

# Systemy komputerowe

## Lista zadań nr 3

Na ćwiczenia 23. i 24. marca 2022

Każde zadanie warte jest 1 punkt.

**Zadanie 1.** Dla poniższego programu narysuj graf przepływu sterowania. Następnie, dla każdej instrukcji  $l$ , podaj zbiory definicji osiagających tę instrukcję (ang. *Reaching Definition sets*), a dokładniej zbiory  $RD_{\circ}(l)$  oraz  $RD_{\bullet}(l)$ .

```
[x := 0]1
[y := 1]2
[i := 1]3
while [i < z]4 do
    [t := x + y]5
    [x := y]6
    [y := t]7
    [i := i + 1]8
od
[y := x]9
```

**Wskazówka:** Zbiory w tym zadaniu wylicz ręcznie, tak jak na slajdach 19-24 z pliku slides1.pdf.

**Zadanie 2.** Dla programu z poprzedniego zadania ułóż równania dla zbiorów  $RD_{\circ}(l)$  oraz  $RD_{\bullet}(l)$ , dla  $l \in \{1, \dots, 9\}$ . Następnie sprawdź, że zbiory wyliczone w poprzednim zadaniu spełniają te równania.

**Wskazówka:** Slajdy 27-31 z pliku slides1.pdf.

**Zadanie 3.** Rozwiąż równania z poprzedniego zadania posługując się algorytmem stałopunktowym. Porównaj otrzymane rozwiązanie z tym otrzymanym w zadaniu 1.

**Wskazówka:** Slajdy 32-33 z pliku slides1.pdf.

**Zadanie 4.**

1. W równaniach opisujących zbiory  $RD_{\circ}(l)$  oraz  $RD_{\bullet}(l)$  z poprzednich zadań wskaż wystąpienia funkcji  $kill_{RD}(\cdot)$  oraz  $gen_{RD}(\cdot)$ .
2. Do języka wprowadzamy instrukcję  $read(x)$  przypisującą do zmiennej  $x$  wartość przeczytaną z wejścia. Jak należy zdefiniować  $kill_{RD}(read(x))$  oraz  $gen_{RD}(read(x))$ ?
3. Załóżmy, że kompilator otrzymuje na wejściu programy w postaci grafu przepływu sterowania i potrzebuje obliczyć zbiory  $RD_{\circ}(l)$  oraz  $RD_{\bullet}(l)$ . W jaki sposób można to zautomatyzować?

**Wskazówka:** Slajdy 17-19 z pliku slides2.pdf.

**Zadanie 5.** Rozważmy poniższy program. Dla każdej instrukcji  $l \in \{1, \dots, 5\}$ , ułóż równania dla zbiorów  $RD_{\circ}(l)$  oraz  $RD_{\bullet}(l)$ . Wylicz ich rozwiązanie za pomocą algorytmu stałopunktowego. Porównaj otrzymane rozwiązanie z wartościami zbiorów  $RD_{\circ}(l)$  oraz  $RD_{\bullet}(l)$  które mogą pojawić się w faktycznym wykonaniu tego programu i wyciągnij wnioski.

```

[x := 1]1
if [x > 0]2 then
    [y := 1]3
else
    [y := -1]4
end
[z := y]5

```

**Wskazówka:** Slajdy 16, 24-26 z pliku slides1.pdf.

**Zadanie 6.** Zdefiniuj analizę zmiennych żywych (ang. *live variable analysis*). Dla programu z zadania 1. przeprowadź tę analizę w sposób formalny, tzn. zdefiniuj układ równań na zbiorach i rozwiąż go.

**Wskazówka:** Slajdy 32-37 z pliku slides2.pdf oraz r. 2.1.4 z "Principles of Program Analysis". W definicji funkcji  $kill/gen$  napis  $FV(a)$  zbiór wszystkich zmiennych występujących w wyrażeniu  $a$ .

**Zadanie 7.** Zdefiniuj analizę łańcuchów użycie-definicja (ang. *use-definition chains*). Wylicz wynik tej analizy dla programu

z zadania 1. posługując się uprzednio wyliczonymi zbiorami  $RD_0(l)$ .

**Wskazówka:** Slajdy 38-42 z pliku slides2.pdf.

**Zadanie 8.** Zdefiniuj analizę dostępnych wyrażeń (ang. *available expressions analysis*). Dla programu z zadania 1. zdefiniuj równania opisujące tę analizę. Spróbuj rozwiązać te równania dokładnie takim samym algorytmem stałopunktowym, co w poprzednich zadaniach. Dlaczego otrzymane rozwiązanie jest nieprawidłowe?

**Wskazówka:** Slajdy 10-16 z pliku slides2.pdf.