за један већи од претходног што и представља низ природних бројева.

Овако извршен програм би се одвијао бесконачно дуго, исписујући све природне бројеве. Ако желимо да се код заврши након неког броја, то можемо да урадимо на више начина, као на пример да поставимо помоћну променљиву u на почетну траку на одређено место. То место можемо да одредимо бројем места између почетне главе и u. Ако желимо да испишемо само број 1, трака би изгледала овако bu (глава је на b). За 2 би постојао размак од 3 празна симбола између почетне главе и u, за број 3 би било 7 празних симбола итд. Ако узмемо да је број до кога желимо да испишемо низ n, тада можемо искористити следећу формулу за рачунање размака између почетне главе и u:

$$\frac{1}{2}(n-1)(n+4)$$

У том случају додајемо линију $f(q_1, u) = (q_+, b, +1)$ којом ће програм бити завршен.

Код:

$f(q_0, b) = (q_1, 1, +1)$
$f(q_1, b) = (q_2, b, -1)$
$f(q_1, 1) = (q_1, 1, -1)$
$f(q_1, u) = (q_+, b, +1)$
$f(q_2, b) = (q_5, b, +1)$
$f(q_2, 1) = (q_3, x, +1)$
$f(q_2, x) = (q_2, x, -1)$
$f(q_3, b) = (q_4, b, +1)$
$f(q_3, x) = (q_3, x, +1)$
$f(q_4, b) = (q_1, 1, -1)$
$f(q_4, 1) = (q_4, 1, +1)$
$f(q_5, b) = (q_6, b, +1)$
$f(q_5, x) = (q_5, 1, +1)$
$f(q_6, b) = (q_1, 1, +1)$
$f(q_6, 1) = (q_6, 1, +1)$

2.4 Прости бројеви и Ератостеново сито

Прост број је сваки број већи од 1 који је дељив једино јединицом и самим собом односно није производ 2 мања природна броја.

Ако нам је дат низ природних бројева до n и од нас се захтева да издвојимо све просте бројеве, можемо употребити неко од простих сита од којих су најпознатија: Ератостеново сито, Сундарамово сито, Аткинсоново сито и разна кружна сита. Сложеност показује да је међу њима Ератостеново сито спорије са сложеношћу $O(n(loglogn))$, док је сложеност за Аткинсоново сито и кружна сита линеарно односно $O(n)$. С друге стране, Ератостеново сито је најлакше за имплементацију.

Алгоритам Ератостеновог сита:

1. Прво исписујемо све природне бројеве од 2 до n.
2. Затим променљивој t додељујемо вредност најмањег броја у низу односно $t = 2$.
3. Прецртавамо све бројеве дељиве са t сем самог броја t.
4. Када стигнемо до краја низа, враћамо се на почетак и тражимо први следећи број већи од t, а да није прецртан. Када нађемо такав број, његову вредност додељујемо променљивој t и понављамо од корака 3. У случају да такав број не постоји, алгоритам се завршава.
5. Када се алгоритам заврши сви непрецртани бројеви представљају просте бројеве.

Пример: Нека је $n = 15$. Прво исписујемо наш низ:

2 3 4 5 6 7 8 9 10 11 12 13 14 15

Узимамо да је $t = 2$ и прецртавамо све бројеве дељиве са 2 сем 2:

2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15

Затим се враћамо на почетак низа и тражимо непрецртани број већи од 2, што је у овом случају $t = 3$ и прецртавамо бројеве дељиве са 3:

2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~

Следећа итерација је за $t = 5$. Примећујемо да ниједан број нећемо прецртати тако да се алгоритам завршава и добијамо низ простих бројева до 15:

2 3 5 7 11 13

Имплементација

Задатак: Написати програм за Тјурингову машину који исписује све просте бројеве мање од n.

Различите имплементације: Постоји више типова Тјурингове машине, што омогућава и више различитих имплементација Ератостеновог сита како бисмо пронашли све просте у бројеве у неком низу.

Рецимо код Тјурингове машине са више трака бисмо могли да на једној траци испишемо наш низ, док на другој траци записујемо редом непрецртане бројеве, односно оне бројеве које смо у претходном примеру обележавали са t и помоћу њих проверамо остале бројеве на првој траци. Тада бисмо и на првој и на другој траци добили низ простих бројева, осим што би на другој траци тај низ био уређенији, односно са мање празних знакова између суседних простих бројева.

Ако бисмо користили Тјурингову машину са једном траком али више глава, процес би био сличан. Након што смо исписали наш низ на траку, једна глава би пронашла први непрецртани број и то би било t из примера, док би друга глава ишла кроз низ и уз помоћ прве главе проверавала да ли је дати број дељив бројем на који показује прва глава. Након што друга глава прође кроз цео низ, прва глава би пронашла следећи непрецртани број, а друга глава опет прошла кроз низ итд.

Ми ћемо користити бесконачну траку са једном главом као што смо радили и у осталим задацима.

Почетна трака и азбука: Азбука коју ћемо користити је $S = \{b, 1, l, m, p, r, n, k, s, t, u\}$. Бројеве ћемо представљати унарним записом, b је празан знак, а остали знакови ће бити помоћни знакови. Почетна трака ће бити иста као у задатку за природне бројеве. Почетна бесконачна трака је попуњена знаковима b и на њој се налази знак u на одговарајућој удаљености од почетне главе.

Поступак: На почетку имамо поптуно исти алгоритам као за исписивање природних бројева. Када стигнемо до u враћамо се на почетак низа и бришемо јединицу јер она свакако није прост број. Померамо главу десно и стижемо до прве јединице броја 2. Одавде крећемо са провером бројева нашег низа да ли су прости, примењујући Ератостеново сито. Ератостеново сито ћемо применити на сваки број посебно. Алгоритам је најбоље приказати на примеру. Рецимо да проверамо да ли је 11 прост.

b b b 1 1 1 1 1 1 1 1 1 1 1 b b b

За почетак ћемо прву јединицу заменити са p, а последњу са k.

b b b p 1 1 1 1 1 1 1 1 1 k b b b

Након тога се враћамо на почетак броја и померамо главу удесно док не наиђемо на прву јединицу. Када је пронађемо, уместо ње уписујемо m. Све p пре m замењујемо са r.

b b b r m 1 1 1 1 1 1 1 1 k b b b

Слично претходном примеру, обележили смо први прост број и сада можемо да обришемо све јединице које се налазе на месту његових садржалаца. Разлика је у томе што овде бришемо и тај прост број. То прво радимо тако што уместо m уписујемо n. Сада треба све r, l и n да померимо удесно за онолико места колико укупно има r, l и n. r и n враћамо у p, а l у јединицу у току пребацивања. На новим местима p пребацујемо у r, јединицу у l, а k у s у случају да пребацујемо r и l. Ако пребацујемо n, у случају да наиђемо на јединицу мењамо је са m, ако наиђемо на p мењамо у n, а k у t.

$b\ b\ b\ r\ n$ 1 1 1 1 1 1 1 1 $k\ b\ b\ b$

$b\ b\ b\ p\ p\ l\ m$ 1 1 1 1 1 1 $k\ b\ b\ b$

$b\ b\ b\ p\ p\ l\ n\ 1\ 1\ 1\ 1\ 1\ 1\ k\ b\ b\ b$

$b\ b\ b\ p\ p\ 1\ p\ l\ m\ 1\ 1\ 1\ 1\ k\ b\ b\ b$

...

$b\ b\ b\ p\ p\ 1\ p\ 1\ p\ 1\ p\ l\ m\ k\ b\ b\ b$

$b\ b\ b\ p\ p\ 1\ p\ 1\ p\ 1\ p\ l\ n\ k\ b\ b\ b$

$b\ b\ b\ p\ p\ 1\ p\ 1\ p\ 1\ p\ 1\ n\ s\ b\ b\ b$

У случају да k заменимо са t, значи да је број који проверамо дељив бројем m, па све бришемо, односно попуњавамо празним знаковима b и прелазимо на следећи број. Међутим, ако смо k заминили са s, показали смо да дати број није дељив бројем m, као ни другим садржаоцима броја m. Враћамо се на почетак низа, а успут s враћамо у k, све r и n у p, а све l у јединице. Опет тражимо прву јединицу и понављамо претходно. У случају да у потрази за јединицом не наиђемо ни на једну, већ стигнемо до k, показали смо да је дати број прост и можемо да пређемо на следећи број.

$b\ b\ b\ p\ p\ 1\ p\ 1\ p\ 1\ p\ 1\ p\ k\ b\ b\ b$

$b\ b\ b\ r\ r\ m\ p\ 1\ p\ 1\ p\ 1\ p\ k\ b\ b\ b$

$b\ b\ b\ r\ r\ n\ p\ 1\ p\ 1\ p\ 1\ p\ k\ b\ b\ b$

...

$b\ b\ b\ p\ p\ p\ p\ 1\ p\ l\ r\ m\ p\ k\ b\ b\ b$

$b\ b\ b\ p\ p\ p\ p\ 1\ p\ l\ r\ n\ p\ k\ b\ b\ b$

$b\ b\ b\ p\ p\ p\ p\ 1\ p\ 1\ p\ n\ r\ s\ b\ b\ b$

$b\ b\ b\ p\ p\ p\ p\ 1\ p\ 1\ p\ p\ p\ k\ b\ b\ b$

...

$b\ b\ b\ p\ p\ p\ p\ p\ p\ p\ p\ p\ p\ p\ k\ b\ b\ b$

Показали смо да 11 јесте прост број и сада враћамо све у јединице и након тога можемо прећи на следећи број.

$b\ b\ b\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ b\ b\ b$

Код:

$f(q_0, b) = (q_1, 1, +1)$
$f(q_1, b) = (q_2, b, -1)$
$f(q_1, 1) = (q_1, 1, -1)$
$f(q_1, u) = (q_{29}, b, +1)$
$f(q_2, b) = (q_5, b, +1)$
$f(q_2, 1) = (q_3, x, +1)$
$f(q_2, x) = (q_2, x, -1)$
$f(q_3, b) = (q_4, b, +1)$
$f(q_3, x) = (q_3, x, +1)$
$f(q_4, b) = (q_1, 1, -1)$
$f(q_4, 1) = (q_4, 1, +1)$
$f(q_5, b) = (q_6, b, +1)$
$f(q_5, x) = (q_5, 1, +1)$
$f(q_6, b) = (q_1, 1, +1)$
$f(q_6, 1) = (q_6, 1, +1)$
$f(q_7, 1) = (q_7, 1, -1)$
$f(q_7, b) = (q_8, b, -1)$
$f(q_8, 1) = (q_7, 1, -1)$
$f(q_8, b) = (q_9, b, +1)$
$f(q_9, b) = (q_{10}, b, +1)$
$f(q_{10}, 1) = (q_{11}, b, +1)$
$f(q_{11}, b) = (q_{12}, b, +1)$
$f(q_{12}, 1) = (q_{13}, p, +1)$
$f(q_{12}, b) = (q_{12}, b, +1)$
$f(q_{12}, u) = (q_+, b, +1)$
$f(q_{13}, 1) = (q_{13}, 1, +1)$
$f(q_{13}, b) = (q_{14}, b, -1)$
$f(q_{14}, 1) = (q_{15}, k, -1)$
$f(q_{15}, 1) = (q_{15}, 1, -1)$
$f(q_{15}, p) = (q_{15}, p, -1)$
$f(q_{15}, b) = (q_{16}, b, +1)$
$f(q_{16}, p) = (q_{16}, p, +1)$
$f(q_{16}, 1) = (q_{17}, m, -1)$
$f(q_{16}, k) = (q_{26}, 1, -1)$
$f(q_{17}, p) = (q_{17}, r, -1)$
$f(q_{17}, b) = (q_{18}, b, +1)$
$f(q_{18}, r) = (q_{18}, r, +1)$
$f(q_{18}, m) = (q_{19}, n, -1)$
$f(q_{19}, r) = (q_{19}, r, -1)$
$f(q_{19}, b) = (q_{20}, b, +1)$
$f(q_{19}, l) = (q_{19}, l, -1)$
$f(q_{19}, n) = (q_{19}, n, -1)$
$f(q_{19}, p) = (q_{20}, p, +1)$
$f(q_{19}, 1) = (q_{20}, 1, +1)$
$f(q_{20}, r) = (q_{21}, p, +1)$
$f(q_{20}, l) = (q_{21}, 1, +1)$
$f(q_{20}, n) = (q_{22}, p, +1)$
$f(q_{21}, r) = (q_{21}, r, +1)$
$f(q_{21}, n) = (q_{21}, n, +1)$
$f(q_{21}, p) = (q_{19}, r, -1)$
$f(q_{21}, 1) = (q_{19}, l, -1)$
$f(q_{21}, k) = (q_{19}, s, -1)$
$f(q_{21}, l) = (q_{21}, l, +1)$
$f(q_{21}, s) = (q_{25}, k, -1)$
$f(q_{22}, r) = (q_{22}, r, +1)$
$f(q_{22}, p) = (q_{23}, n, -1)$
$f(q_{22}, 1) = (q_{23}, m, -1)$
$f(q_{22}, l) = (q_{22}, l, +1)$
$f(q_{22}, s) = (q_{25}, k, -1)$
$f(q_{22}, k) = (q_{23}, t, -1)$
$f(q_{23}, l) = (q_{24}, l, +1)$
$f(q_{23}, r) = (q_{24}, r, +1)$
$f(q_{24}, m) = (q_{19}, n, -1)$
$f(q_{24}, n) = (q_{19}, n, -1)$
$f(q_{24}, t) = (q_{28}, b, -1)$
$f(q_{25}, b) = (q_{16}, b, +1)$
$f(q_{25}, 1) = (q_{25}, 1, -1)$
$f(q_{25}, p) = (q_{25}, p, -1)$
$f(q_{25}, r) = (q_{25}, p, -1)$
$f(q_{25}, l) = (q_{25}, 1, -1)$
$f(q_{25}, n) = (q_{25}, p, -1)$
$f(q_{26}, p) = (q_{26}, 1, -1)$
$f(q_{26}, b) = (q_{27}, b, +1)$
$f(q_{27}, 1) = (q_{27}, 1, +1)$
$f(q_{27}, b) = (q_{12}, b, +1)$
$f(q_{28}, b) = (q_{12}, b, +1)$
$f(q_{28}, 1) = (q_{28}, b, -1)$
$f(q_{28}, p) = (q_{28}, b, -1)$
$f(q_{28}, r) = (q_{28}, b, -1)$
$f(q_{28}, l) = (q_{28}, b, -1)$
$f(q_{29}, b) = (q_{30}, u, -1)$
$f(q_{30}, b) = (q_7, b, -1)$

2.5 THUE-MORSEов низ

Задатак: Направити програм за Тјурингову машину која генерише `THUE-MORSE`ов низ.

`THUE-MORSE`ов низ је бесконачан бинарни низ. Он се добија почев од 0, а потом се низ продужава низом цифара које представљају комплемент постојећег низа, на следећи начин:

```
0
01
0110
01101001
0110100110010110
...
```

Односно, може се одредити рекурентном формулом

$$t_0 = 0$$

$$t_{2n} = t_n$$

$$t_{2n+1} = 1 - t_n$$

за све ненегативне целе бројеве n.

Алтернативно, може се добити морфизмом тако што сваки симбол "0" у наредној итерацији мењамо са "01", а сваки симбол "1" sa "10". Можемо дефинисати супституције $\mu(0) = 01$ и $\mu(1) = 10$. Тада се низ добија као $\mu^\infty(0) = \mu(\mu(\mu(...\mu(0)...)))$

Још један начин дефинисања овог низа је

$$\prod_{i=0}^{\infty}(1 - x^{2^i}) = \sum_{j=0}^{\infty}(-1)^{t_j} x^j$$

Занимљиво за овај низ је да је он фрактал, тј уклањањем сваког непарног члана из низа добија се полазни низ.

`THUE-MORSE`ов низ такође има особину да се било који његов подниз дужине 2^k или $3 \cdot 2^k$ појављује два пута узастопно негде у оквиру овог бесконачног низа, али да се ниједан такав подниз не појављује три пута узастопно. `THUE-MORSE`ов низ се често користи при налажењу низова који немају оваква и слична понављања у себи. На пример, можемо доказати да је већ од азбуке са 3 различита знака могуће направити низ без двоструких узастопних понављања делова низа.

Још једна примена `THUE-MORSE`овог низа је у комбинаторној теорији игара, односно у анализи комбинаторних игара (попут игре судоку или шаха) са елементима вероватноће. Проучавање оваквих игара корисно је и за развој вештачке интелигенције.

Сам низ откривен је као решење `PROUHET-TARRY-ESCOTT`овог проблема, који тражи два дисјунктна мултискупа A и B од по n целих бројева чији су полиноми суме експонената елемената једнаки, односно

$$\sum_{a \in A} a^i = \sum_{b \in B} b^i$$

где је i цео број између 1 и k, при чему је $k < n$.

Без даљег проучавања особина овог низа (а има их много!) прелазимо на једну од имплементација генератора низа на Тјуринговој машини.

Како је `THUE-MORSE`ов низ по својој природи бесконачан, а потребно нам је да се програм изврши за коначно време, адаптираћемо проблем да се програм извршава у коначном броју итерација, односно да добијемо коначан низ. Као и у задатку са геометрикским низом, низ ће бити ограничен бројачем који корисник дефинише, у истом формату као и у том задатку: број унетих посебних знакова на почетну траку одговара броју итерација извршавања програма. Као и раније, сличне методе се могу користити за ограничавање низа, било то у виду бројача или у виду ограничавања броја места за упис.

Ради једноставности, овај програм неће исписивати редом кораке генерисања низа на траку, већ ће само на траци остати низ добијен коначним бројем понављања поступка продужавања низа.

Након завршетка програма, на траци остаје део `THUE-MORSE`овог низа у својој правој представи, као секвенца јединица и нула.

Број цифара у коначном низу је 2^{g-1}, где је g број посебних знакова које корисник уноси на почетну траку.

Азбука: Машина ће користити азбуку $S = \{0, 1, o, i, g, p, x, t, b\}$. Њихова функција је следећа:

- 0 - бинарни символ 0
- 1 - бинарни символ 1

Помоћу ова два симбола биће представљен низ, у бинарном запису

- g - знак који корисник уноси на почетну траку, одређује број итерација рачунања низа и служи као бројач
- b - бланко карактер
- остали знаци су помоћни, и њихова функција је описана у одељку *Поступак.*

Почетна и крајња трака: На почетној траци дефинише се број итерација поступка добијања низа, као и у задатку са геометријским, у следећем формату (за 3 итерације)

```
ggg
^
```

где се пре и после овог скупа симбола g налазе бланко карактери, а глава је на првом g, и где број симбола g представља број понављања.

За ову почетну траку, након извршавања програма, на траци остаје коначан `THUE-MORSE`ов низ окружен бланко карактерима:

```
bbbbb0110bb
         ^
```

одакле се види да је за 3 понављања `THUE-MORSE`ов низ `0110`.

Поступак и имплементација: Код овакве имплементације, при свакој итерацији се по један симбол g мења помоћним симболом 0, као ознака да је једна итерација одрађена. У овом решењу се редом узимају цифре (симболи) из оригиналног низа, инвертују и истим редом постављају на крај низа.

Уписивање првог члана

Слично задатку са геометријским низом, овај програм уместо првог g уписује привремену ознаку p, а затим десно од групе симбола g уписује прву цифру 0 и десно од ње привремен маркер t, који служи за обележавање краја тренутног низа.

Провера

Након извршене једне итерације, глава се враћа улево до групе g, где проверава да ли је остало још симбола g, и ако није (што ће препознати по томе да се лево од бланко знака налази p, а не нешто друго) прелази се у завршни режим. Ако је остало још g, још једно (највише лево) g се мења у p, и започиње се процес продужавања низа, који се састоји из фазе копирања и комплементирања, и фазе шифтовања.

Копирање и комплементирање

Овде програм иде редом с лева на десно и чита појединачно знакове низа: ако је најлевљи необележени члан 1 (тј 0), он се привремено обележава симболом i (тј o), а уместо првог бланко знака десно од маркера t уписује симбол 0 (тј 1). Овиме је извшено и инвертовање члана.

Затим се глава враћа десно до најлевљег необележеног знака (он је увек једно место десно од симбола i или o), и овај процес се понавља све док више нема 1 или 0 лево од маркера t. Тада се прелази на фазу шифтовања.

Померање улево

Пошто је продужавање завршено, и даље између оригиналног дела низа и новонасталог продужетка постоји маркер t, потребно је продужетак померити улево за једно место. Ово се постиже тиме што програм прочита први симбол десно од t, упише га на место t, а празно место обележи помоћним знаком x. Потом се опет чита први знак десно од, овог пута, знака x, упише га на место x, а празно место поново обележи помоћним знаком x. Ово се понавља све док се десно од x не нађе бланко карактер, када се уместо x уписује маркер t, као ознака да је то крај новог низа који ће се продужавати.

Након овога се глава помера улево све до групе g, успут враћајући све привремене ознаке i у 1 и o у 0, и поново се започиње провера.

Завршавање програма

Када су сви g замењени са p, глава још једном прелази преко свих уписаних симбола, и мења привремене ознаке p и t у b.

Потребно је напоменути да ова реализација можда и није најоптималнија. Наиме, уместо постављања инверза првог члана низа на прво место после t, инверза другог члана на друго место после t итд, могао је просто да се копира *последњи* члан оригиналног низа на прво слободно место после оригиналног низа, *претпоследњи* члан на друго слободно место после оригиналног низа итд (успут обележавајући који чланови су већ иксопирани). Овиме се елиминише потреба за маркером t, а самим тим и померањем улево комплементираног продужетка, као и само комплементирање. У овој реализацији, међутим, био је циљ да се макар мало прикажу процеси инвертовања и померања.

Изглед траке кроз неколико корака

`ggg`	`ppgbotx`	`pppboit10`
`pgg`	`ppgbo1x`	`pppboi1x0`
`pggb0t`	`ppgbo1t`	`pppboi10x`
`ppgb0t`	`ppgb01t`	`pppboi10t`
`ppgbot`	`pppb01t`	`pppb0110t`
`ppgbot1`	`pppbo1t1`	`bbbb0110b`

Кратак опис стања

q0 - почетно стање. Одмах мења прво g у p и мења у стање 1.
q1 - помера главу удесно све док не наиђе на b и мења у стање 2
q2 - уписује почетну вредност (0) десно од првог b и мења у стање 3.
q3 - намешта маркер t десно од почетне вредности, и мења у стање 4.
q4 - враћа главу улево док не наиђе на g, када се мења у стање 5, или p, када се покреће поступак завршавања (стање 20). Приликом враћања главе, уместо x намешта на t (нови крај), док све привремене ознаке i мења у 1, а o мења у 0.
q5 - иде улево до првог p, помера главу удесно за 1 и мења у стање 6.
q6 - мења g у p, а потом у стање 7.
q7 - помера главу удесно до првог b, а потом мења у стање 8.
q8 - помера главу удесно, прескачући све i и o, и ако наиђе на 1, стање се мења на 9, и уместо њега се уписује i, а ако наиђе на 0, стање се мења на 10, и уместо ње се уписује o. Ако глава наиђе на t, то значи да су сви чланови искомплементирани, и иде се у стање 12, а глава се помера за једно место удесно.
q9 - глава иде десно док не наиђе на b, уместо њега пише 0, а потом се мења на стање 11.
q10 - глава иде десно док не наиђе на b, уместо њега пише 1, а потом се мења на стање 11.
q11 - глава се помера улево све до првог b или прве привремене ознаке i или o, које оставља, и враћа машину у стање 8.
q12 - Започиње процес шифтовања свега десно од t за једно место улево. Помера главу удесно, и ако наиђе на 1 одлази у стање 13, а ако наиђе на 0 одлази у стање 14. Том приликом, уместо цифре за собом оставља привремену ознаку x и помера главу за једно место улево. Ако наиђе на b, процес шифтовања је завршен, знак b остаје а машина одлази назад у стање 4.
q13 - пошто је глава померена улево, уписује се 1, а стање враћа на 12.
q14 - пошто је глава померена улево, уписује се 0, а стање враћа на 12.
q20 - Завршни процес. Помера главу улево док не наиђе на b, успут мењајући све p у b, а потом помера главу удесно док не наиђе на , које мења у b и завршава програм

Инструкције

$f(q_0, g) = (q_1, p, +1)$

$f(q_1, g) = (q_1, g, +1)$
$f(q_1, b) = (q_2, b, +1)$

$f(q_2, b) = (q_3, 0, +1)$

$f(q_3, b) = (q_4, t, -1)$

$f(q_4, 1) = (q_4, 1, -1)$
$f(q_4, 0) = (q_4, 0, -1)$
$f(q_4, b) = (q_4, b, -1)$
$f(q_4, g) = (q_5, g, -1)$
$f(q_4, p) = (q_{20}, b, -1)$
$f(q_4, x) = (q_4, t, -1)$
$f(q_4, i) = (q_4, 1, -1)$
$f(q_4, o) = (q_4, 0, -1)$

$f(q_5, g) = (q_5, g, -1)$
$f(q_5, p) = (q_6, p, +1)$

$f(q_6, g) = (q_7, p, +1)$

$f(q_7, g) = (q_7, g, +1)$
$f(q_7, b) = (q_8, b, +1)$

$f(q_8, i) = (q_8, i, +1)$
$f(q_8, o) = (q_8, o, +1)$
$f(q_8, 1) = (q_9, i, +1)$
$f(q_8, 0) = (q_{10}, o, +1)$
$f(q_8, t) = (q_{12}, t, +1)$

$f(q_9, t) = (q_9, t, +1)$
$f(q_9, 0) = (q_9, 0, +1)$
$f(q_9, 1) = (q_9, 1, +1)$
$f(q_9, b) = (q_{11}, 0, -1)$

$f(q_{10}, t) = (q_{10}, t, +1)$
$f(q_{10}, 0) = (q_{10}, 0, +1)$
$f(q_{10}, 1) = (q_{10}, 1, +1)$
$f(q_{10}, b) = (q_{11}, 1, -1)$

$f(q_{11}, t) = (q_{11}, t, -1)$
$f(q_{11}, 1) = (q_{11}, 1, -1)$
$f(q_{11}, 0) = (q_{11}, 0, -1)$
$f(q_{11}, i) = (q_8, i, +1)$
$f(q_{11}, o) = (q_8, o, +1)$
$f(q_{11}, b) = (q_8, b, +1)$

$f(q_{12}, x) = (q_{12}, x, +1)$
$f(q_{12}, 1) = (q_{13}, x, -1)$
$f(q_{12}, 0) = (q_{14}, x, -1)$
$f(q_{12}, b) = (q_4, b, -1)$

$f(q_{13}, t) = (q_{12}, 1, +1)$
$f(q_{13}, x) = (q_{12}, 1, +1)$

$f(q_{14}, t) = (q_{12}, 0, +1)$
$f(q_{14}, x) = (q_{12}, 0, +1)$

$f(q_{20}, p) = (q_{20}, b, -1)$
$f(q_{20}, b) = (q_{20}, b, +1)$
$f(q_{20}, 1) = (q_{20}, 1, +1)$
$f(q_{20}, 0) = (q_{20}, 0, +1)$
$f(q_{20}, t) = (q_+, b, +1)$

Литература

[1] https://sr.wikipedia.org/sr-el/Zlatni_presek

[2] https://en.wikipedia.org/wiki/Fibonacci#Fibonacci_sequence

[3] https://sh.wikipedia.org/wiki/Fibona%C4%8Dijev_niz

[4] https://mathworld.wolfram.com/FibonacciNumber.html

[5] https://hr.wikipedia.org/wiki/Geometrijski_niz

[6] http://www.cs.cornell.edu/courses/cs4820/2014sp/notes/kozen-computability-sp13.pdf

[7] https://www.cs.hmc.edu/~oneill/papers/Sieve-JFP.pdf

[8] https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes

[9] https://en.wikipedia.org/wiki/Natural_number

[10] https://en.wikipedia.org/wiki/Thue%E2%80%93Morse_sequence

[11] https://jjj.de/fxt/fxtbook.pdf

[12] http://www.palacios-huerta.com/docs/EI-Tournaments_and_PTM_sequence.pdf

[13] https://matematis.weebly.com/uploads/3/8/3/2/38325799/_.pdf

[14] https://en.wikipedia.org/wiki/Thue%E2%80%93Morse_sequence

[15] https://en.wikipedia.org/wiki/Prouhet%E2%80%93Tarry%E2%80%93Escott_problem

[16] https://sites.math.washington.edu/~morrow/336_12/papers/christopher.pdf

[17] https://encyclopediaofmath.org/wiki/Thue-Morse_sequence

[18] https://sumo.stanford.edu/pdfs/smt2014/thuemorse-solutions.pdf