

Programowanie
Egzamin zasadniczy
16 czerwca 2003
Deklaracja przystąpienia do egzaminu

Nr zawodnika: 00

W poniższe pole proszę *bardzo starannie* wpisać swoje imię i nazwisko.

Proszę uważnie przeczytać i podpisać poniższą deklarację.

Oświadczam, że przystępuję do egzaminu zasadniczego z przedmiotu „Programowanie”. Mam świadomość, iż podczas egzaminu nie wolno korzystać z żadnych notatek, książek itp., ani pomocy innych osób i nie wolno odpisywać rozwiązań od innych osób, również w sytuacji, gdy będzie to możliwe bez zwrócenia uwagi egzaminatora. Przyjmuję też do wiadomości, że w razie gdybym oszukiwał i zostało to wykryte (zarówno w czasie egzaminu, jaki i sprawdzania pracy), otrzymam ocenę niedostateczną i sprawa zostanie skierowana do Dziekana.

Podpis zdającego

Po podpisaniu proszę odpiąć tę kartkę i przekazać osobie prowadzącej egzamin.
Proszę nie rozpinać i nie odwracać pozostałych kartek przed ogłoszeniem rozpoczęcia egzaminu.

Proszę nie rozpinać i nie odwracać tych kartek przed ogłoszeniem rozpoczęcia egzaminu.

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 rozpoczęcie zadania otrzymuje się -5 punktów (dlatego podane w treści punkty za każde zadanie sumują się do 30). Za brak rozwiązania zadania otrzymuje się 0 punktów. Punkty z egzaminu zasadniczego 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. Ocena z egzaminu zasadniczego jest wystawiana na podstawie sumy punktów z egzaminu i punktów bonusowych. Liczba punktów bonusowych jest częścią całkowitą ilorazu: $(C - 40)/10$, gdzie C oznacza całkowitą liczbę punktów uzyskanych na zaliczenie ćwiczeń. Punkty bonusowe z ćwiczeń dolicza się tylko do wyników egzaminu zasadniczego i tylko w przypadku, gdy ćwiczenia te zaliczono w tym samym semestrze, w którym odbywa się egzamin. Zatem za ćwiczenia zaliczone w poprzednich latach punktów bonusowych nie dopisuje się.

Proszę pisać rozwiązania zadań bezpośrednio na kartkach z ich treścią we wskazanych miejscach. Uwaga: *nie ma* dodatkowego papieru. Odpowiedzi proszę zmieścić w podanych polach. Imienia, nazwiska i numeru indeksu proszę **nie** wpisywać na kartkach z rozwiązaniami. Na zakończenie egzaminu proszę oddać do sprawdzenia tylko kartki z rozwiązanymi zadaniami. Brudnopis proszę zatrzymać.

Powodzenia!

Proszę uzupełnić, wypełniając zaznaczone pola, podane poniżej reguły wnioskowania dla częściowej poprawności programów w języku D (po 2 pkt. za każdą regułę).

(przypisanie)

$$\left\{ \boxed{} \right\} X = e \quad \{\phi\}$$

$$\models \boxed{\phantom{\{\phi\} C \{\psi\}}} \quad \{\phi\} C \{\psi\}$$

(wstępny)

$$\{\phi\} C \{\psi\} \models \boxed{\phantom{\{\phi\} C \{\psi\}}}$$

(końcowy)

$$\left\{ \boxed{\phantom{\{\phi\} C \{\psi\}}} \right\} C \{\phi\}$$

(while)

(if-else)

$$\{\phi\} \text{ if } (b) \ C_1 \text{ else } C_2 \ \{\psi\}$$

ciąg dalszy na odwrocie \longrightarrow

Ciąg dalszy zadania 1.

Pomiędzy wiersze programu proszę wpisać asercje tak, by można z nich było odtworzyć formalny dowód poprawności tego programu (20 pkt). Na prawo od każdej asercji proszę podać nazwę reguły (napisanej na poprzedniej stronie), której użyto, by wywnioskować daną asercję. Uwaga: pomiędzy sąsiednimi wierszami programu jest miejsce dla dwóch asercji, co nie oznacza, że wszędzie należy wpisywać po dwie asercje — w niektórych miejscach wystarczy jedna.

{true}

R = X;

Q = 0;

while (R >= Y) (

 R = R - Y;

 Q = Q + 1;

)

{X = Q * Y + R ∧ R < Y}

Rozważmy język programowania ML++ opisany następującą składnią abstrakcyjną:

typy: $\sigma ::= \text{int} \mid \sigma \rightarrow \tau \mid \sigma \times \tau$
stałe całkowitoliczbowe: n
zmienne: x
operatory arytmetyczne: $\oplus ::= + \mid - \mid < \mid \text{orelse} \mid \text{andalso}$
wyrażenia: $e ::= n \mid x \mid e_1 \oplus e_2 \mid (\lambda x : \sigma. e) \mid (e_1 e_2) \mid (e_1, e_2) \mid$
 $\text{let } x = e_1 \text{ in } e_2 \mid \text{if } e_1 \text{ then } e_2 \text{ else } e_3$

Język ten jest typowany *explicite* i monomorficznie (tak jak Pascal). Operatory porównania i logiczne są, podobnie jak w języku C, operatorami arytmetycznymi (wartość zero oznacza fałsz, wartość różna od zero oznacza prawdę). Tak jak w SML-u język jest wartościowany gorliwie do słabej czołowej postaci normalnej. Wyjątkiem są wyrażenia *orelse*, *andalso* i *if*, wartościowane leniwie. Wyrażenie *let* jest nierekurencyjne. Nie ma ograniczeń na typy parametrów i wyników funkcji — same mogą być dowolnie skomplikowanymi funkcjami.

Napisz poniżej zestaw reguł typowania wyrażeń języka ML++ (8 pkt).

ciąg dalszy na odwrocie →

Ciąg dalszy zadania 2.

Napisz poniżej zestaw reguł semantyki operacyjnej języka ML++ (8 pkt).

Zdefiniuj poniżej semantykę denotacyjną języka ML++ (14 pkt). Zauważ, że dla wyrażeń każdego typu z osobna musisz mieć osobną dziedzinę interpretacji.

Rozważmy algebrę $\mathcal{W} = \langle \{0, 1\}^*, \cdot, \epsilon \rangle$ słów nad alfabetem $\{0, 1\}$ wraz z operacją konkatenacji \cdot i słowem pustym ϵ .

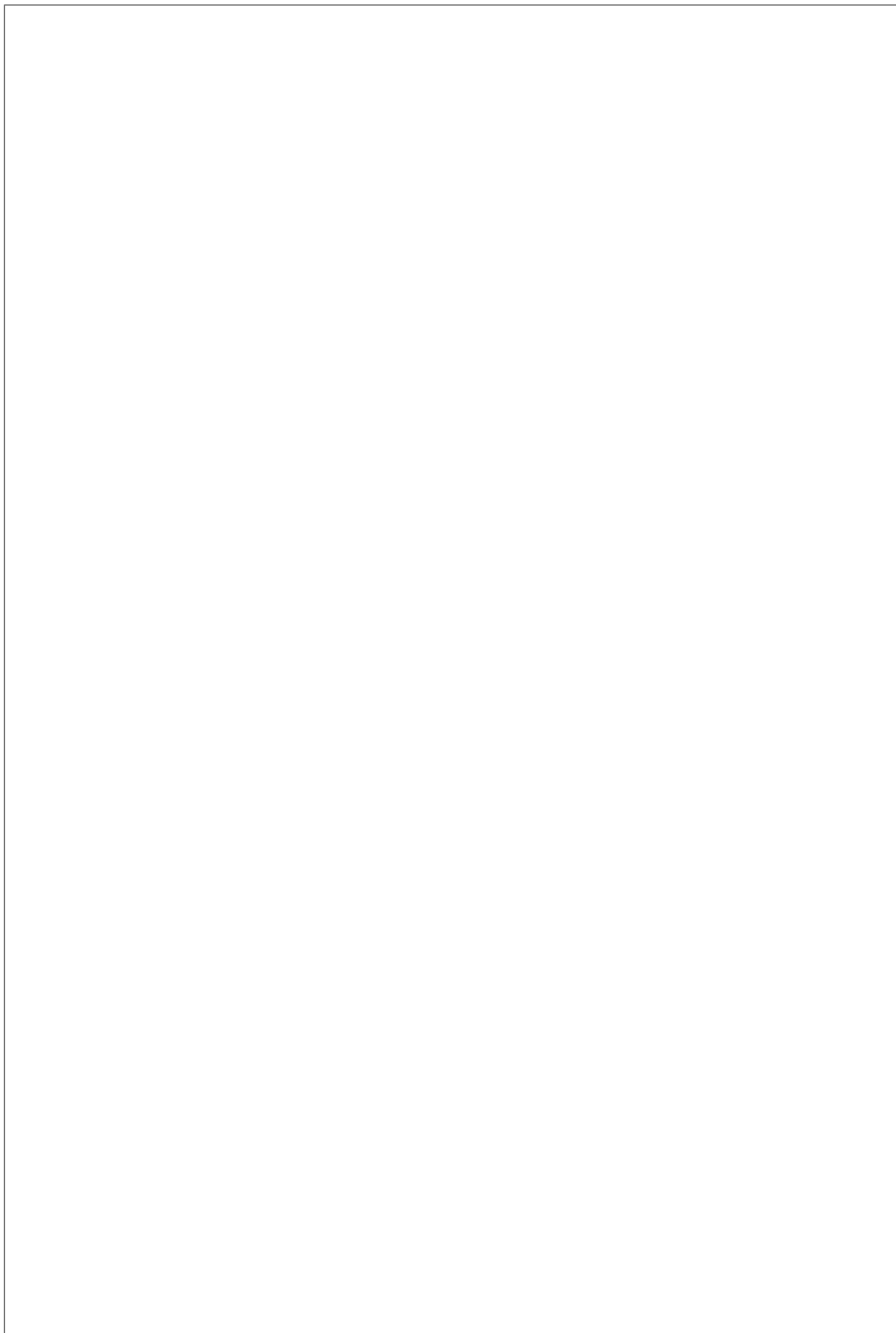
Napisz poniżej, co to oznacza, że język $L \subseteq \{0, 1\}^*$ jest *regularny* (3 pkt).

Napisz poniżej, co to oznacza, że odwzorowanie $h : \{0, 1\}^* \rightarrow \{0, 1\}^*$ jest *homomorfizmem* algebry \mathcal{W} w nią samą (3 pkt).

Napisz poniżej (24 pkt) dowód następującego twierdzenia: Jeżeli $L \subseteq \{0, 1\}^*$ jest językiem regularnym, a $h : \{0, 1\}^* \rightarrow \{0, 1\}^*$ homomorfizmem, to język $h(L)$ też jest regularny.

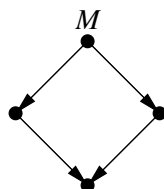
dowód twierdzenia możesz kontynuować na odwrocie \longrightarrow

Ciąg dalszy dowodu twierdzenia

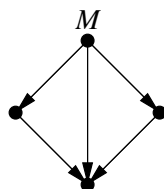


Napisz poniżej (5 pkt), co to jest *graf redukcji lambda wyrażenia*.

Napisz poniżej (5 pkt), przykład zamkniętego lambda wyrażenia M , którego graf redukcji ma następujący kształt:



Udowodnij poniżej (20 pkt), że nie istnieje lambda wyrażenie M , którego graf redukcji miałby następujący kształt:



dowód twierdzenia możesz kontynuować na odwrocie →

Ciąg dalszy dowodu twierdzenia

