

Stefan Sokołowski

JĘZYKI FORMALNE I METODY KOMPILACJI

wykład 12

**Inst. Informatyki Stosowanej, ANS
Elbląg, 2024/2025**

Dwuetapowe tłumaczenie programu

Obliczenia skompilowanego programu odbywają się **na stosie**.

Dwuetapowe tłumaczenie programu

Obliczenia skompilowanego programu odbywają się **na stosie**. Dlatego tłumaczenie programu odbywa się w dwóch etapach

Dwuetaapowe tłumaczenie programu

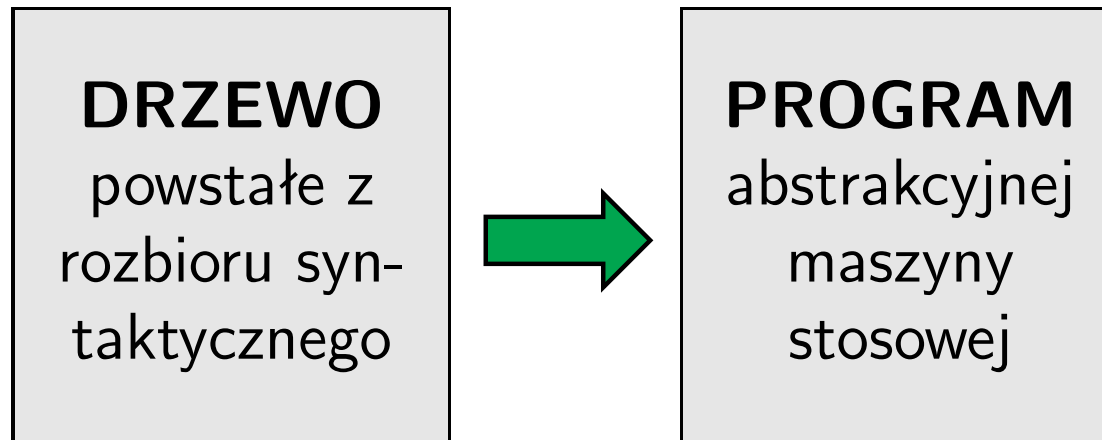
Obliczenia skompilowanego programu odbywają się **na stosie**. Dlatego tłumaczenie programu odbywa się w dwóch etapach:

DRZEWO

powstałe z
rozbioru syn-
taktycznego

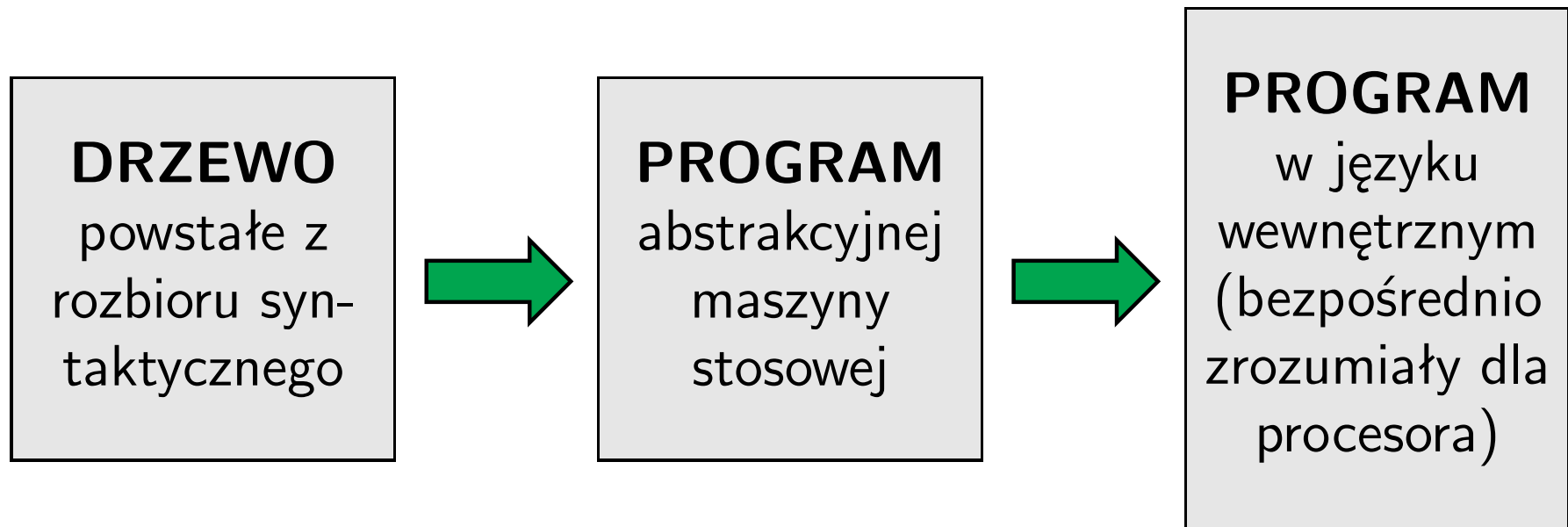
Dwuetapowe tłumaczenie programu

Obliczenia skompilowanego programu odbywają się **na stosie**. Dlatego tłumaczenie programu odbywa się w dwóch etapach:



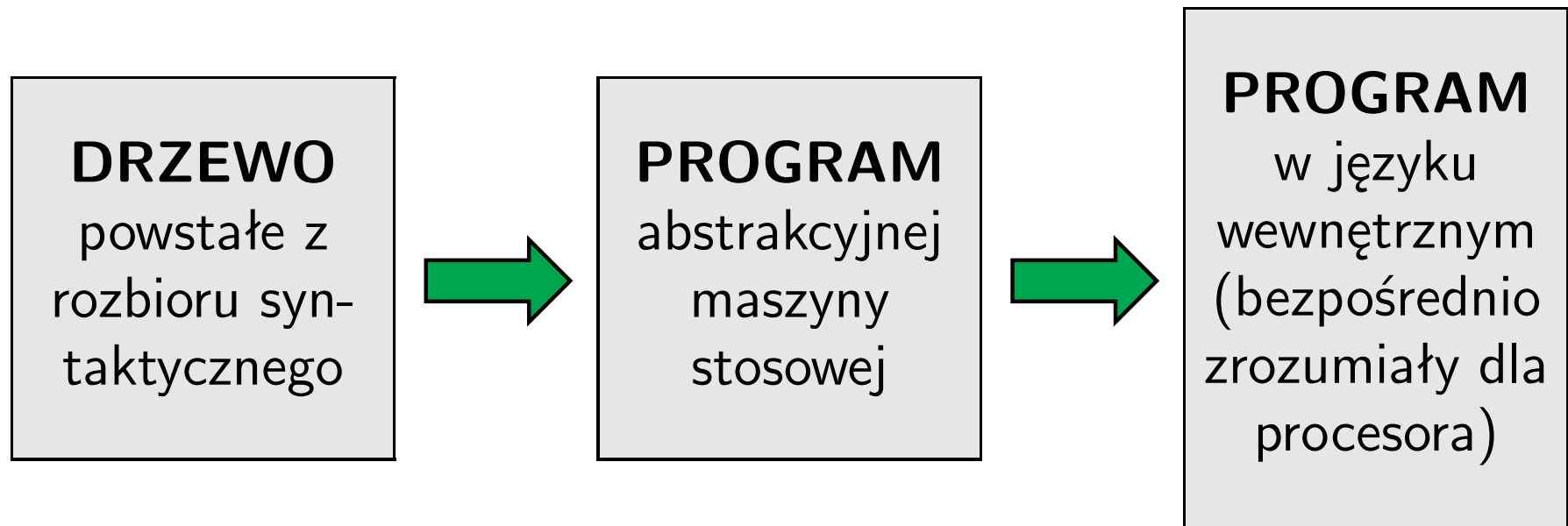
Dwuetapowe tłumaczenie programu

Obliczenia skompilowanego programu odbywają się **na stosie**. Dlatego tłumaczenie programu odbywa się w dwóch etapach:



Dwuetapowe tłumaczenie programu

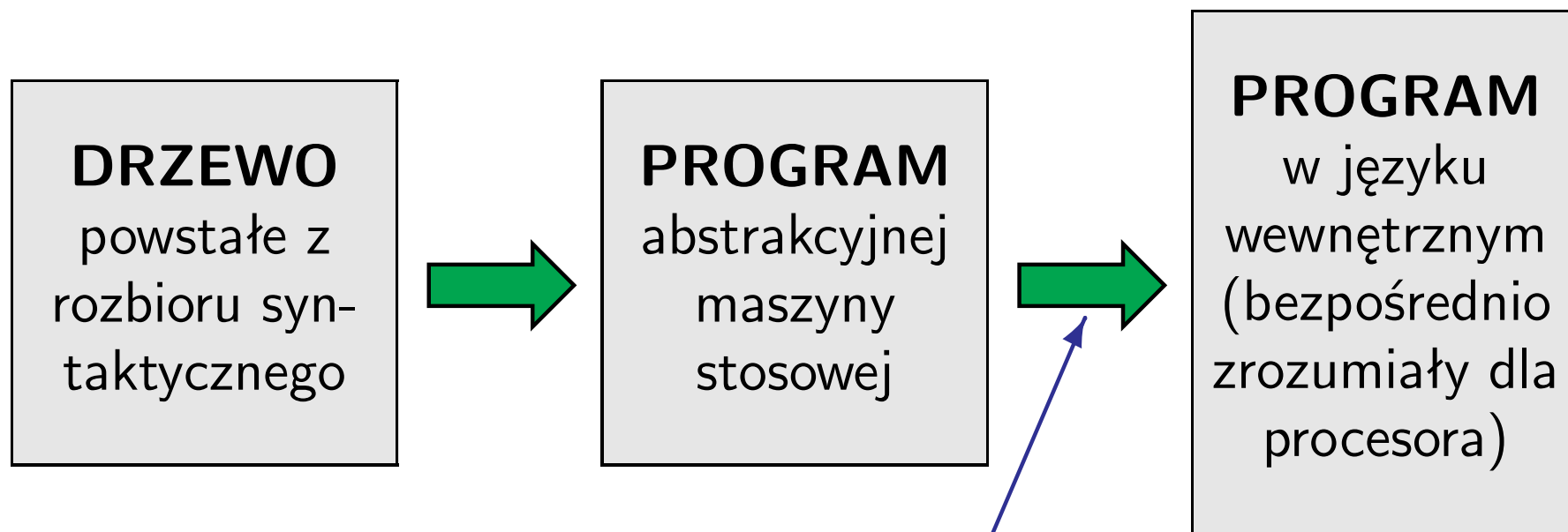
Obliczenia skompilowanego programu odbywają się **na stosie**. Dlatego tłumaczenie programu odbywa się w dwóch etapach:



Abstrakcyjna maszyna stosowa jest niezależna od platformy.

Dwuetapowe tłumaczenie programu

Obliczenia skompilowanego programu odbywają się **na stosie**. Dlatego tłumaczenie programu odbywa się w dwóch etapach:



Abstrakcyjna maszyna stosowa jest niezależna od platformy. Przy przeniesieniu kompilatora na inną platformę trzeba przeprogramować **tylko** drugi etap.

Tłumaczenie komend

Po analizie składniowej komendy są w postaci drzew

Tłumaczenie komend

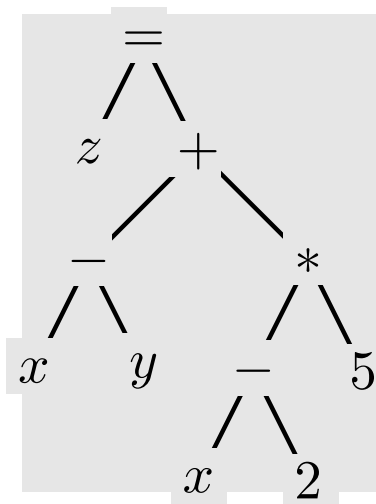
Po analizie składniowej komendy są w postaci drzew; np.

$$z = (x - y) + (x - 2) * 5;$$

Tłumaczenie komend

Po analizie składniowej komendy są w postaci drzew; np.

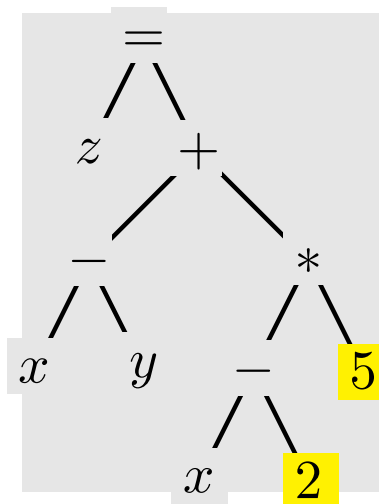
$$z = (x - y) + (x - 2) * 5;$$



Tłumaczenie komend

Po analizie składniowej komendy są w postaci drzew; np.

$$z = (x - y) + (x - 2) * 5;$$

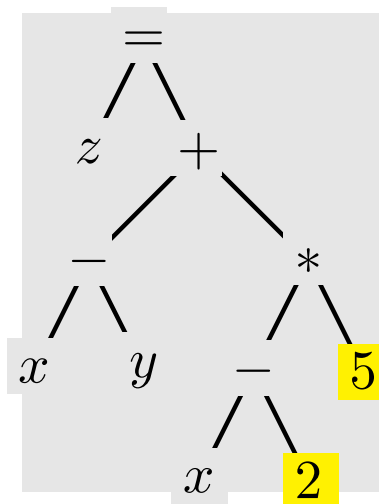


przekład stałej s

Tłumaczenie komend

Po analizie składniowej komendy są w postaci drzew; np.

$$z = (x - y) + (x - 2) * 5;$$



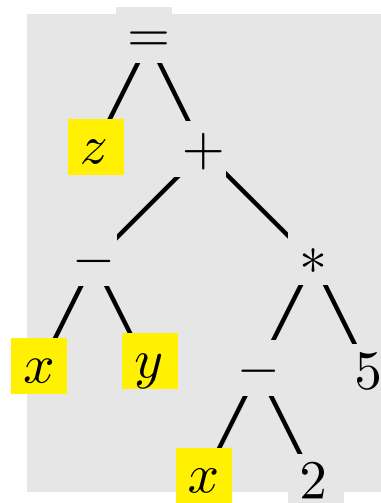
przekład stałej s

na stos liczbę s

Tłumaczenie komend

Po analizie składniowej komendy są w postaci drzew; np.

$$z = (x - y) + (x - 2) * 5;$$



przekład stałej s

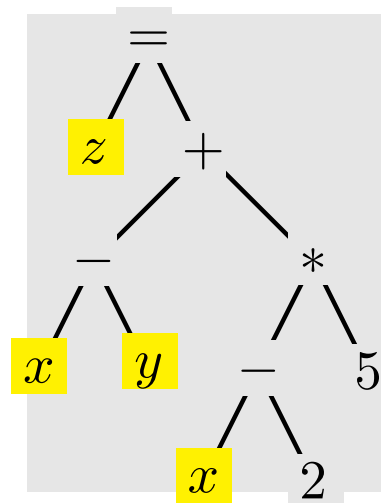
na stos liczbę s

przekład zmiennej o adr. z
po prawej str. przypisania

Tłumaczenie komend

Po analizie składniowej komendy są w postaci drzew; np.

$$z = (x - y) + (x - 2) * 5;$$



przekład stałej s

przekład zmiennej o adr. z
po prawej str. przypisania

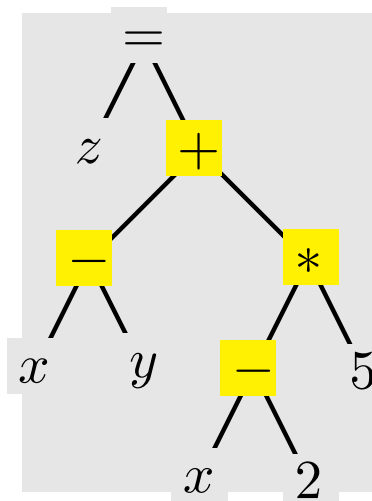
na stos liczbę s

na stos liczbę spod adr. z

Tłumaczenie komend

Po analizie składniowej komendy są w postaci drzew; np.

$$z = (x - y) + (x - 2) * 5;$$



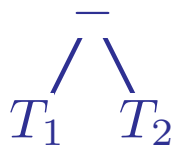
przekład stałej s

na stos liczbę s

przekład zmiennej o adr. z
po prawej str. przypisania

na stos liczbę spod adr. z

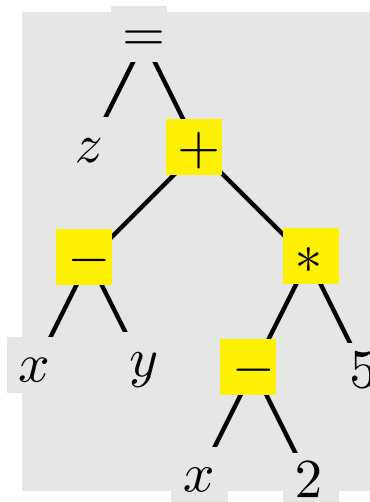
przekład wyrażenia



Tłumaczenie komend

Po analizie składniowej komendy są w postaci drzew; np.

$$z = (x - y) + (x - 2) * 5;$$



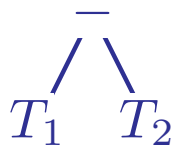
przekład stałej s

na stos liczbę s

przekład zmiennej o adr. z
po prawej str. przypisania

na stos liczbę spod adr. z

przekład wyrażenia



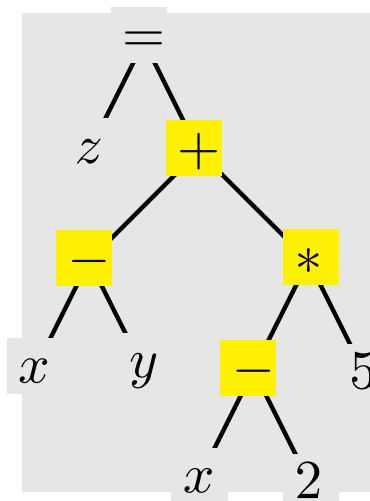
przekład T_1 ; przekład T_2 ;

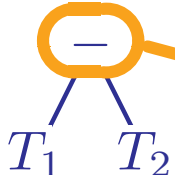
odejmij od przedostatniego na stosie
ostatni na stosie

Tłumaczenie komend

Po analizie składniowej komendy są w postaci drzew; np.

$$z = (x - y) + (x - 2) * 5;$$

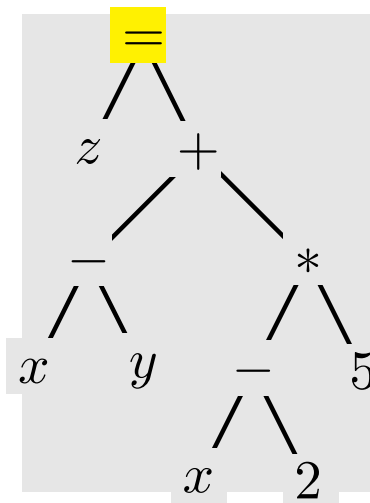


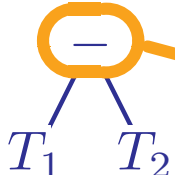
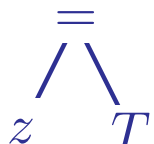
przekład stałej s	na stos liczbę s
przekład zmiennej o adr. z po prawej str. przypisania	na stos liczbę spod adr. z
przekład wyrażenia 	przekład T_1 ; przekład T_2 ; odejmij od przedostatniego na stosie ostatni na stosie

Tłumaczenie komend

Po analizie składniowej komendy są w postaci drzew; np.

$$z = (x - y) + (x - 2) * 5;$$

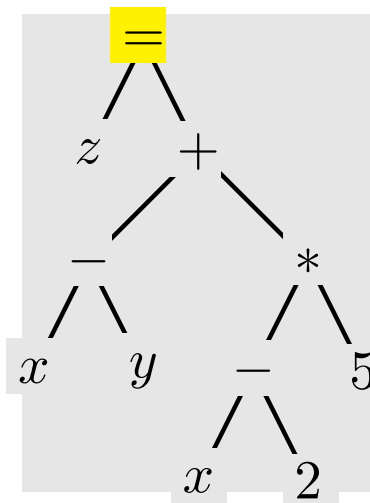


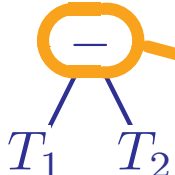
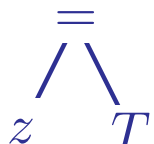
przekład stałej s	na stos liczbę s
przekład zmiennej o adr. z po prawej str. przypisania	na stos liczbę spod adr. z
przekład wyrażenia 	przekład T_1 ; przekład T_2 ; odejmij od przedostatniego na stosie ostatni na stosie
przekład przypisania 	

Tłumaczenie komend

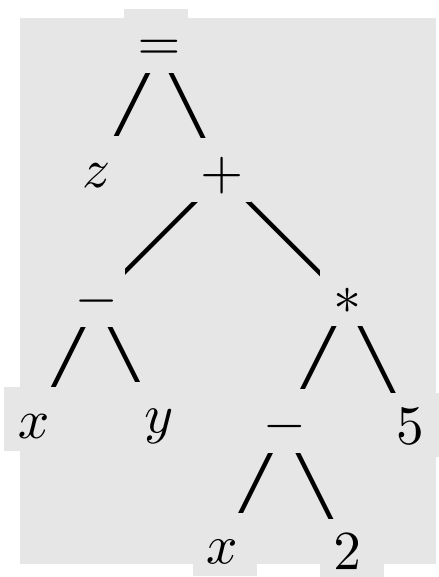
Po analizie składniowej komendy są w postaci drzew; np.

$$z = (x - y) + (x - 2) * 5;$$



przekład stałej s	na stos liczbę s
przekład zmiennej o adr. z po prawej str. przypisania	na stos liczbę spod adr. z
przekład wyrażenia 	przekład T_1 ; przekład T_2 ; odejmij od przedostatniego na stosie ostatni na stosie
przekład przypisania 	na stos adres z ; przekład T ; pod adres przedostatni na stosie włóż wartość ostatnią na stosie

Tłumaczenie komend



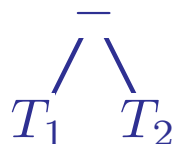
przekład st. s

na stos liczbę s

przekład zmiennej o adr. z
po prawej str. przypisania

na stos liczbę spod adr. z

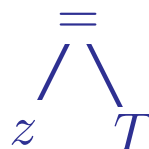
przekład wyr.



przekład T_1 ; przekład T_2 ;

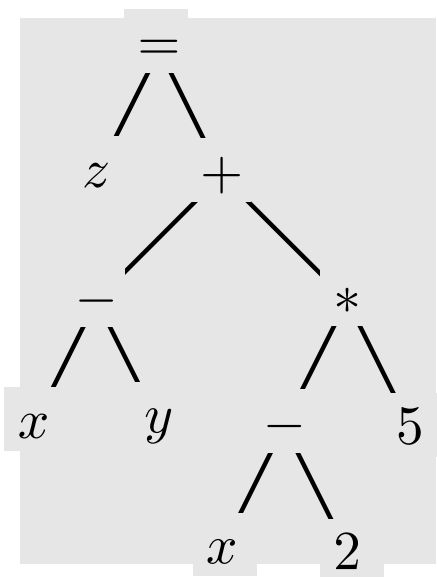
odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

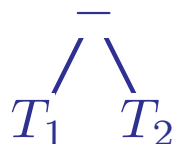


na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

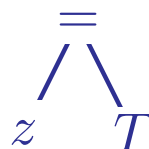
Tłumaczenie komend

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

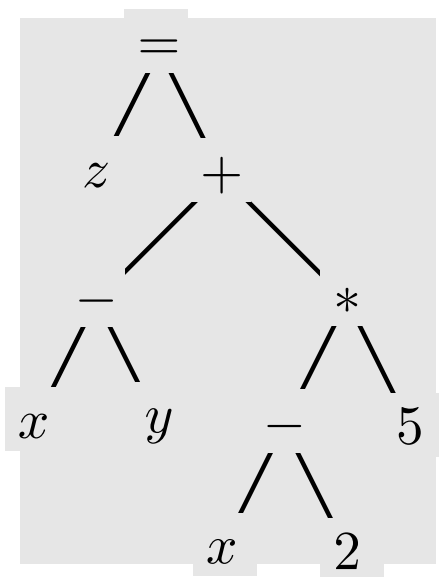
przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

Tłumaczenie komend



PROGRAM:

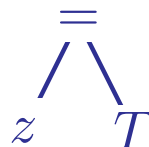
przekład st. s

przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.



na stos liczbę s

na stos liczbę spod adr. z

przekład T_1 ; przekład T_2 ;

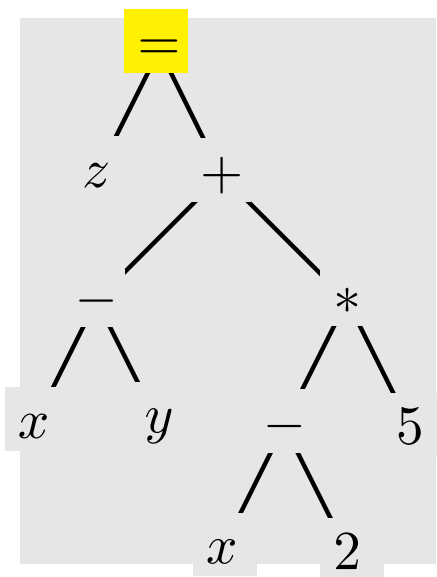
odejmij od przedost. na stosie
ost. na stosie

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

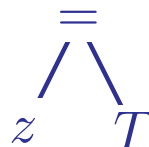
Tłumaczenie komend

przekład st. s przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.

na stos liczbę s na stos liczbę spod adr. z przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosiena stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

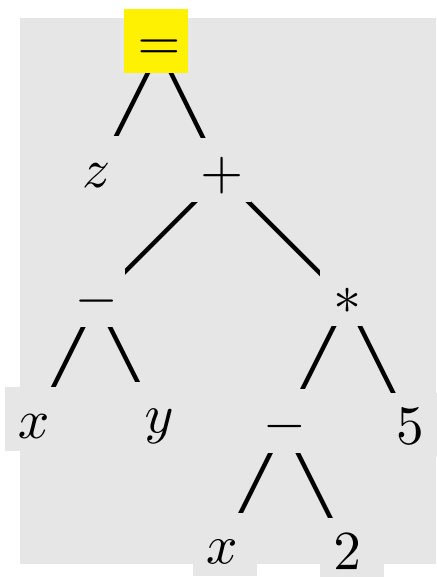
adres z przekład $(x - y) + (x - 2) * 5$

PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

przypisanie

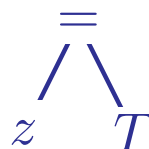
Tłumaczenie komend

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

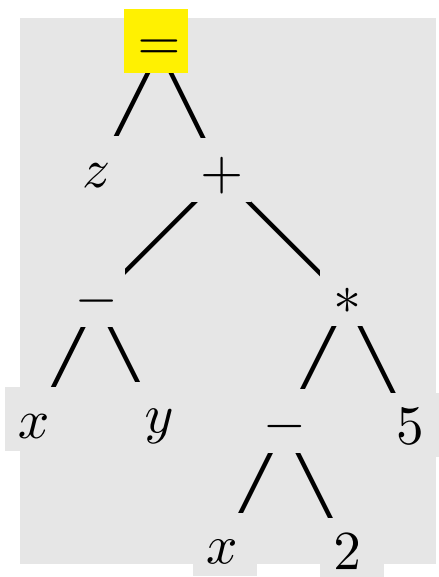
adres z ← na stos adres 52;
przekład $(x - y) + (x - 2) * 5$

PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

przypisanie

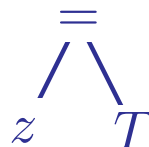
Tłumaczenie komend

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

na stos adres 52;

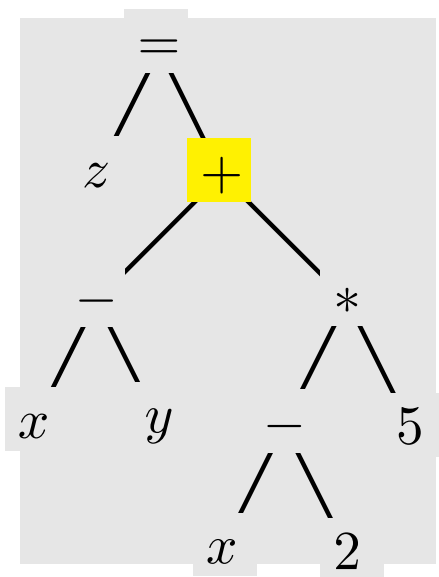
przekład $(x - y) + (x - 2) * 5$

PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

przypisanie

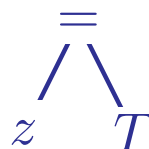
Tłumaczenie komend

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

na stos adres 52;

przekład $(x - y) + (x - 2) * 5$

PAMIĘĆ:

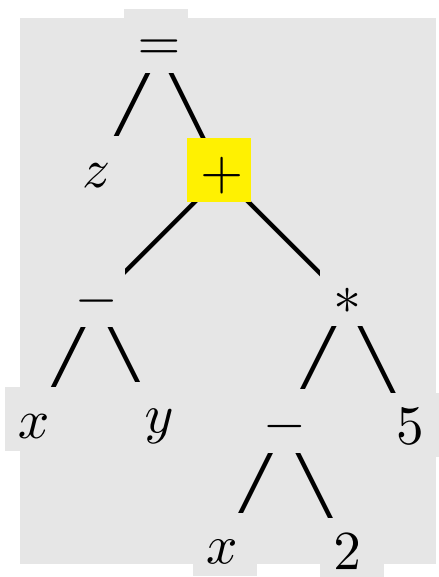
	50	51	52	
...	15	3	-4	...
	x	y	z	

przekład $x - y$ przekład $(x - 2) * 5$

dodawanie

przypisanie

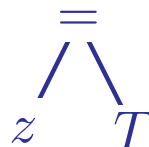
Tłumaczenie komend

przekład st. s przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.

na stos liczbę s na stos liczbę spod adr. z przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosiena stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

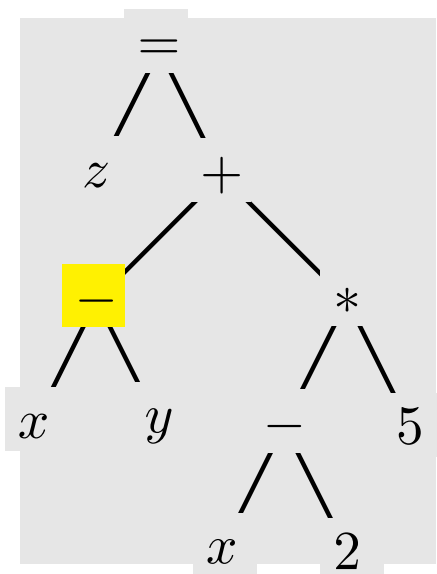
PROGRAM:

na stos adres 52;
przekład $x - y$ przekład $(x - 2) * 5$ dodawanie;
przypisanie

PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

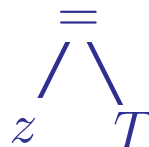
Tłumaczenie komend

przekład st. s przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.

na stos liczbę s na stos liczbę spod adr. z przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosiena stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

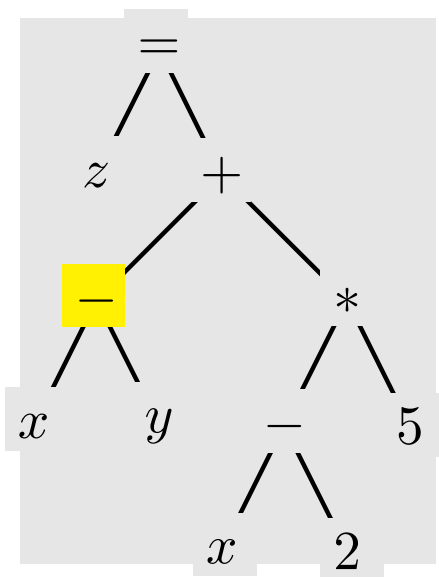
na stos adres 52;
przekład $x - y$ przekład $(x - 2) * 5$

PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowaniedodawanie;
przypisanie

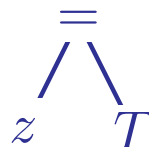
Tłumaczenie komend

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

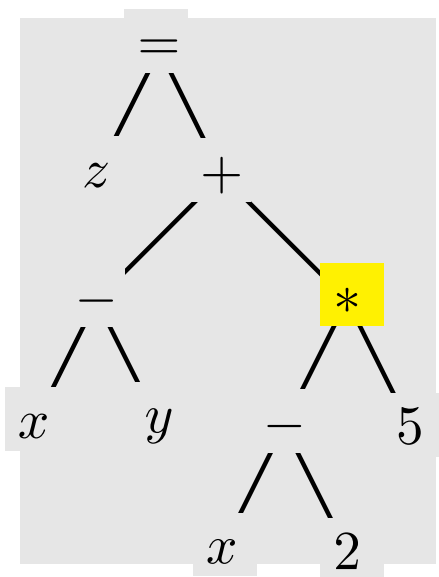
na stos adres 52;
 na stos liczbę spod adr. 50;
 na stos liczbę spod adr. 51;
 odejmowanie;
 przekład $(x - 2) * 5$

PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

dodawanie;
 przypisanie

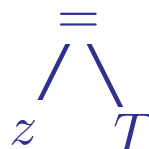
Tłumaczenie komend

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

na stos adres 52;
 na stos liczbę spod adr. 50;
 na stos liczbę spod adr. 51;
 odejmowanie;
 przekład $(x - 2) * 5$

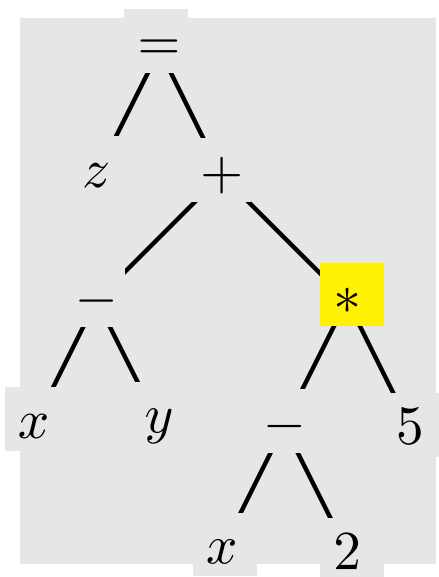
PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

przekład $x - 2$
 na stos liczbę 5;
 mnożenie

dodawanie;
 przypisanie

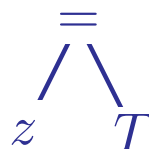
Tłumaczenie komend

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

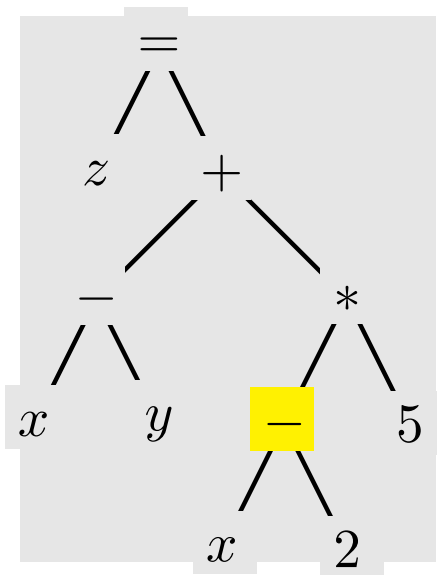
na stos adres 52;
 na stos liczbę spod adr. 50;
 na stos liczbę spod adr. 51;
 odejmowanie;
 przekład $x - 2$

na stos liczbę 5;
 mnożenie;
 dodawanie;
 przypisanie

PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

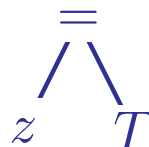
Tłumaczenie komend

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

na stos adres 52;
 na stos liczbę spod adr. 50;
 na stos liczbę spod adr. 51;
 odejmowanie;
 przekład $x - 2$

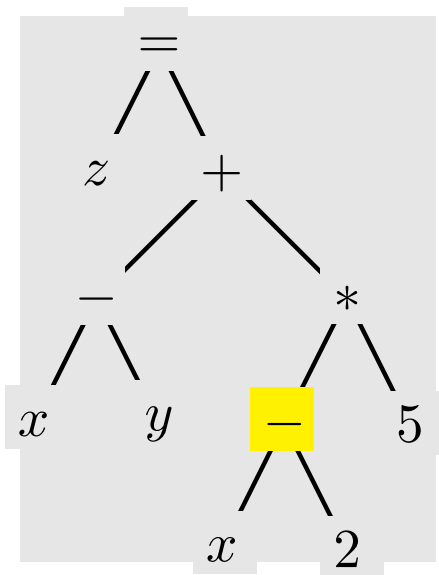
na stos liczbę 5;
 mnożenie;
 dodawanie;
 przypisanie

na stos liczbę spod adr. 50;
 na stos liczbę 2;
 odejmowanie

PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

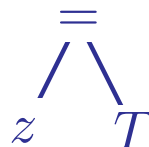
Tłumaczenie komend

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

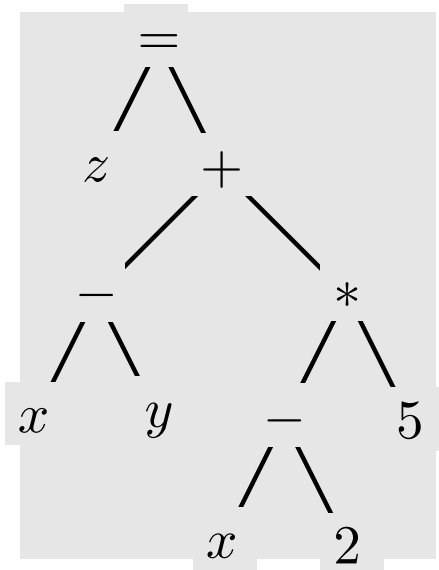
```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

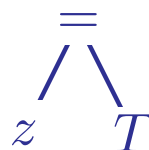
Wykonywanie przekładu

przekład st. s przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.

na stos liczbę s na stos liczbę spod adr. z przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosiena stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

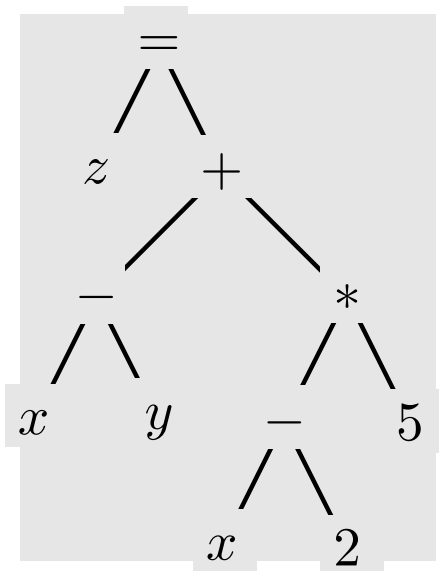
PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

STOS:



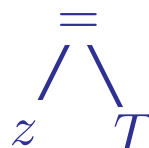
Wykonywanie przekładu

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

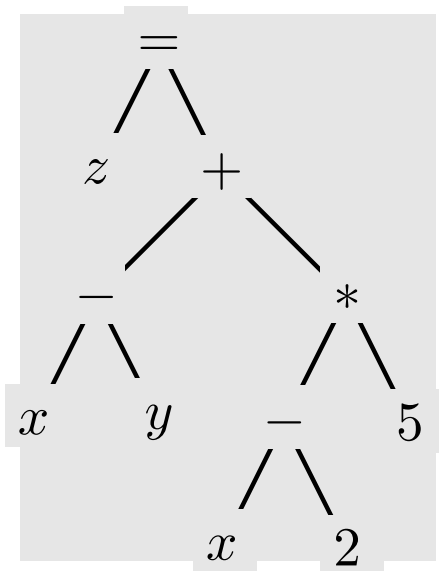
PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

STOS:

52

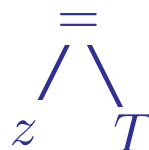
Wykonywanie przekładu

przekład st. s przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.

na stos liczbę s na stos liczbę spod adr. z przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosiena stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

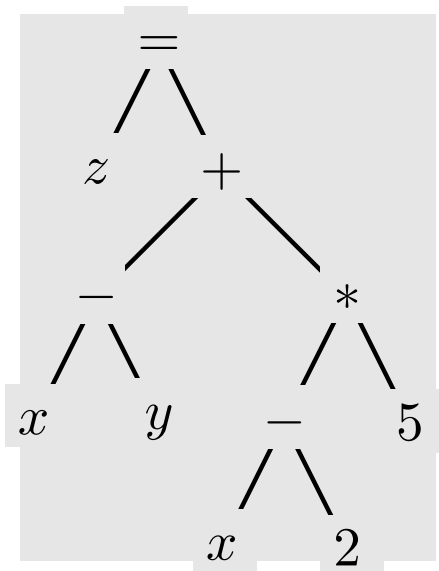
PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

STOS:

52	15
----	----

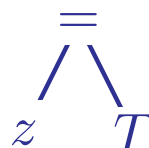
Wykonywanie przekładu

przekład st. s przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.

na stos liczbę s na stos liczbę spod adr. z przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosiena stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

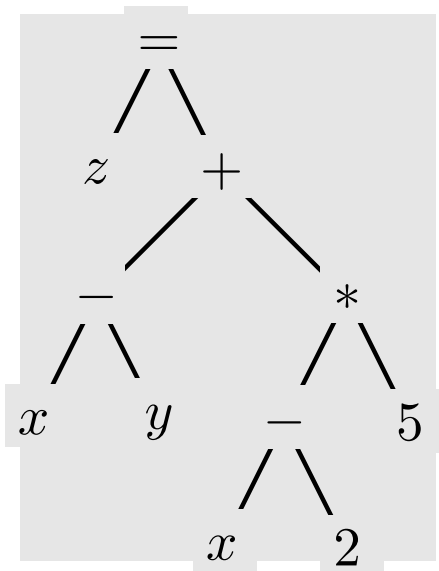
PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

STOS:

52	15	3
----	----	---

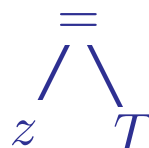
Wykonywanie przekładu

przekład st. s przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.

na stos liczbę s na stos liczbę spod adr. z przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosiena stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

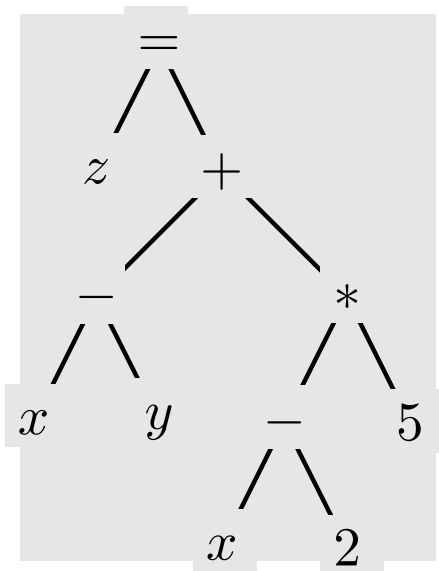
PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

STOS:

52	12
----	----

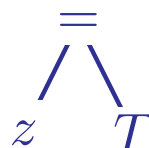
Wykonywanie przekładu

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie

```

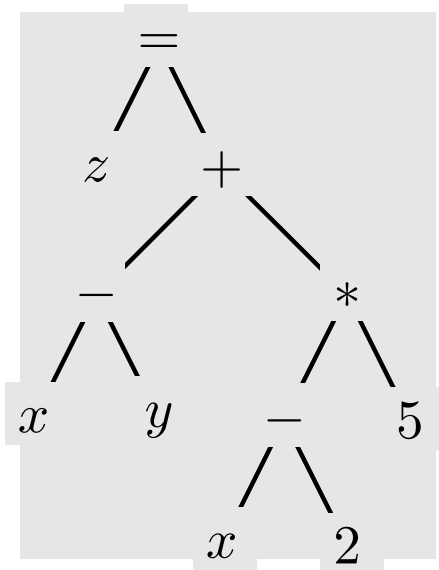
PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

STOS:

52	12	15
----	----	----

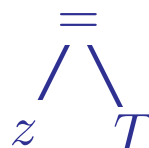
Wykonywanie przekładu

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

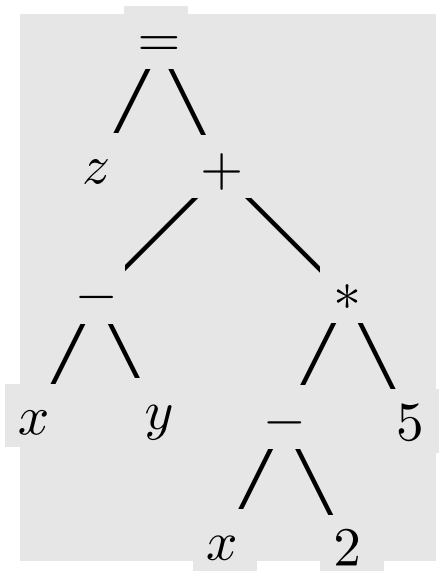
PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

STOS:

52	12	15	2
----	----	----	---

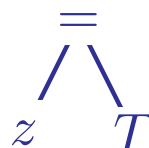
Wykonywanie przekładu

przekład st. s przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.

na stos liczbę s na stos liczbę spod adr. z przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosiena stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

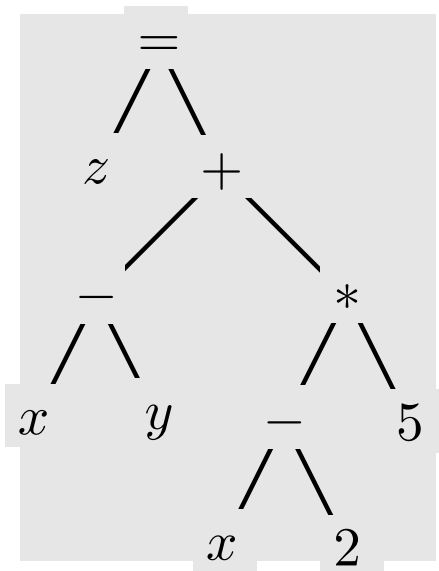
PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

STOS:

52	12	13
----	----	----

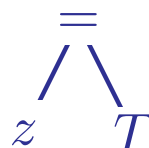
Wykonywanie przekładu

przekład st. s przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.

na stos liczbę s na stos liczbę spod adr. z przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosiena stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

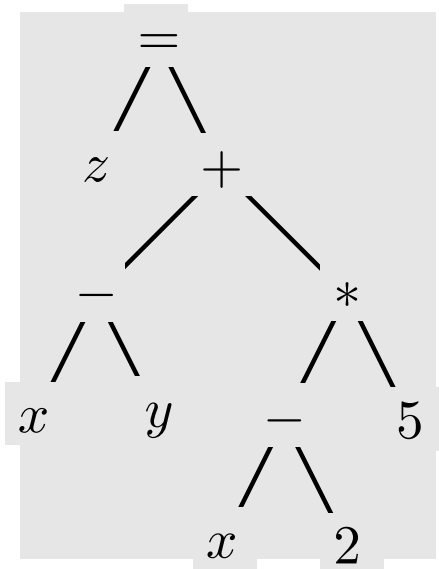
PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

STOS:

52	12	13	5
----	----	----	---

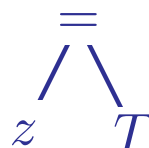
Wykonywanie przekładu

przekład st. s przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.

na stos liczbę s na stos liczbę spod adr. z przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosiena stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

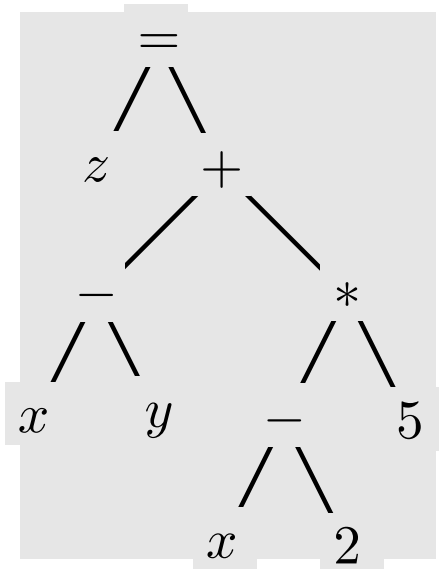
PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

STOS:

52	12	65
----	----	----

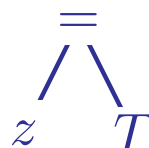
Wykonywanie przekładu

przekład st. s przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.

na stos liczbę s na stos liczbę spod adr. z przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosiena stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

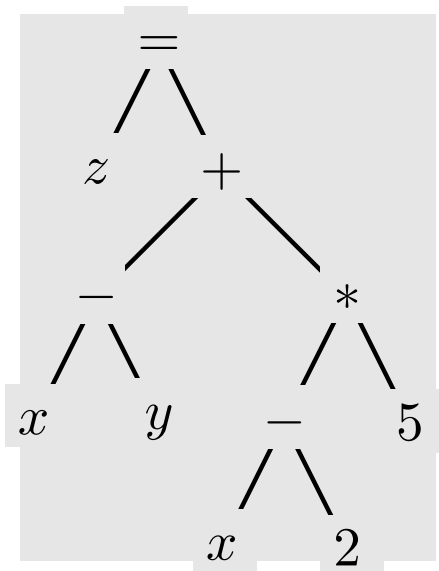
PAMIĘĆ:

	50	51	52	
...	15	3	-4	...
	x	y	z	

STOS:

52	77
----	----

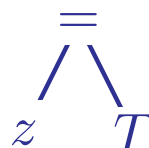
Wykonywanie przekładu

przekład st. s przekład zmiennej o adr. z
po prawej str. przypisania

przekład wyr.



przekład przyp.

na stos liczbę s na stos liczbę spod adr. z przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosiena stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

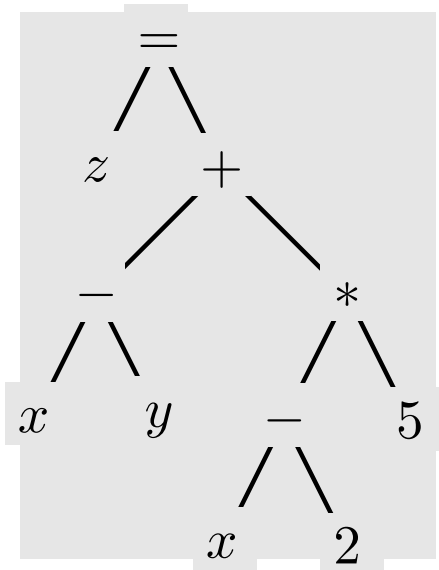
PAMIĘĆ:

	50	51	52	
...	15	3	77	...
	x	y	z	

STOS:

52	77
----	----

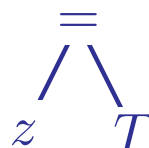
Wykonywanie przekładu

przekład st. s na stos liczbę s przekład zmiennej o adr. z
po prawej str. przypisaniana stos liczbę spod adr. z

przekład wyr.

przekład T_1 ; przekład T_2 ;odejmij od przedost. na stosie
ost. na stosie

przekład przyp.

na stos adres z ; przekład T ;
pod adres przedost. na stosie
włóż wartość ost. na stosie

PROGRAM:

```

na stos adres 52;
na stos liczbę spod adr. 50;
na stos liczbę spod adr. 51;
odejmowanie;
na stos liczbę spod adr. 50;
na stos liczbę 2;
odejmowanie;
na stos liczbę 5;
mnożenie;
dodawanie;
przypisanie
  
```

PAMIĘĆ:

	50	51	52	
...	15	3	77	...
	x	y	z	

STOS:

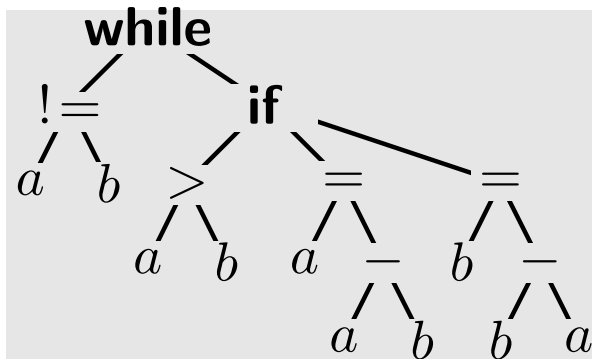


Tłumaczenie komend złożonych

```
while (a != b)
  if (a > b)
    a = a-b;
  else
    b = b-a;
```

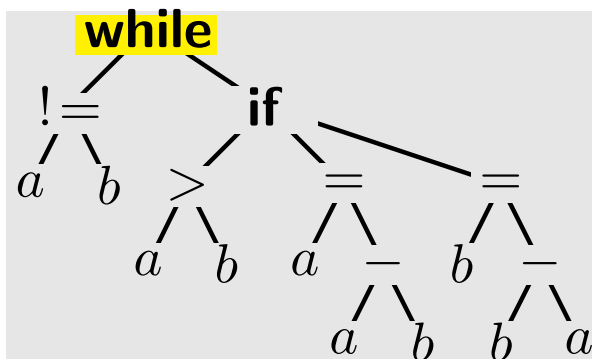

Tłumaczenie komend złożonych

```
while (a != b)
  if (a > b)
    a = a-b;
  else
    b = b-a;
```



Tłumaczenie komend złożonych

```
while (a != b)
  if (a > b)
    a = a-b;
  else
    b = b-a;
```

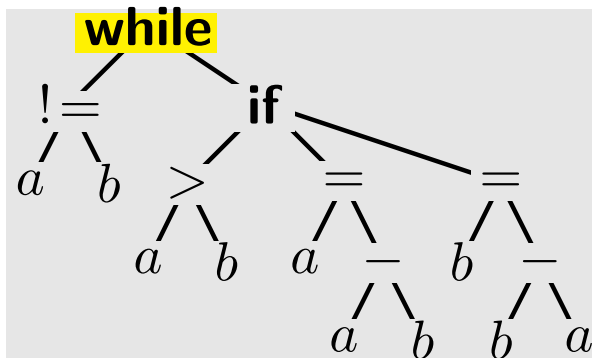


przekład instr.

while
 $T_0' \setminus T$

Tłumaczenie komend złożonych

```
while (a != b)
  if (a > b)
    a = a-b;
  else
    b = b-a;
```



przekład instr.

while
 T_0 / T

E_i : przekład T_0 ;

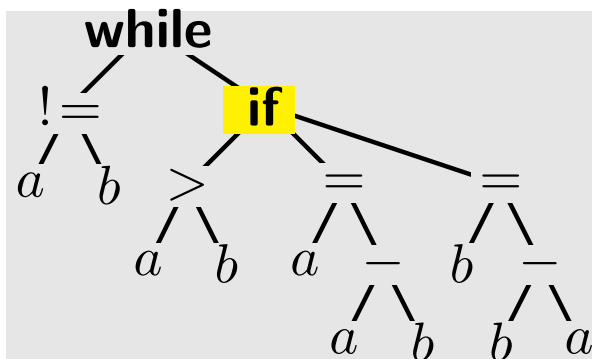
jeśli na st. *false* to do E_{i+1} ;

przekład T ; do E_i ;

E_{i+1} :

Tłumaczenie komend złożonych

```
while (a != b)
  if (a > b)
    a = a-b;
  else
    b = b-a;
```



przekład instr.

while
 T_0 T

E_i : przekład T_0 ;
 jeśli na st. *false* to do E_{i+1} ;
 przekład T ; do E_i ;
 E_{i+1} :

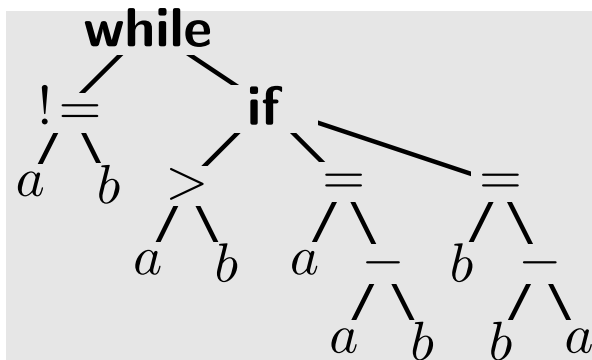
przekład instr.

if
 T_0 T_1 T_2

przekład T_0 ;
 jeśli na st. *false* to do E_i ;
 przekład T_1 ; do E_{i+1} ;
 E_i : przekład T_2 ;
 E_{i+1} :

Tłumaczenie komend złożonych

```
while (a != b)
  if (a > b)
    a = a-b;
  else
    b = b-a;
```



PROGRAM:

przekład instr.

while
 T_0 T

E_i : przekład T_0 ;
 jeśli na st. *false* to do E_{i+1} ;
 przekład T ; do E_i ;
 E_{i+1} :

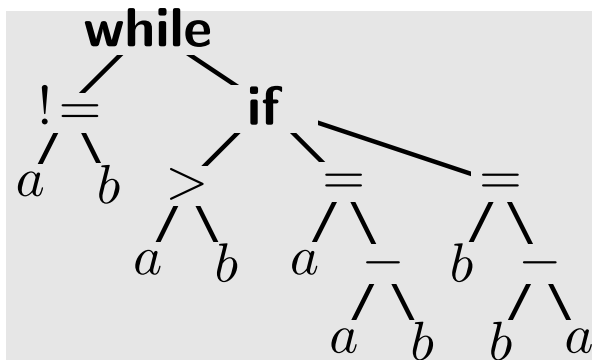
przekład instr.

if
 T_0 T_1 T_2

przekład T_0 ;
 jeśli na st. *false* to do E_i ;
 przekład T_1 ; do E_{i+1} ;
 E_i : przekład T_2 ;
 E_{i+1} :

Tłumaczenie komend złożonych

```
while (a != b)
  if (a > b)
    a = a-b;
  else
    b = b-a;
```



PROGRAM:

E_1 : przekład $a! = b...$

jeśli na stosie *false*
to do E_2 ;
przekład if...

przekład instr.

while
 T_0 T

E_i : przekład T_0 ;
jeśli na st. *false* to do E_{i+1} ;
przekład T ; do E_i ;
 E_{i+1} :

przekład instr.

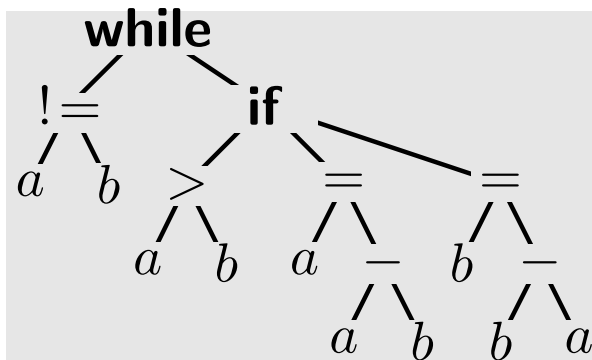
if
 T_0 T_1 T_2

przekład T_0 ;
jeśli na st. *false* to do E_i ;
przekład T_1 ; do E_{i+1} ;
 E_i : przekład T_2 ;
 E_{i+1} :

do E_1 ;
 E_2 :

Tłumaczenie komend złożonych

```
while (a != b)
  if (a > b)
    a = a-b;
  else
    b = b-a;
```



PROGRAM:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 przekład if...

przekład instr.

while
 T_0 T

E_i : przekład T_0 ;
 jeśli na st. *false* to do E_{i+1} ;
 przekład T ; do E_i ;
 E_{i+1} :

przekład instr.

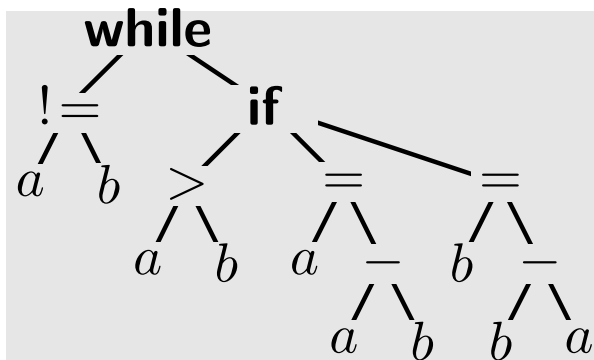
if
 T_0 T_1 T_2

przekład T_0 ;
 jeśli na st. *false* to do E_i ;
 przekład T_1 ; do E_{i+1} ;
 E_i : przekład T_2 ;
 E_{i+1} :

do E_1 ;
 E_2 :

Tłumaczenie komend złożonych

```
while (a != b)
  if (a > b)
    a = a-b;
  else
    b = b-a;
```



przekład instr.

while
 $T_0 \quad T$

E_i : przekład T_0 ;
 jeśli na st. *false* to do E_{i+1} ;
 przekład T ; do E_i ;
 E_{i+1} :

przekład instr.

if
 $T_0 \quad T_1 \quad T_2$

przekład T_0 ;
 jeśli na st. *false* to do E_i ;
 przekład T_1 ; do E_{i+1} ;
 E_i : przekład T_2 ;
 E_{i+1} :

PROGRAM:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 przekład $a > b$

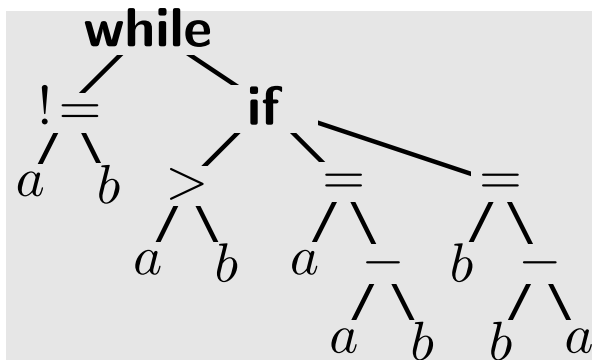
jeśli na stosie *false*
 to do E_3 ;
 przekład $a = a - b$;

do E_4 ;
 E_3 : przekład $b = b - a$;

E_4 : do E_1 ;
 E_2 :

Tłumaczenie komend złożonych

```
while (a != b)
  if (a > b)
    a = a-b;
  else
    b = b-a;
```



przekład instr.

while
 $T_0 \quad T$

E_i : przekład T_0 ;
 jeśli na st. *false* to do E_{i+1} ;
 przekład T ; do E_i ;
 E_{i+1} :

przekład instr.

if
 $T_0 \quad T_1 \quad T_2$

przekład T_0 ;
 jeśli na st. *false* to do E_i ;
 przekład T_1 ; do E_{i+1} ;
 E_i : przekład T_2 ;
 E_{i+1} :

PROGRAM:

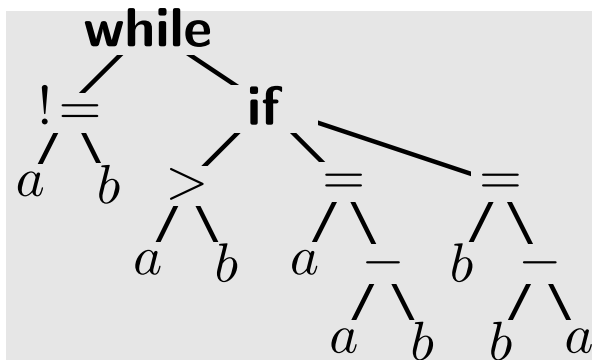
E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 większość;
 jeśli na stosie *false*
 to do E_3 ;
 przekład $a = a - b$;

do E_4 ;
 E_3 : przekład $b = b - a$;

E_4 : do E_1 ;
 E_2 :

Tłumaczenie komend złożonych

```
while (a != b)
  if (a > b)
    a = a-b;
  else
    b = b-a;
```



przekład instr.

while
 T_0 T

E_i : przekład T_0 ;
 jeśli na st. *false* to do E_{i+1} ;
 przekład T ; do E_i ;
 E_{i+1} :

przekład instr.

if
 T_0 T_1 T_2

przekład T_0 ;
 jeśli na st. *false* to do E_i ;
 przekład T_1 ; do E_{i+1} ;
 E_i : przekład T_2 ;
 E_{i+1} :

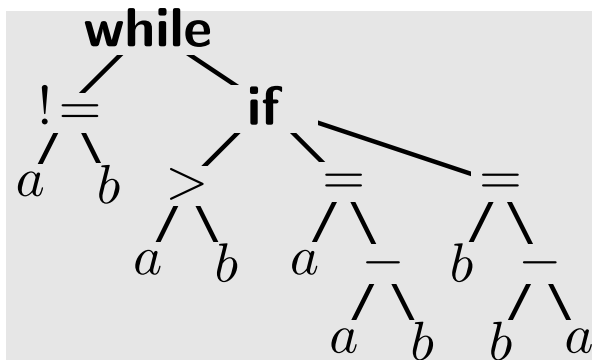
PROGRAM:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 większość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;
 E_3 : przekład $b = b - a$;

E_4 : do E_1 ;
 E_2 :

Tłumaczenie komend złożonych

```
while (a != b)
  if (a > b)
    a = a-b;
  else
    b = b-a;
```



przekład instr.

while
 $T_0 \quad T$

E_i : przekład T_0 ;
 jeśli na st. *false* to do E_{i+1} ;
 przekład T ; do E_i ;
 E_{i+1} :

przekład instr.

if
 $T_0 \quad T_1 \quad T_2$

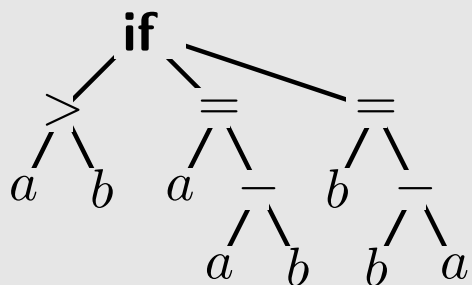
przekład T_0 ;
 jeśli na st. *false* to do E_i ;
 przekład T_1 ; do E_{i+1} ;
 E_i : przekład T_2 ;
 E_{i+1} :

PROGRAM:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 większość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;
 E_3 : na stos adres b ;
 na stos wart.zm. b ;
 na stos wart.zm. a ;
 odejmowanie;
 przypisanie;
 E_4 : do E_1 ;
 E_2 :

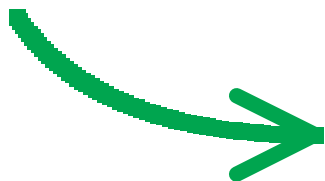
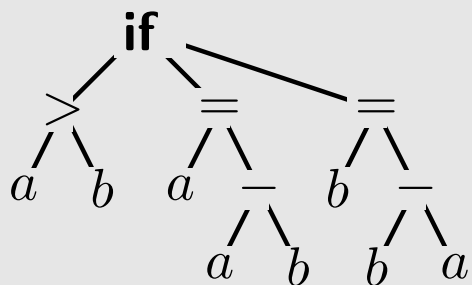
Drugi etap tłumaczenia

DRZEWA:



Drugi etap tłumaczenia

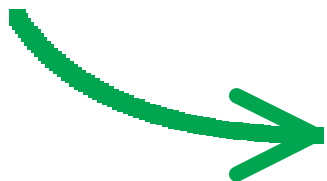
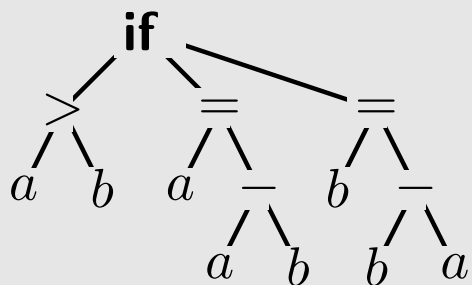
DRZEWA:



ABSTRAKCYJNA
MASZYNA
STOSOWA

Drugi etap tłumaczenia

DRZEWA:



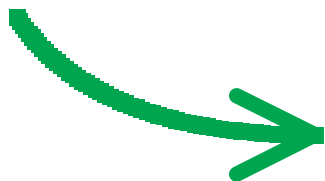
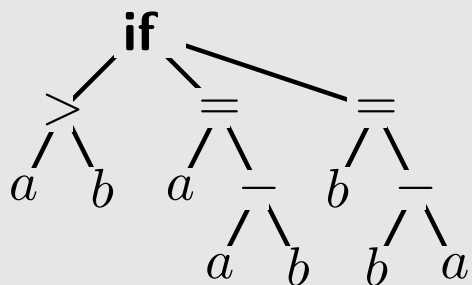
ABSTRAKCYJNA
MASZYNA
STOSOWA

typowe rozkazy:

na stos liczbę 300

Drugi etap tłumaczenia

DRZEWA:



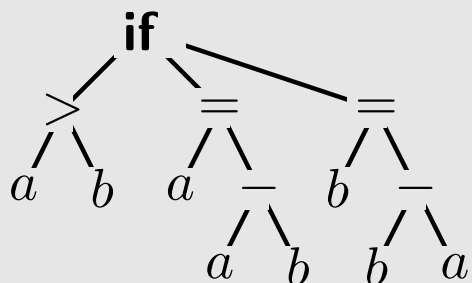
ABSTRAKCYJNA MASZYNA STOSOWA

typowe rozkazy:

na stos liczbę 300
na stos adres *a*

Drugi etap tłumaczenia

DRZEWA:



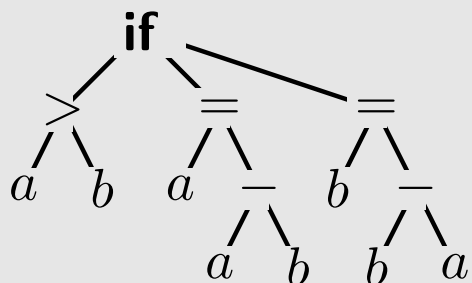
ABSTRAKCYJNA MASZYNA STOSOWA

typowe rozkazy:

na stos liczbę 300
na stos adres *a*
na stos wart.zm. *a*

Drugi etap tłumaczenia

DRZEWA:



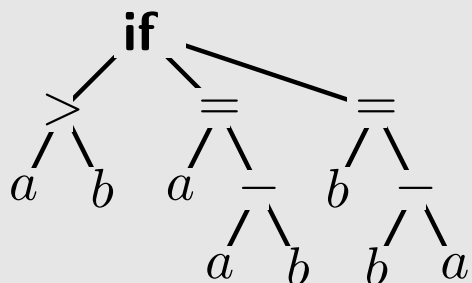
ABSTRAKCYJNA MASZYNA STOSOWA

typowe rozkazy:

na stos liczbę 300
 na stos adres *a*
 na stos wart.zm. *a*
 nierówność
 większość
 . . .

Drugi etap tłumaczenia

DRZEWA:



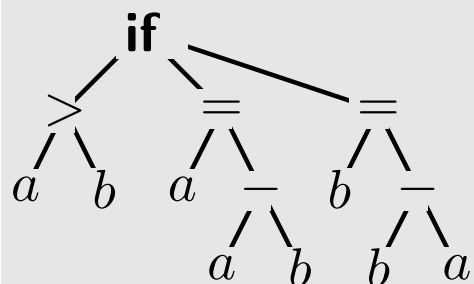
ABSTRAKCYJNA MASZYNA STOSOWA

typowe rozkazy:

na stos liczbę 300
 na stos adres *a*
 na stos wart.zm. *a*
 nierówność
 większość
 . . .
 dodawanie
 odejmowanie
 . . .

Drugi etap tłumaczenia

DRZEWA:



ABSTRAKCYJNA MASZYNA STOSOWA

typowe rozkazy:

na stos liczbę 300

na stos adres *a*

na stos wart.zm. *a*

nierówność

większość

...

dodawanie

odejmowanie

...

do *E*₂

jeśli na stosie *false*

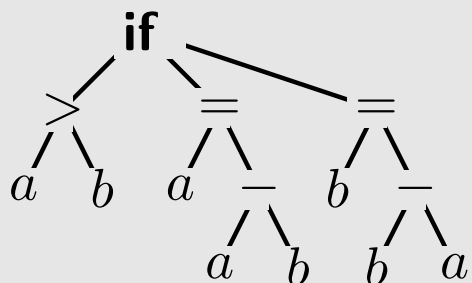
to do *E*

jeśli na stosie *true*

to do *E*

Drugi etap tłumaczenia

DRZEWA:



ABSTRAKCYJNA MASZYNA STOSOWA

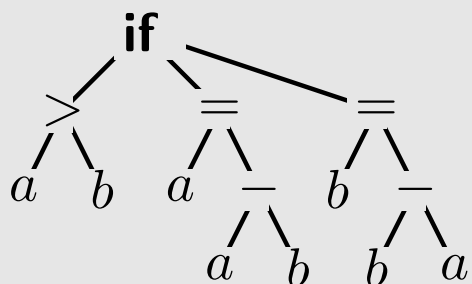
typowe rozkazy:

na stos liczbę 300
 na stos adres *a*
 na stos wart.zm. *a*
 nierówność
 większość
 . . .
 dodawanie
 odejmowanie
 . . .
 do *E*₂
 jeśli na stosie *false*
 to do *E*
 jeśli na stosie *true*
 to do *E*

PRAWDZIWY JĘZYK WEWNĘTRZNY

Drugi etap tłumaczenia

DRZEWA:



ABSTRAKCYJNA MASZYNA STOSOWA

typowe rozkazy:

na stos liczbę 300
 na stos adres *a*
 na stos wart.zm. *a*
 nierówność
 większość

• • •
 dodawanie
 odejmowanie

• • •
 do E_2
 jeśli na stosie *false*
 to do E
 jeśli na stosie *true*
 to do E

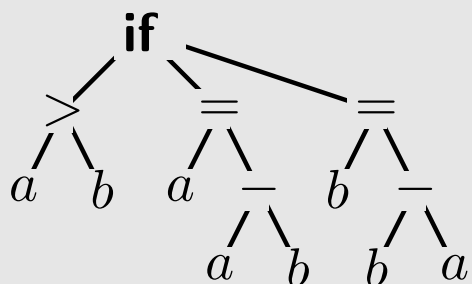
PRAWDZIWY JĘZYK WEWNĘTRZNY

typowe rozkazy:

pobierz spod adr. 300
 włóż pod adres 300

Drugi etap tłumaczenia

DRZEWA:



ABSTRAKCYJNA MASZYNA STOSOWA

typowe rozkazy:

na stos liczbę 300
 na stos adres *a*
 na stos wart.zm. *a*
 nierówność
 większość

• • •
 dodawanie
 odejmowanie

• • •
 do *E*₂
 jeśli na stosie *false*
 to do *E*
 jeśli na stosie *true*
 to do *E*

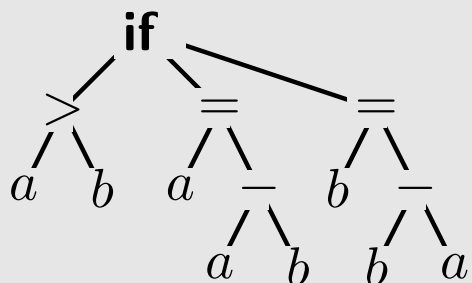
PRAWDZIWY JĘZYK WEWNĘTRZNY

typowe rozkazy:

pobierz spod adr. 300
 włóż pod adres 300
 dodaj spod adr. 300
 odejmij spod adr. 300
 . . .

Drugi etap tłumaczenia

DRZEWA:



ABSTRAKCYJNA MASZYNA STOSOWA

typowe rozkazy:

na stos liczbę 300
 na stos adres *a*
 na stos wart.zm. *a*
 nierówność
 większość

• • •
 dodawanie
 odejmowanie

• • •
 do *E*₂
 jeśli na stosie *false*
 to do *E*
 jeśli na stosie *true*
 to do *E*

PRAWDZIWY JĘZYK WEWNĘTRZNY

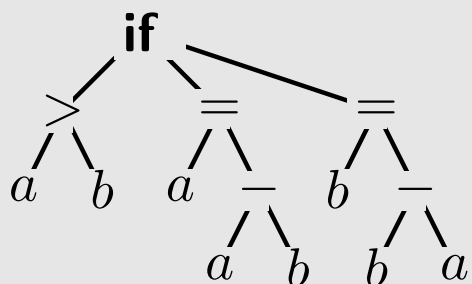
typowe rozkazy:

pobierz spod adr. 300
 włóż pod adres 300
 dodaj spod adr. 300
 odejmij spod adr. 300

• • •
 zwiększ [308] o 4
 zmniejsz [308] o 4

Drugi etap tłumaczenia

DRZEWA:



ABSTRAKCYJNA MASZYNA STOSOWA

typowe rozkazy:

na stos liczbę 300
na stos adres *a*
na stos wart.zm. *a*
nierówność
większość

• • •
dodawanie
odejmowanie

• • •
do *E*₂
jeśli na stosie *false*
to do *E*
jeśli na stosie *true*
to do *E*

PRAWDZIWY JĘZYK WEWNĘTRZNY

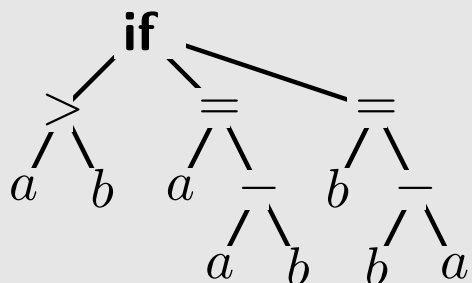
typowe rozkazy:

pobierz spod adr. 300
włóż pod adres 300
dodaj spod adr. 300
odejmij spod adr. 300

• • •
zwiększ [308] o 4
zmniejsz [308] o 4
do *E*₂

Drugi etap tłumaczenia

DRZEWA:



ABSTRAKCYJNA MASZYNA STOSOWA

typowe rozkazy:

na stos liczbę 300
na stos adres *a*
na stos wart.zm. *a*
nierówność
większość

• • •
dodawanie
odejmowanie

• • •
do *E*₂
jeśli na stosie *false*
to do *E*
jeśli na stosie *true*
to do *E*

PRAWDZIWY JĘZYK WEWNĘTRZNY

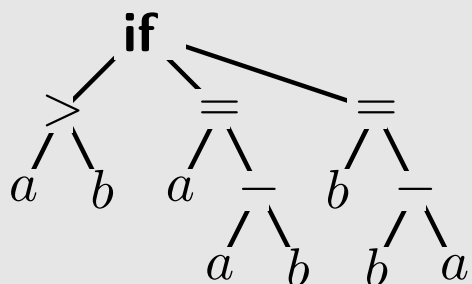
typowe rozkazy:

pobierz spod adr. 300
włóż pod adres 300
dodaj spod adr. 300
odejmij spod adr. 300

• • •
zwiększ [308] o 4
zmniejsz [308] o 4
do *E*₂
czy zero pod adr. 300
czy dodatnie pod adr. 300
• • •

Drugi etap tłumaczenia

DRZEWA:



ABSTRAKCYJNA MASZYNA STOSOWA

typowe rozkazy:

na stos liczbę 300
na stos adres *a*
na stos wart.zm. *a*
nierówność
większość

• • •
dodawanie
odejmowanie

• • •
do *E*₂
jeśli na stosie *false*
to do *E*
jeśli na stosie *true*
to do *E*

PRAWDZIWY JĘZYK WEWNĘTRZNY

typowe rozkazy:

pobierz spod adr. 300
włóż pod adres 300
dodaj spod adr. 300
odejmij spod adr. 300

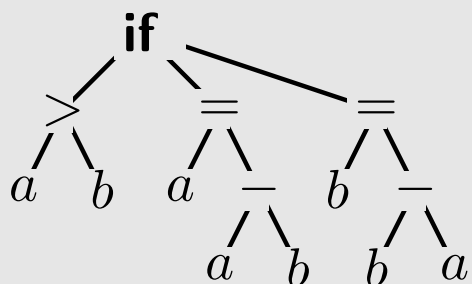
• • •
zwiększ [308] o 4
zmniejsz [308] o 4
do *E*₂

czy zero pod adr. 300
czy dodatnie pod adr. 300

• • •
jeśli tak to do *E*
jeśli nie to do *E*

Drugi etap tłumaczenia

DRZEWA:



ABSTRAKCYJNA MASZYNA STOSOWA

typowe rozkazy:

na stos liczbę 300
na stos adres a
na stos wart.zm. a
nierówność
większość

• • •
dodawanie
odejmowanie

• • •
do E_2
jeśli na stosie *false*
to do E
jeśli na stosie *true*
to do E

PRAWDZIWY JĘZYK WEWNĘTRZNY

typowe rozkazy:

pobierz spod adr. 300
włóż pod adres 300
dodaj spod adr. 300
odejmij spod adr. 300

• • •
zwiększ $[308]$ o 4
zmniejsz $[308]$ o 4
do E_2

czy zero pod adr. 300
czy dodatnie pod adr. 300

• • •
jeśli tak to do E
jeśli nie to do E

modyfikacja adresu: $+ [300]$

Język stosowy $\longrightarrow$ język wewnętrzny

PROGRAM STOSOWY:

```

E1: na stos wart.zm. a;
      na stos wart.zm. b;
      nierówność;
      jeśli na stosie false
          to do E2;
      na stos wart.zm. a;
      na stos wart.zm. b;
      większość;
      jeśli na stosie false
          to do E3;
      na stos adres a;
      na stos wart.zm. a;
      na stos wart.zm. b;
      odejmowanie;
      przypisanie;
      do E4;
E3: na stos adres b;
      na stos wart.zm. b;
      na stos wart.zm. a;
      odejmowanie;
      przypisanie;
E4: do E1;
E2:

```

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 większość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;

E_3 : na stos adres b ;
 na stos wart.zm. b ;
 na stos wart.zm. a ;
 odejmowanie;
 przypisanie;

E_4 : do E_1 ;

E_2 :

PROGRAM W JĘZ.WEWN.:

S T O S		324
		320
		316
		312
	n	0 308
	b	11 304
	a	20 300

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

```

E1: na stos wart.zm. a;
      na stos wart.zm. b;
      nierówność;
      jeśli na stosie false
          to do E2;
      na stos wart.zm. a;
      na stos wart.zm. b;
      większość;
      jeśli na stosie false
          to do E3;
      na stos adres a;
      na stos wart.zm. a;
      na stos wart.zm. b;
      odejmowanie;
      przypisanie;
      do E4;
E3: na stos adres b;
      na stos wart.zm. b;
      na stos wart.zm. a;
      odejmowanie;
      przypisanie;
E4: do E1;
E2:
  
```

PROGRAM W JĘZ.WEWN.:

```

zwiększ [308] o 4
pobierz spod adr. 300
włóż pod adres 308+[308]
  
```

		324
		320
		316
		312
		308
<i>n</i>	0	308
<i>b</i>	11	304
<i>a</i>	20	300

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

```

E1: na stos wart.zm. a;
      na stos wart.zm. b;
      nierówność;
      jeśli na stosie false
          to do E2;
      na stos wart.zm. a;
      na stos wart.zm. b;
      większość;
      jeśli na stosie false
          to do E3;
      na stos adres a;
      na stos wart.zm. a;
      na stos wart.zm. b;
      odejmowanie;
      przypisanie;
      do E4;
E3: na stos adres b;
      na stos wart.zm. b;
      na stos wart.zm. a;
      odejmowanie;
      przypisanie;
E4: do E1;
E2:
  
```

PROGRAM W JĘZ.WEWN.:

```

zwiększ [308] o 4
pobierz spod adr. 300
włóż pod adres 308+[308]
  
```

S T O S		324
		320
		316
	20	312
	<i>n</i>	4 308
	<i>b</i>	11 304
	<i>a</i>	20 300

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

```

E1: na stos wart.zm. a;
    na stos wart.zm. b;
    nierówność;
    jeśli na stosie false
        to do E2;
    na stos wart.zm. a;
    na stos wart.zm. b;
    większość;
    jeśli na stosie false
        to do E3;
    na stos adres a;
    na stos wart.zm. a;
    na stos wart.zm. b;
    odejmowanie;
    przypisanie;
    do E4;
E3: na stos adres b;
    na stos wart.zm. b;
    na stos wart.zm. a;
    odejmowanie;
    przypisanie;
E4: do E1;
E2:

```

S T O S		324
		320
		316
	20	312
	n	4
	b	11
	a	20
		300

PROGRAM W JĘZ.WEWN.:

```

zwiększ [308] o 4
pobierz spod adr. 300
włóż pod adres 308+[308]

```

```

zwiększ [308] o 4
pobierz spod adr. 304
włóż pod adres 308+[308]

```


Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 większość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;
 E_3 : na stos adres b ;
 na stos wart.zm. b ;
 na stos wart.zm. a ;
 odejmowanie;
 przypisanie;
 E_4 : do E_1 ;
 E_2 :

S T O S		324
		320
	11	316
	20	312
	n	308
	b	304
	a	300

PROGRAM W JĘZ.WEWN.:

zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

 zwiększ $[308]$ o 4
 pobierz spod adr. 304
 włóż pod adres $308+[308]$

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

```

E1: na stos wart.zm. a;
      na stos wart.zm. b;
      nierówność;
      jeśli na stosie false
          to do E2;
      na stos wart.zm. a;
      na stos wart.zm. b;
      większość;
      jeśli na stosie false
          to do E3;
      na stos adres a;
      na stos wart.zm. a;
      na stos wart.zm. b;
      odejmowanie;
      przypisanie;
      do E4;
E3: na stos adres b;
      na stos wart.zm. b;
      na stos wart.zm. a;
      odejmowanie;
      przypisanie;
E4: do E1;
E2:
  
```

S T O S			324
			320
	11		316
	20		312
	<i>n</i>	8	308
	<i>b</i>	11	304
	<i>a</i>	20	300

PROGRAM W JĘZ.WEWN.:

```

zwiększ [308] o 4
pobierz spod adr. 300
włóż pod adres 308+[308]
-----
zwiększ [308] o 4
pobierz spod adr. 304
włóż pod adres 308+[308]
-----
zmniejsz [308] o 4
pobierz spod adr. 308+[308]
odejmij spod adr. 312+[308]
włóż pod adres 308+[308]
  
```

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

```

E1: na stos wart.zm. a;
     na stos wart.zm. b;
     nierówność;
     jeśli na stosie false
         to do E2;
     na stos wart.zm. a;
     na stos wart.zm. b;
     większość;
     jeśli na stosie false
         to do E3;
     na stos adres a;
     na stos wart.zm. a;
     na stos wart.zm. b;
     odejmowanie;
     przypisanie;
     do E4;
E3: na stos adres b;
     na stos wart.zm. b;
     na stos wart.zm. a;
     odejmowanie;
     przypisanie;
E4: do E1;
E2:

```

S T O S		324
		320
	11	316
	9	312
	n	4
	b	11
	a	20
		300

PROGRAM W JĘZ.WEWN.:

```

zwiększ [308] o 4
pobierz spod adr. 300
włóż pod adres 308+[308]
-----
zwiększ [308] o 4
pobierz spod adr. 304
włóż pod adres 308+[308]
-----
zmniejsz [308] o 4
pobierz spod adr. 308+[308]
odejmij spod adr. 312+[308]
włóż pod adres 308+[308]

```

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 większość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;
 E_3 : na stos adres b ;
 na stos wart.zm. b ;
 na stos wart.zm. a ;
 odejmowanie;
 przypisanie;
 E_4 : do E_1 ;
 E_2 :

		324
		320
	11	316
	9	312
n	4	308
b	11	304
a	20	300

PROGRAM W JĘZ.WEWN.:

zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

 zwiększ $[308]$ o 4
 pobierz spod adr. 304
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 pobierz spod adr. $308+[308]$
 odejmij spod adr. $312+[308]$
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 czy zero pod adr. $312+[308]$
 jeśli tak to do E_2

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 większość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;
 E_3 : na stos adres b ;
 na stos wart.zm. b ;
 na stos wart.zm. a ;
 odejmowanie;
 przypisanie;
 E_4 : do E_1 ;
 E_2 :

		324
		320
	11	316
	9	312
S		
T		
O		
S		
n	0	308
b	11	304
a	20	300

PROGRAM W JĘZ.WEWN.:

zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

 zwiększ $[308]$ o 4
 pobierz spod adr. 304
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 pobierz spod adr. $308+[308]$
 odejmij spod adr. $312+[308]$
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 czy zero pod adr. $312+[308]$
 jeśli tak to do E_2

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 większość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;
 E_3 : na stos adres b ;
 na stos wart.zm. b ;
 na stos wart.zm. a ;
 odejmowanie;
 przypisanie;
 E_4 : do E_1 ;
 E_2 :

			324
			320
		11	316
		9	312
	n	0	308
	b	11	304
	a	20	300

PROGRAM W JĘZ.WEWN.:

zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

zwiększ $[308]$ o 4
 pobierz spod adr. 304
 włóż pod adres $308+[308]$

zmniejsz $[308]$ o 4
 pobierz spod adr. $308+[308]$
 odejmij spod adr. $312+[308]$
 włóż pod adres $308+[308]$

zmniejsz $[308]$ o 4
 czy zero pod adr. $312+[308]$
 jeśli tak to do E_2

zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 większość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;
 E_3 : na stos adres b ;
 na stos wart.zm. b ;
 na stos wart.zm. a ;
 odejmowanie;
 przypisanie;
 E_4 : do E_1 ;
 E_2 :

S T O S		324
		320
	11	316
	20	312
	n	308
	b	304
	a	300

PROGRAM W JĘZ.WEWN.:

zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

 zwiększ $[308]$ o 4
 pobierz spod adr. 304
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 pobierz spod adr. $308+[308]$
 odejmij spod adr. $312+[308]$
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 czy zero pod adr. $312+[308]$
 jeśli tak to do E_2

 zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 większość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;
 E_3 : na stos adres b ;
 na stos wart.zm. b ;
 na stos wart.zm. a ;
 odejmowanie;
 przypisanie;
 E_4 : do E_1 ;
 E_2 :

S T O S		324
		320
	11	316
	20	312
	n	308
	b	304
	a	300

PROGRAM W JĘZ.WEWN.:

zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

 zwiększ $[308]$ o 4
 pobierz spod adr. 304
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 pobierz spod adr. $308+[308]$
 odejmij spod adr. $312+[308]$
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 czy zero pod adr. $312+[308]$
 jeśli tak to do E_2

 zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

 zwiększ $[308]$ o 4
 pobierz spod adr. 304
 włóż pod adres $308+[308]$

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 większość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;
 E_3 : na stos adres b ;
 na stos wart.zm. b ;
 na stos wart.zm. a ;
 odejmowanie;
 przypisanie;
 E_4 : do E_1 ;
 E_2 :

S T O S		324
		320
	11	316
	20	312
n	8	308
b	11	304
a	20	300

PROGRAM W JĘZ.WEWN.:

```

zwiększ [308] o 4
pobierz spod adr. 300
włóż pod adres 308+[308]
-----
zwiększ [308] o 4
pobierz spod adr. 304
włóż pod adres 308+[308]
-----
zmniejsz [308] o 4
pobierz spod adr. 308+[308]
odejmij spod adr. 312+[308]
włóż pod adres 308+[308]
-----
zmniejsz [308] o 4
czy zero pod adr. 312+[308]
jeśli tak to do  $E_2$ 
-----
zwiększ [308] o 4
pobierz spod adr. 300
włóż pod adres 308+[308]
-----
zwiększ [308] o 4
pobierz spod adr. 304
włóż pod adres 308+[308]
  
```

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 zwiększość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;
 E_3 : na stos adres b ;
 na stos wart.zm. b ;
 na stos wart.zm. a ;
 odejmowanie;
 przypisanie;
 E_4 : do E_1 ;
 E_2 :

S T O S		324
		320
	11	316
	20	312
n	8	308
b	11	304
a	20	300

PROGRAM W JĘZ.WEWN.:

zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

 zwiększ $[308]$ o 4
 pobierz spod adr. 304
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 pobierz spod adr. $308+[308]$
 odejmij spod adr. $312+[308]$
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 czy zero pod adr. $312+[308]$
 jeśli tak to do E_2

 zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

 zwiększ $[308]$ o 4
 pobierz spod adr. 304
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 pobierz spod adr. $308+[308]$
 odejmij spod adr. $312+[308]$
 włóż pod adres $308+[308]$

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 zwiększość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;
 E_3 : na stos adres b ;
 na stos wart.zm. b ;
 na stos wart.zm. a ;
 odejmowanie;
 przypisanie;
 E_4 : do E_1 ;
 E_2 :

S T O S		324
		320
	11	316
	9	312
	n	308
	b	304
	a	300

PROGRAM W JĘZ.WEWN.:

zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

 zwiększ $[308]$ o 4
 pobierz spod adr. 304
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 pobierz spod adr. $308+[308]$
 odejmij spod adr. $312+[308]$
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 czy zero pod adr. $312+[308]$
 jeśli tak to do E_2

 zwiększ $[308]$ o 4
 pobierz spod adr. 300
 włóż pod adres $308+[308]$

 zwiększ $[308]$ o 4
 pobierz spod adr. 304
 włóż pod adres $308+[308]$

 zmniejsz $[308]$ o 4
 pobierz spod adr. $308+[308]$
 odejmij spod adr. $312+[308]$
 włóż pod adres $308+[308]$

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

```

E1: na stos wart.zm. a;
      na stos wart.zm. b;
      nierówność;
      jeśli na stosie false
          to do E2;
      na stos wart.zm. a;
      na stos wart.zm. b;
      większość;
      jeśli na stosie false
          to do E3;
      na stos adres a;
      na stos wart.zm. a;
      na stos wart.zm. b;
      odejmowanie;
      przypisanie;
      do E4;
E3: na stos adres b;
      na stos wart.zm. b;
      na stos wart.zm. a;
      odejmowanie;
      przypisanie;
E4: do E1;
E2:

```

S T O S		324
		320
	11	316
	9	312
	n	308
	b	304
	a	300

PROGRAM W JĘZ.WEWN.:

```

zwiększ [308] o 4
pobierz spod adr. 300
włóż pod adres 308+[308]
-----
zwiększ [308] o 4
pobierz spod adr. 304
włóż pod adres 308+[308]
-----
zmniejsz [308] o 4
pobierz spod adr. 308+[308]
odejmij spod adr. 312+[308]
włóż pod adres 308+[308]
-----
zmniejsz [308] o 4
czy zero pod adr. 312+[308]
jeśli tak to do E2
-----
zwiększ [308] o 4
pobierz spod adr. 300
włóż pod adres 308+[308]
-----
zwiększ [308] o 4
pobierz spod adr. 304
włóż pod adres 308+[308]
-----
zmniejsz [308] o 4
pobierz spod adr. 308+[308]
odejmij spod adr. 312+[308]
włóż pod adres 308+[308]
-----

```

...

Język stosowy → język wewnętrzny

PROGRAM STOSOWY:

E_1 : na stos wart.zm. a ;
 na stos wart.zm. b ;
 nierówność;
 jeśli na stosie *false*
 to do E_2 ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 większość;
 jeśli na stosie *false*
 to do E_3 ;
 na stos adres a ;
 na stos wart.zm. a ;
 na stos wart.zm. b ;
 odejmowanie;
 przypisanie;
 do E_4 ;
 E_3 : na stos adres b ;
 na stos wart.zm. b ;
 na stos wart.zm. a ;
 odejmowanie;
 przypisanie;
 E_4 : do E_1 ;
 E_2 :

		324
		320
	11	316
	9	312
n	4	308
b	11	304
a	20	300

PROGRAM W JĘZ.WEWN.:

zwiększ [308] o 4
 pobierz spod adr. 300
 włóż pod adres 308+[308]
 zwiększ [308] o 4
 pobierz spod adr. 304
 włóż pod adres 308+[308]
 zmniejsz [308] o 4
 pobierz spod adr. 308+[308]
 odejmij spod adr. 312+[308]
 włóż pod adres 308+[308]
 zmniejsz [308] o 4
 czy zero pod adr. 312+[308]
 jeśli tak to do E_2
 zwiększ [308] o 4
 pobierz spod adr. 300
 włóż pod adres 308+[308]
 zwiększ [308] o 4
 pobierz spod adr. 304
 włóż pod adres 308+[308]
 zmniejsz [308] o 4
 pobierz spod adr. 308+[308]
 odejmij spod adr. 312+[308]
 włóż pod adres 308+[308]

...