Politechnika Wrocławska

Wstęp do programowania

INP001213Wcl

rok akademicki 2018/19

semestr zimowy

Wykład 12

Karol Tarnowski

karol.tarnowski@pwr.edu.pl

A-1 p. 411B

Politechnika Wrocławska

Plan prezentacji

- Własności algorytmów
- Poprawność algorytmów
- Skończoność algorytmów

Na podstawie:

- M. M. Sysło, *Algorytmy*
- D. Harel, *Rzecz o istocie algorytmiki*

Politechnika Wrocławska

Własności algorytmów

Najważniejsze własności można podzielić na trzy grupy:

- poprawność
- skończoność
- efektywność

Politechnika Wrocławska

Poprawność algorytmów

Algorytm poprawny – rozwiązuje problem, dla którego został opracowany. Poprawność algorytmu określamy względem specyfikacji problemu.

Politechnika Wrocławska

Poprawność algorytmów

Specyfikacja – ścisła definicja problemu, składa się z opisu danych, wyników oraz wiążących je relacji.

Poprawność algorytmów

Specyfikacja - przykład

Problem przeszukiwania zbioru uporządkowanego

Dane (I): uporządkowany ciąg liczb w tablicy $a[k..l]$, gdzie $k \le l$, $a_k \le a_{k+1} \le \ldots \le a_l$; oraz element y: $a_k \le y \le a_l$.

Wynik (R): indeks s ($k \le s \le l$) oraz $a_s = y$, lub $s = -1$, gdy $y \ne a_i$, dla każdego i ($k \le i \le l$).

Politechnika Wrocławska

Poprawność algorytmów

Częściowa poprawność

Algorytm jest częściowo poprawny jeśli dla każdych obliczeń, które się kończą, wynik jest poprawny względem warunków początkowego i końcowego.

Politechnika Wrocławska

Poprawność algorytmów

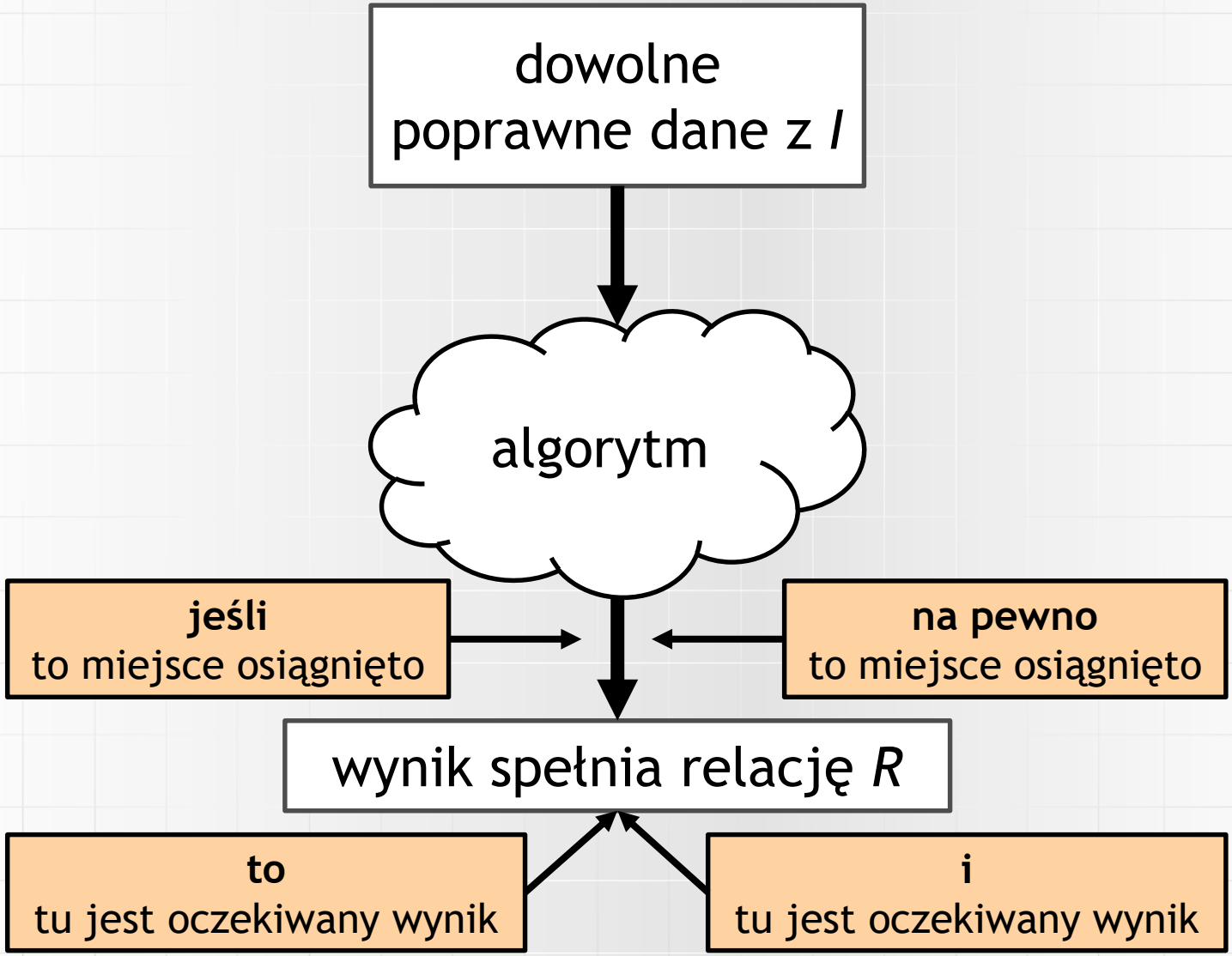
Całkowita poprawność

Algorytm jest całkowicie poprawny względem warunków początkowego i końcowego, gdy dla wszystkich danych zgodnych ze specyfikacją obliczenia algorytmu kończą się i wyniki spełniają warunek końcowy.

Poprawność algorytmów

częściowa poprawność

całkowita poprawność



Politechnika Wrocławska

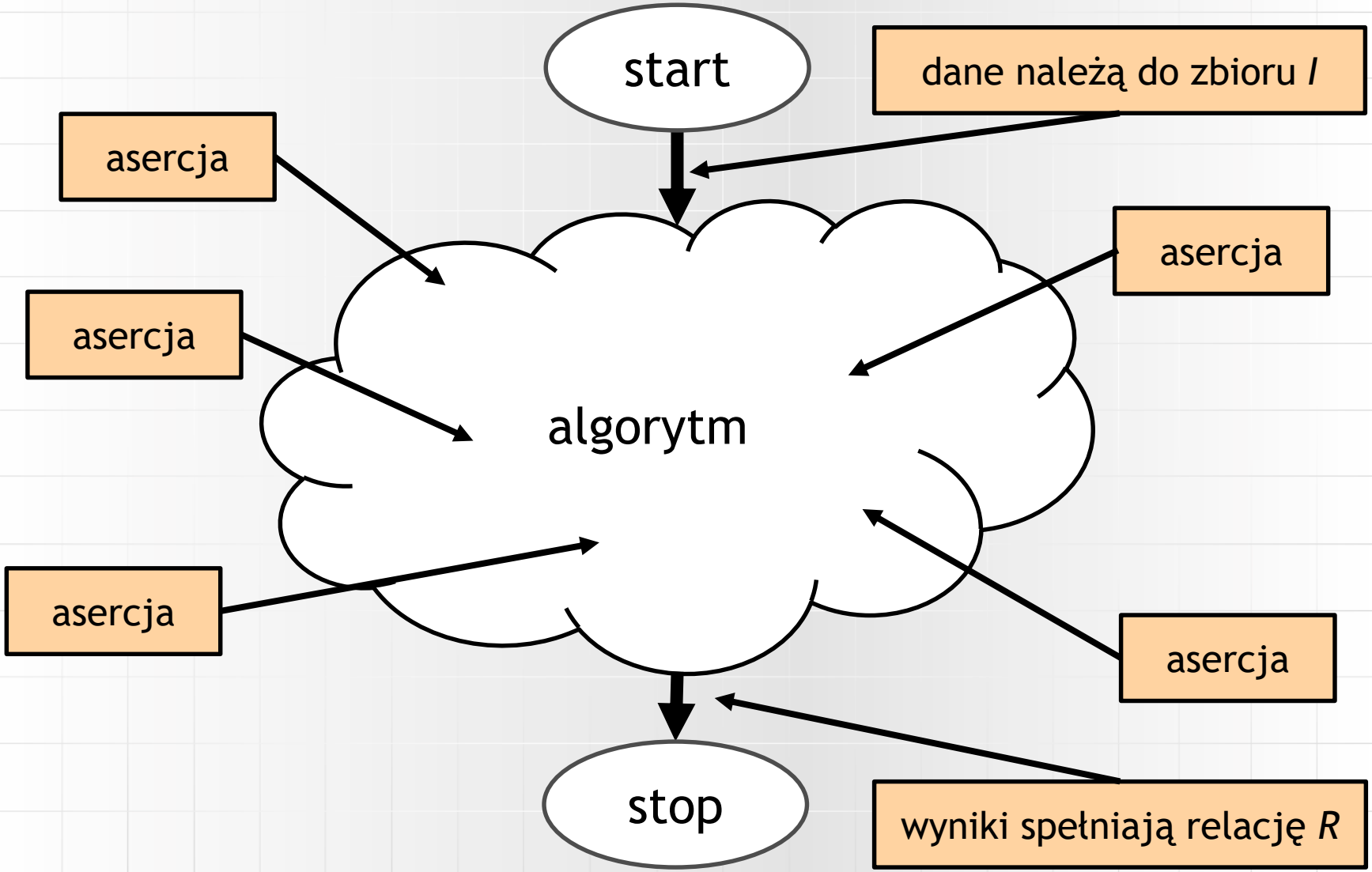
Poprawność algorytmów

Dobra określoność opisu

Wszystkie występujące w opisie polecenie i operacja są zrozumiałe, kolejność wykonywania działań nie pozostawia wątpliwości.

Poprawność algorytmów

Metoda niezmienników



Poprawność algorytmów

Dowodzenie poprawności - przykład

Problem przeszukiwania zbioru uporządkowanego

Dane (I): uporządkowany ciąg liczb w tablicy $a[k..l]$, gdzie $k \le l$, $a_k \le a_{k+1} \le \ldots \le a_l$; oraz element y: $a_k \le y \le a_l$.

Wynik (R): indeks s ($k \le s \le l$) oraz $a_s = y$, lub $s = -1$, gdy $y \ne a_i$, dla każdego i ($k \le i \le l$).

Krok 1. *lewy* = k, *prawy* = l;

Krok 2. Jeśli *lewy* > *prawy*, to $s = -1$ i zakończ;

Krok 3. s = (*lewy* + *prawy*) / 2;

Jeśli $a_s = y$ to zakończ;

Jeśli $a_s < y$, to *lewy* = $s + 1$, w p.p. *prawy* = $s - 1$;

Wróć do kroku 2.

Poprawność algorytmów

Dowodzenie poprawności – przykład

Wykazując, że:

1. dla każdego wykonania algorytmu, które rozpoczyna się z danymi należącymi do zbioru *I*, jeśli obliczenia docierają do końca, to wynik spełnia warunek końcowy *R*;
2. dla wszystkich danych należących do *I*, obliczenia się kończą;

pokażemy **całkowitą poprawność** algorytmu.

Poprawność algorytmów

Dowodzenie poprawności - przykład

W dowodzie poprawności działania algorytmu posłużymy się następującym niezmiennikiem:

jeśli istnieje *s* takie, że $lewy \leq s \leq prawy$ i $a_s = y$, to $a_{lewy} \leq y \leq a_{prawy}$

(jeśli *y* jest w przeszukiwanej tablicy, to jest w tej części, która jest jeszcze przeszukiwana)

Poprawność algorytmów

Dowodzenie poprawności - przykład

jeśli istnieje *s* takie, że $lewy \leq s \leq prawy$ i $a_s = y$, to $a_{lewy} \leq y \leq a_{prawy}$

1. Czy warunek jest spełniony przed wykonaniem pierwszej iteracji kroków 2 i 3?
2. Jeśli warunek jest prawdziwy przed wykonaniem iteracji kroków 2 i 3, to po zakończeniu iteracji pozostaje prawdziwy.
3. Algorytm kończy działanie, czyli kończy się iteracja kroków 2 i 3, i wynik spełnia warunek końcowy.

Poprawność algorytmów

Dowodzenie poprawności - przykład

jeśli istnieje *s* takie, że $lewy \le s \le prawy$ i $a_s = y$, to $a_{lewy} \le y \le a_{prawy}$

1. Czy niezmiennik jest spełniony przed wykonaniem pierwszej iteracji kroków 2 i 3?

Dane (*I*):	uporządkowany ciąg liczb w tablicy $a[k..l]$, gdzie $k \le l$, $a_k \le a_{k+1} \le \ldots \le a_l$; oraz element y: $a_k \le y \le a_l$.
Wynik (*R*):	indeks s ($k \le s \le l$) oraz $a_s = y$, lub $s = -1$, gdy $y \ne a_i$, dla każdego i ($k \le i \le l$).
Krok 1.	$lewy = k$, $prawy = l$;
…	

Poprawność algorytmów

Dowodzenie poprawności - przykład

jeśli istnieje *s* takie, że *lewy* ≤ *s* ≤ *prawy* i $a_s = y$, to $a_{lewy} \leq y \leq a_{prawy}$

2. Jeśli niezmiennik jest prawdziwy przed wykonaniem iteracji kroków 2 i 3, to po zakończeniu iteracji pozostaje prawdziwy.

Krok 2. Jeśli *lewy* > *prawy*, to *s* =-1 i zakończ;

Krok 3. *s* = (*lewy* + *prawy*) / 2;

Jeśli $a_s = y$ to zakończ;

Jeśli $a_s < y$, to *lewy* = *s* + 1, w p.p. *prawy* = *s* – 1;

Wróć do kroku 2.

Poprawność algorytmów

Dowodzenie poprawności - przykład

jeśli istnieje s takie, że $lewy \le s \le prawy$ i $a_s = y$, to $a_{lewy} \le y \le a_{prawy}$

3. Algorytm kończy działanie, czyli kończy się iteracja kroków 2 i 3, i wynik spełnia warunek końcowy.

Wynik (R): indeks s ($k \le s \le l$) oraz $a_s = y$, lub $s = -1$, gdy $y \ne a_i$, dla każdego i ($k \le i \le l$).

Krok 2. Jeśli $lewy > prawy$, to s =-1 i zakończ;

Krok 3. $s = (lewy + prawy) / 2$;

Jeśli $a_s = y$ to zakończ;

Jeśli $a_s < y$, to $lewy = s + 1$, w p.p. $prawy = s - 1$;

Wróć do kroku 2.

Poprawność algorytmów

Dowodzenie poprawności - przykład 2

Dane: ciąg znaków *S*

Wynik: ciąg znaków *Y* będący odwróconym ciągiem *S*

przykładowo:

odwrócone("Elf układał kufle") = "elfuk ładałku flE"

odwrócone("ajjQdt8") = "8tdQjja"

Politechnika Wrocławska

Poprawność algorytmów

Dowodzenie poprawności - przykład 2

funkcje pomocnicze:

głowa("ajjQdt8") = "a"

ogon("ajjQdt8") = "jjQdt8"

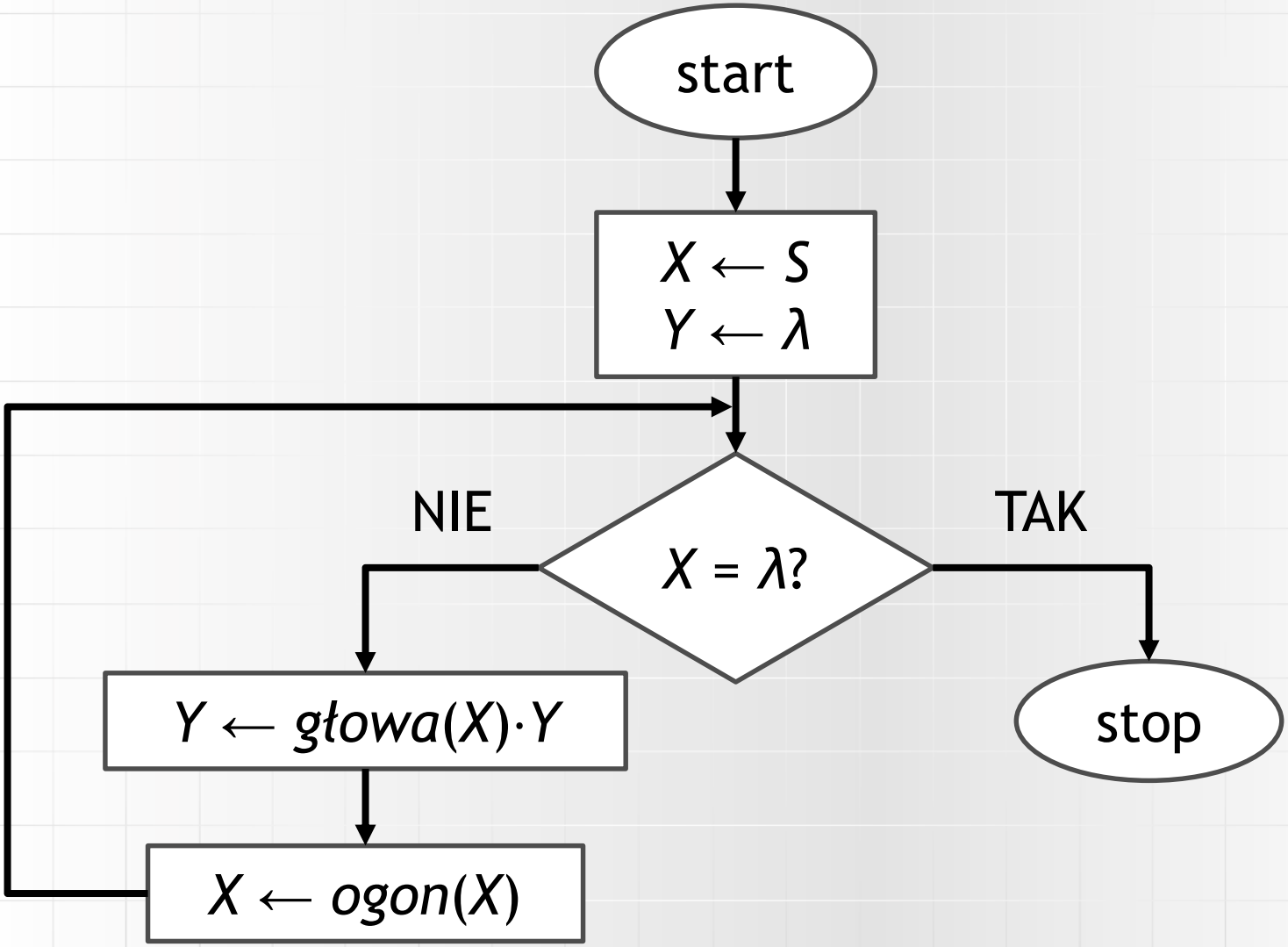
operacja konkatenacji ·

"ajjQdt8" · „tdd54q" = "ajjQdt8tdd54q"

łańcuch pusty λ

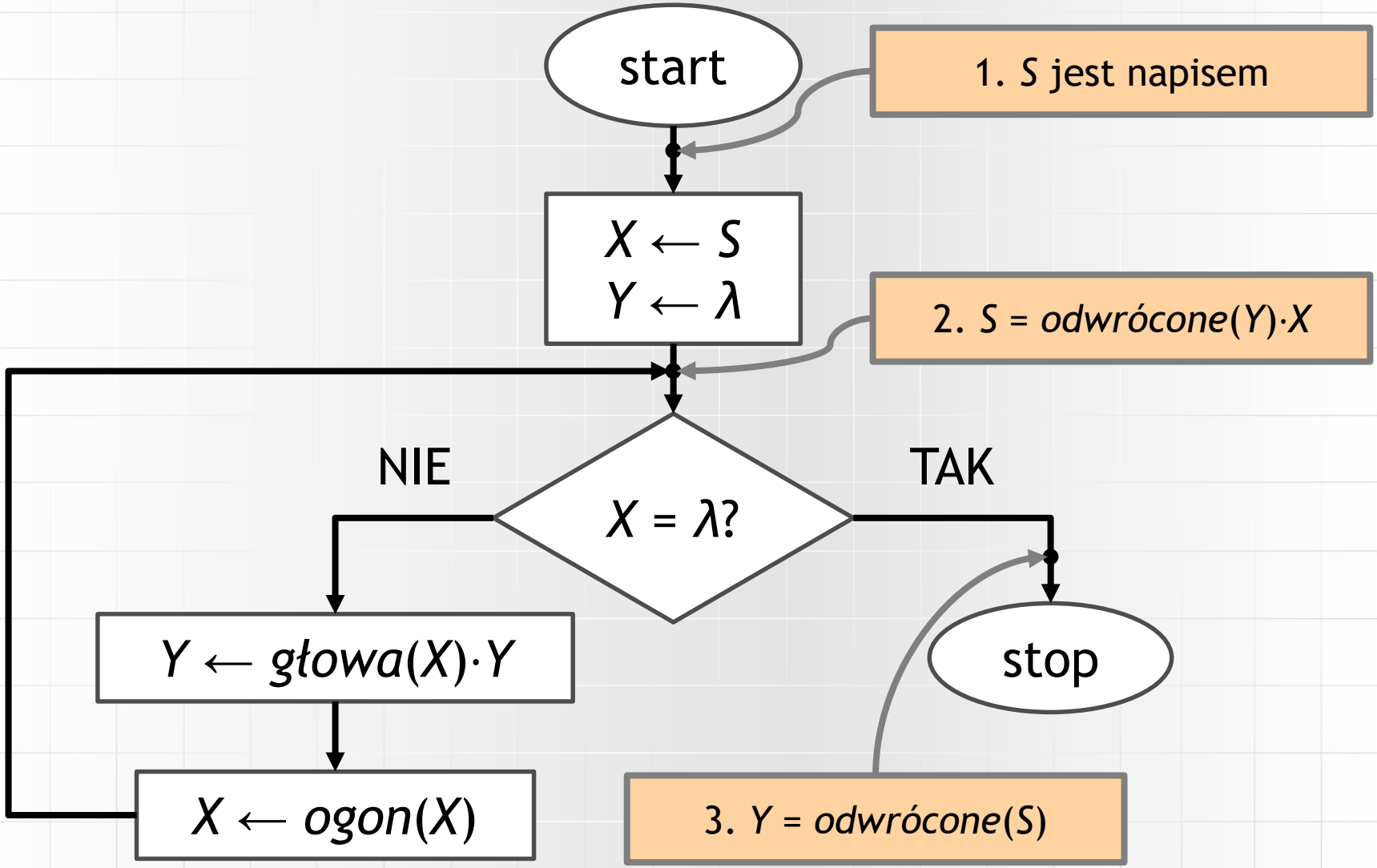
Poprawność algorytmów

Dowodzenie poprawności - przykład 2



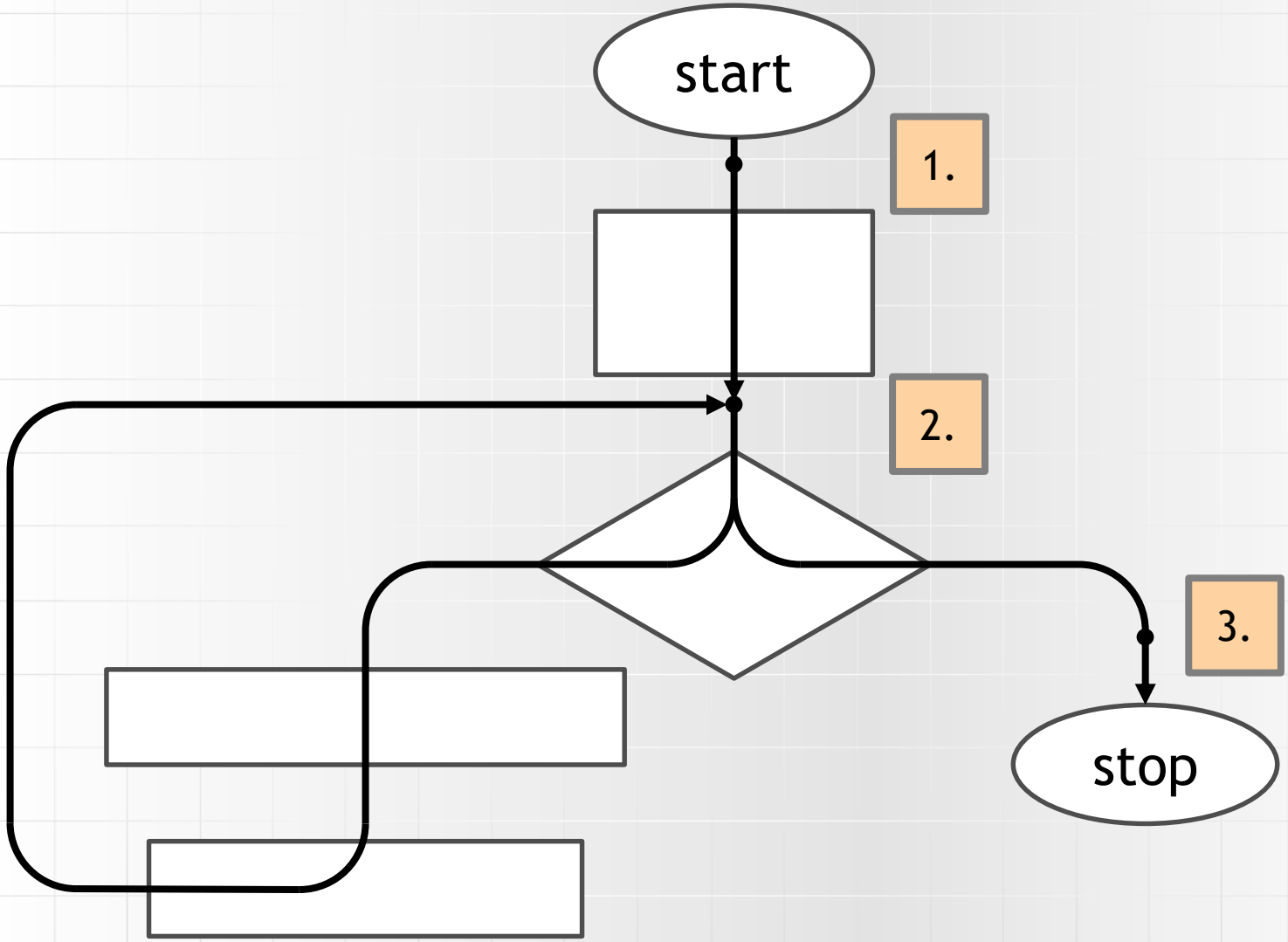
Poprawność algorytmów

Dowodzenie poprawności - przykład 2



Poprawność algorytmów

Dowodzenie poprawności - przykład 2



Poprawność algorytmów

Dowodzenie poprawności - przykład 2

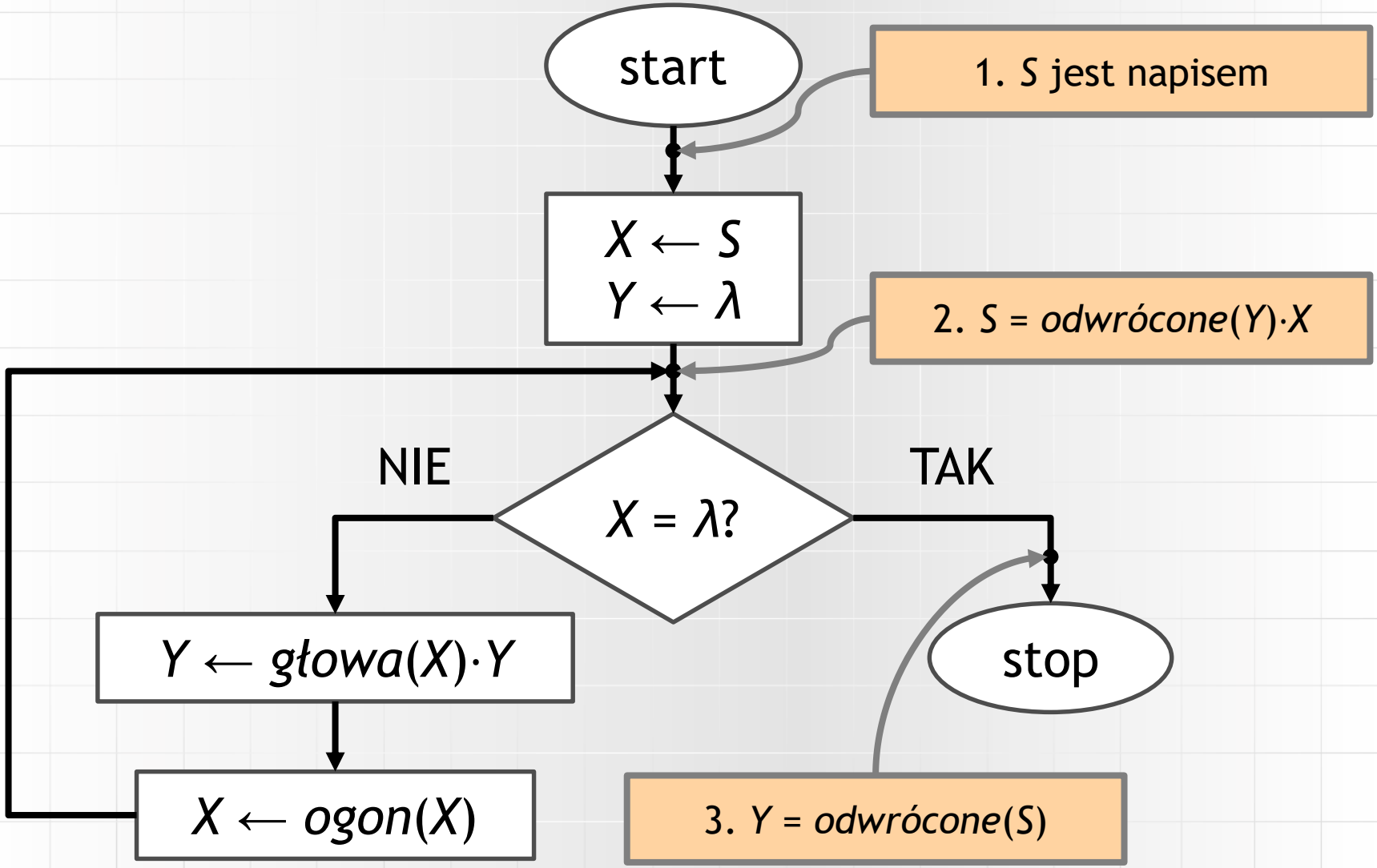
(1→2) dla każdego napisu S, po wykonaniu dwóch instrukcji $X \leftarrow S$, $Y \leftarrow \lambda$ będzie spełniona równość $S = \mathit{odwrócone}(Y) \cdot X$

(2→3) jeśli $S = \text{odwrócone}(Y) \cdot X$ i $X = \lambda$ to $S = \mathit{odwrócone}(Y)$

(2→2) jeśli $S = \mathit{odwrócone}(Y) \cdot X$ i $X \neq \lambda$ to po wykonaniu instrukcji $Y \leftarrow \mathit{głowa}(X) \cdot Y$, $X \leftarrow \mathit{ogon}(X)$, ta równość będzie prawdziwa dla nowych X oraz Y

Poprawność algorytmów

Dowodzenie poprawności - przykład 2



Poprawność algorytmów

Dowodzenie poprawności - przykład 2

(2→2) jeśli $S = odwrócone(Y)\cdot X$ i $X \neq \lambda$ to po wykonaniu instrukcji $Y \leftarrow głowa(X)\cdot Y$, $X \leftarrow ogon(X)$, ta równość będzie prawdziwa dla nowych X oraz Y

Należy uzasadnić, że dla dowolnych S, Y, oraz X (niepustego), jeśli

$S = odwrócone(Y)\cdot X$

to

$S = odwrócone(głowa(X)\cdot Y)\cdot ogon(X)$

Poprawność algorytmów

Dowodzenie poprawności - przykład 2

Należy uzasadnić, że dla dowolnych S, Y, oraz X (niepustego), jeśli

$S = odwrócone(Y) \cdot X$

to

$S = odwrócone(głowa(X) \cdot Y) \cdot ogon(X)$

$odwrócone(głowa(X) \cdot Y) \cdot ogon(X) = odwrócone(Y) \cdot X$

Poprawność algorytmów

Dowodzenie poprawności - przykład 2

odwrócone(*głowa*(*X*)·*Y*)·*ogon*(*X*) = *odwrócone*(*Y*)·*X*

Y: `ROGLA` *X*: `YTMIKA`

odwrócone(*Y*)·*X*: `ALGOR` `YTMIKA`

odwrócone(*głowa*(*X*)·*Y*)·*ogon*(*X*) `ALGORY` `TMIKA`

głowa(*X*)·*Y*: `YROGLA` *ogon*(*X*): `TMIKA`

głowa(*X*): `Y` *Y*: `ROGLA`

Politechnika Wrocławska

Poprawność algorytmów

Dowodzenie poprawności - przykład 2

Zbieżnikiem może być długość napisu *X*.

Politechnika Wrocławska

Poprawność algorytmów

Poprawność programu

Program może zawierać błędy składniowe i błędy logiczne

- błędy składniowe – niezgodność ze składnią języka,
- błędy logiczne – nie dla wszystkich poprawnych danych, program generuje poprawne wyniki.

Komputery są nieomylne – jeśli program zwraca niepoprawne wyniki, to albo podano nieprawidłowe dane, albo program zawiera błąd.

Politechnika Wrocławska

Poprawność algorytmów

Uruchamianie i testowanie programów

Najprostsza metoda, polega na testowaniu poprawności programu z zastosowaniem danych testowych, dla których wiemy jak ma działać i jaki dać wynik.

Technika ta pozwala znaleźć błędy, ale jej użycie nie wystarcza do wykazania poprawności programów.

Politechnika Wrocławska

Skończoność

Badanie skończoności działania algorytmu jest częścią dowodzenia całkowitej poprawności.

Powodem nie zatrzymywania się programu może być sprawdzanie spełnienia dokładnych równości wielkości, które mogą przyjmować niedokładne wartości.

Skończoność

Przykłady

Ciąg określony rekurencyjnie:

c_0 – liczba naturalna

$$c_{n+1} = \begin{cases} \frac{1}{2} c_n & \text{gdy } c_n \text{ jest parzysta} \\ 3c_n + 1 & \text{gdy } c_n \text{ jest nieparzysta} \end{cases}$$

Politechnika Wrocławska

Podsumowanie

- Poprawność częściowa i całkowita
- Metoda niezmienników
- Przykłady zastosowań metody niezmienników
- Skończoność