Co potrafimy policzyć?

Pytanie:

- Czy można wyliczyć wartość funkcji $f(x,y)=2x+3y$ dla dowolnych danych wejściowych (czyli dowolnych liczb x i y)?

Odpowiedź:

- Tak!

$$2 \cdot x + 3 \cdot y$$

Przepis na obliczenie funkcji f:

- pomnóż wartość x przez 2
- pomnóż wartość y przez 3
- dodaj do siebie wyliczone wcześniej wartości 2x oraz 3y

przepis = algorytm

Algorytm – przepis na sukces

Algorytm
– skończony ciąg jasno zdefiniowanych czynności, koniecznych do wykonania pewnego rodzaju zadań.

- Algorytm ma przeprowadzić system z pewnego stanu początkowego do pożądanego stanu końcowego.
- Badaniem algorytmów zajmuje się algorytmika.
- Algorytm może zostać zaimplementowany w postaci programu komputerowego.

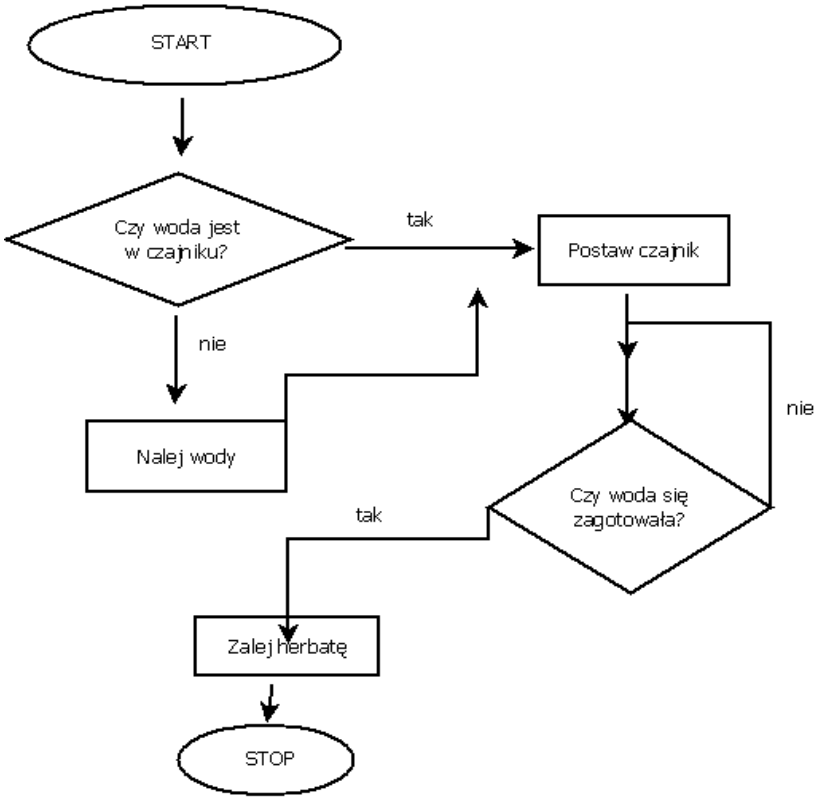
Jak radzić sobie z problemami?

Etapy rozwiązywania problemów

- Sformułowanie zadania
- Określenie danych wejściowych
- Określenie celu, czyli wyniku
- Poszukiwanie metody rozwiązania, czyli algorytmu
- Przedstawienie algorytmu w postaci:
  - opisu słownego
  - listy kroków
  - schematu blokowego
  - języka programowania
- Analiza poprawności rozwiązania
- Testowanie rozwiązania dla różnych danych. Ocena efektywności przyjętej metody.

Gotowanie wody na herbatę

1. Sprawdź czy w czajniku jest woda.
2. Jeśli nie, nalej wody do czajnika.
3. Postaw czajnik na ogniu.
4. Czkaj aż woda się zagotuje.
5. Zalej herbatę



lista kroków

schemat blokowy

Obliczalność

- Funkcja jest obliczalna, gdy istnieje efektywna metoda (algorytm), za pomocą której można obliczyć jej wartość dla dowolnego ciągu jej argumentów.

Uwaga:

Istnieją funkcje, które nie są obliczalne!

Obliczalność – teraz i w przyszłości

Teza Churcha-Turinga

nasze intuicyjne pojęcie obliczalności jest równoważne pojęciu obliczalności definiowanemu w terminach maszyn licznikowych (równoważnie: maszyn Turinga czy inhibitorowych sieci liczących).

Co to znaczy?

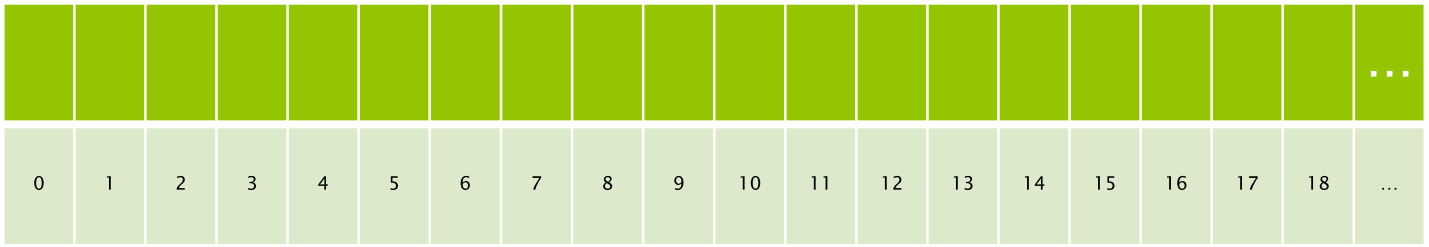
Wszystko co „da się policzyć", teraz lub w przyszłości i na komputerach, które dopiero powstaną, da się policzyć z użycie maszyny licznikowej (odpowiednio: maszyny Turinga czy inhibitorowej sieci liczącej).

Maszyna licznikowa – wprowadzenie

- Maszyna licznikowa jest teoretycznym modelem komputera wykonującym programy tworzone w języku przypominającym prosty assembler.

- Funkcje obliczalne na maszynie licznikowej nazywamy funkcjami ML-obliczalnymi.

Maszyna licznikowa



nieskończona pamięć
(składa się z rejestrów)

licznik rozkazów
(zawiera numer kolejnej
instrukcji do wykonania)

Maszyna licznikowa – rejestry

0	3	2	5	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	…
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	…

- rejestr może zawierać dowolnie dużą liczbę naturalną
- maszyna licznikowa może wykorzystać dowolnie wiele rejestrów
- wszystkie rejestry, poza skończoną liczbą zawierają wartość 0
- zawartość rejestru o numerze n oznaczamy przez z[n]

Przykład: z[3]=5

Maszyna licznikowa – lista instrukcji

Kod instrukcji	Kod adresów	Oznaczenie	Semantyka	Uwagi
0	n	**Z(n)**	z[n]:=0	licznik rozkazów zwiększ o 1
1	n	**S(n)**	z[n]:=z[n]+1	licznik rozkazów zwiększ o 1
2	$\pi(m,n)$	**T(m,n)**	z[n]:=z[m]	licznik rozkazów zwiększ o 1
3	$\beta(m,n,q)$	**I(m,n,q)**	if z[m]=z[n] then goto q	licznik rozkazów zwiększ o 1 gdy z[m]≠z[n], w przeciwnym przypadku umieść w nim q

$$\pi(n,m) = 2^n(2m+1)-1$$
$$\beta(n,m,q) = \pi(\pi(n,m),q)$$

Maszyna licznikowa – co potrafimy?

- zerowanie rejestru
- inkrementowanie (zwiększanie o jeden) zawartości rejestru
- przepisywanie zawartości jednego rejestru do drugiego
- porównywanie zawartości dwóch rejestrów i w przypadku zgodności skok do instrukcji o zadanym numerze

Jak wykonać proste operacje?

Jak wykonać działanie: x+5?

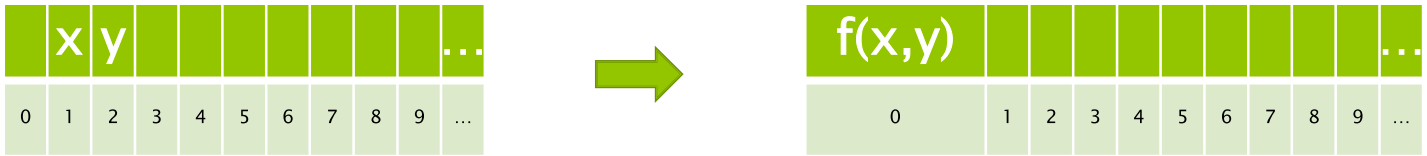
z[1]=x

Operacja	z[1]
	x
S(1)	(x+1)
S(1)	((x+1)+1)
S(1)	(((x+1)+1)+1)
S(1)	((((x+1)+1)+1)+1)
S(1)	(((((x+1)+1)+1)+1)+1)

Jak działa program?

Przyjęta konwencja:

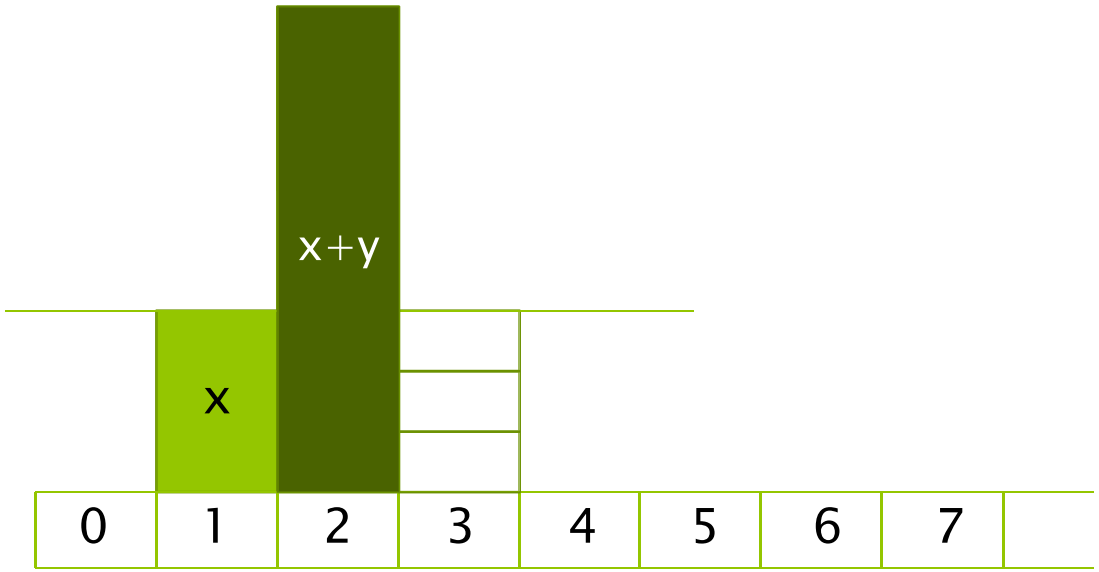
- argumenty wstawiamy na taśmę począwszy od rejestru pierwszego
- program rozpoczyna działanie, podczas którego może modyfikować zawartość dowolnych rejestrów
- po zakończeniu działania programu wynik znajduje się w rejestrze zerowym



Jak wykonać proste operacje?

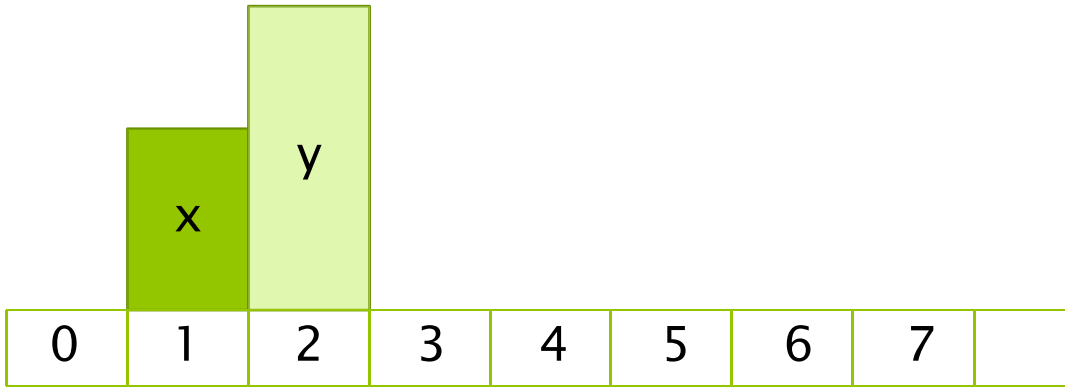
Jak wykonać działanie: x+y?

Musimy x razy dodać jedynkę do y (lub y razy dodać jedynkę do x).



Jak wygląda maszyna licznikowa?

0: I(1,3,4)
1: S(2)
2: S(3)
3: I(0,0,0)
4: T(2,0)



Jaką funkcję liczy następujący program?

0	I(1,2,5)	if z[1]=z[2] then goto 5
1	S(2)	z[2]:= z[2]+1
2	S(3)	z[3]:= z[3]+1
3	I(1,2,5)	if z[1]=z[2] then goto 5
4	I(1,1,1)	if z[1]=z[1] then goto 1
5	T(3,0)	z[0]:= z[3]

Ćwiczenia

Ćwiczenie 1

Napisz program na maszynę licznikową obliczający następujące funkcje:

a) $f_1(x)=x \bmod 2$

– reszta z dzielenia liczby x przez 2

b) $f_2(x,y)=\begin{cases} 1 & x \leq y \\ 0 & x > y \end{cases}$

c) $f_3(x,y)=\min(x,y)$

Ćwiczenia

Ćwiczenie 2

Wykaż, że następujące funkcje są ML-obliczalne:

a) $f(x) = \begin{cases} \frac{1}{2}x & \text{gdy x jest parzyste} \\ \infty & \text{w przeciwnym przypadku} \end{cases}$

b) $f(x) = \begin{cases} 0 & x=0 \\ 1 & x\neq 0 \end{cases}$

c) $f(x) = \begin{cases} 0 & x=y \\ 1 & x\neq y \end{cases}$

d) $f(x) = \begin{cases} x-1 & x>0 \\ 0 & x\neq 0 \end{cases}$

e) $f(x,y) = \begin{cases} x-y & x\geq y \\ 0 & x<y \end{cases}$

f) $f(x)=x \text{ div } 2$

g) $f(x,y)=|x-y|$

Ćwiczenia

Ćwiczenia dodatkowe

Napisz programy na maszynę licznikową obliczające następujące funkcje:

- $f(x,y) = x \cdot y$
- $f(n) = n^5$
- $f(n) = n^{10}$
- $f(a,b,c,d,e,f,x) = ax^5+bx^4+cx^3+dx^2+ex+f$
- $f(n)=2^n$