

Programowanie 2004

Egzamin poprawkowy

6 września 2004

Egzamin trwa trzy godziny zegarowe. Za każde z czterech zadań można otrzymać od -5 do 25 punktów, zatem za cały egzamin można otrzymać do 100 punktów. Za przedstawienie zadania do oceny otrzymuje się -5 punktów. Za brak rozwiązania zadania otrzymuje się 0 punktów. Punkty z egzaminu poprawkowego przeliczają się na oceny następująco: mniej niż 33: ndst, od 33 do 45: dst, od 46 do 58: dst+, od 59 do 71: db, od 72 do 84: db+, od 85: bdb. Każde zadanie proszę pisać na osobnych kartkach. Proszę podpisać własnym imieniem i nazwiskiem każdą kartkę przedstawianą do oceny. Na zakończenie egzaminu proszę oddać do sprawdzenia tylko kartki z rozwiązanymi zadaniami. Brudnopisy i treści zadań proszę zatrzymać.

Zadanie 1. Opisz algorytm, który dla podanych automatów skończonych $\mathcal{A} = \langle \Sigma, Q, q_0, F, \gamma \rangle$ i $\mathcal{B} = \langle \Sigma, R, r_0, G, \delta \rangle$ sprawdza, czy $L(\mathcal{A}) \subseteq L(\mathcal{B})$.

Zadanie 2. Niech F_n oznacza n -ty wyraz ciągu Fibonacciego, tzn. $F_0 = F_1 = 1$ oraz $F_{n+2} = F_{n+1} + F_n$ dla $n \in \mathbb{N}$. Udekoruj poniższy program asercjami tak, by dało się z nich odtworzyć dowód częściowej poprawności programu.

```
{N = n ∧ n ≥ 0}
X=1;
Y=1;
while (N>0) (
    Z=Y;
    Y=X+Y;
    X=Z;
    N=N-1;
)
{X = Fn}
```

Zadanie 3. Język D rozszerzamy o instrukcję `do C while (b)`, gdzie C jest instrukcją zaś b — wyrażeniem logicznym. Przyjmując, że instrukcje interpretujemy w przestrzeni $\mathbb{S}^{\mathbb{S}}$, gdzie \mathbb{S} jest przestrzenią stanów maszyny, zdefiniuj semantykę denotacyjną tej instrukcji. Rozszerz semantykę operacyjną języka D dodając reguły wnioskowania dla tej instrukcji.

Zadanie 4. Udowodnij, że w prostym systemie typów nie istnieje zamknięte lambda wyrażenie M w postaci normalnej, takie, że $\vdash M : (\alpha \rightarrow \beta) \rightarrow \alpha$.

Powodzenia!