Systemy komputerowe

Lista zadań nr 4

Na ćwiczenia 30. i 31. marca 2022

Każde zadanie warte jest 1 punkt.

Zadanie 1. Używając algorytmu stałopunktowego (w wariancie obliczającym największy punkt stały) wylicz rozwiązania równań otrzymanych w zadaniu 8. z listy 3.

Zadanie 2. Zdefiniuj wariant analizy dostępnych wyrażeń liczący wyrażenie dostępne w konkretnej zmiennej: nietrywialne (tzn. różne od zmiennej) wyrażenie a jest dostępne w zmiennej x w etykiecie l, jeśli

a) zostało przypisane do tej zmiennej na wszystkich ścieżkach prowadzących do l, oraz
b) zmienna x oraz wartości zmiennych występujących w a nie uległy zmianie od tego czasu.

Zdefiniuj dziedzinę zbiorów występujących w tej analizie, funkcje $kill$ oraz gen. Podaj ogólną postać równań występujących w tej analizie, określ jej typ (may/must, backward/forward).

Zadanie 3. Podaj równania dla analizy z poprzedniego zadania oraz programu z listy 3. Rozwiąż otrzymany układ równań na zbiorach używając algorytmu stałopunktowego.

Zadanie 4. Przypomnij definicję zmiennej żywej z analizy zmiennych żywych. Rozważmy następujący program

$$[x := 1]^1; [x := x - 1]^2; [x := 2]^3$$

Zmienna x jest martwa (nie żywa) na wyjściu z etykiet 2 i 3. Natomiast x jest żywa na wyjściu z etykiety 1, mimo iż x jest użyta do obliczenia wartości zmiennej martwej. Powiemy, że zmienna jest *zemdlona* jeśli jest martwa lub jeśli jest używana wyłącznie do obliczenia wartości zmiennych zemdlonych. W przeciwnym przypadku zmienną nazwiemy *silnie żywą*. W powyższym przykładzie x jest zemdlona na wyjściu z każdej etykiety. Zdefiniuj analizę przepływu danych, która wykrywa zmienne silnie żywe. Tzn. podaj dziedzinę zbiorów, funkcje $kill$ oraz gen, ogólną postać równań oraz typ analizy.

Zadanie 5. Przetłumacz następujące instrukcje pętli języka C na kod trójkowy. Postaraj się użyć jak najmniejszej liczby instrukcji skoku. Możesz użyć zmiennych (rejestrów) tymczasowych.

1. $while\ (b)\ \{\ \dots\ \}$
2. $for\ (i\ =\ 0;\ i\ <\ n;\ i\ ++)\ \{\ \dots\ \}$
3. $do\ \{\dots\}\ while\ (b)$

Zmienne i oraz b są całkowite.

Zadanie 6. Przetłumacz następujący program na kod trójkowy.

```
x = a*a*a + 4*a*a*b + 4*a*b*b + b*b*b
```

Występujące w nim zmienne są typu całkowitego i mają po 4 bajty. Zmienne a, b, c są niemodyfikowalne. Użyj jak najmniejszej liczby zmiennych (rejestrów) tymczasowych. Następnie załóż, że w dodatkowym rejestrze o nazwie mem zapamiętany jest adres początku tablicy bajtów, którą możesz wykorzystać do pamiętania tymczasowych wyników obliczeń. Wykonaj ponownie tłumaczenie minimalizując liczbę wykorzystanych rejestrów tymczasowych "przelewając" je (ang. *register spilling*) do pamięci.

Zadanie 7. W kodzie trójkowym zaimplementuj dowolny algorytm sortowania tablicy bajtów t o znanym rozmiarze n.

Zadanie 8. Rozważmy algorytm mnożenia tablicowego liczb 5-bitowych. Realizujący go układ cyfrowy (Appendix J. Fig. J.27) składa się z trzech sumatorów CSA oraz jednego sumatora RCA. Jest to układ kombinacyjny, tzn. wartości na jego wyjściach zależą jedynie od wartości podanych na wejściach; nie posiada on pamięci oraz nie jest synchronizowany sygnałem zegarowym.

1. Zaproponuj układ sekwencyjny, wykorzystujący pamięć oraz pracujący w kilku etapach (cyklach) wyznaczonych przez sygnał zegarowy, który mnoży dwie liczby 5-bitowe, ale używa tylko jednego sumatora CSA (i jednego RCA).
2. Układ z punktu 1. potrzebuje wielu cykli do wyliczenia wyników jednego mnożenia. Zaproponuj układ sekwencyjny, który ma tyle samo sumatorów CSA i RCA co oryginalny układ mnożenia tablicowego, ale w każdym cyklu (z

wyjątkiem kilku początkowych) będzie wyprowadzał na wyjście wyniki mnożeń kolejnych par liczb podawanych na wejściu.

Uwaga: W tym zadaniu należy podać ideę układu i jego schemat wysokiego poziomu (w stylu Fig. J.27a). Nie wymagam schematu układu z dokładnością do bramki.