

Tentamen Vertalerbouw—23 juni 2011  
Zernikegebouw zaal 13, 09.00-12.00 uur

De nagekeken tentamens zijn in te zien op kamer Bernoulliborg 366.

*Opmerkingen:*

- Schrijf netjes en duidelijk, met zwarte of blauwe pen.
- Zet op het eerste blad naam, studentnummer en het totaal aantal ingeleverde bladen, en nummer de ingeleverde bladen.
- Lees de opgaven eerst goed door.
- Motiveer uw antwoorden.

1. (45 minuten)

a) Geef voor alle nonterminals uit onderstaande produkties de sets *first* en *follow*.

b) Is de grammatica, gegeven door de volgende produkties met startsymbool  $S$ ,  $LL(1)$ ,  $LR(0)$ ,  $SLR(1)$ ,  $LR(1)$ ?

Wanneer U conflicten ontdekt geef deze dan duidelijk aan. Geef, ingeval de conflicten volgens U oplosbaar zijn, aan hoe de oplossing verloopt.

```
S → aABS
, S → AB
, A → aB
, A →
, B → Bb
, B → b
```

2. (50 minuten)

Gegeven is het volgende Pascal-achtige programma:

```
PROGRAM tentamen;

TYPE kwartet = ARRAY [1..4] of int;

VAR p,q: kwartet;
    i,n: int;

PROCEDURE copyplus (a: kwartet; VAR b: kwartet);
    VAR i: int;

    PROCEDURE cp_one;
    BEGIN b[i] := a[i];
    END;
    (* 1 *)
```

3. (45 minuten)

Gegeven is een eenvoudig taaltje, dat syntactisch gespecificeerd wordt door een grammatica (met startsymbool LE) met de volgende productieregels:

```
LE → lpar lambda xsym arrow Body rpar num
, Body → T X
, X →
, X → plus T X
, T → num
, T → xsym
, T → LE
```

Een voorbeeld volgens deze grammatica is:

$(\lambda x \rightarrow x + 5) \ 4$

$(\lambda x \rightarrow x + (\lambda x \rightarrow 2 + x + x) \ 2) \ 3$

a) Laat zien dat deze grammatica  $LL(1)$  is.

b) Schrijf een topdown parser met expliciete stapel voor deze grammatica.

U mag ervan uitgaan dat de volgende declaraties beschikbaar zijn en U kunt ze dus vrijelijk gebruiken:

TYPE

```
symbol = (lpar, rpar, xsym, arrow, num, lambda, plus, eof,
          LE, Body, T, X); (* terminals + nonterminals *)
tsymbol = lpar..eof; (* alleen de terminals *)
```

VAR sym: tsymbol; (\* zie nextsym \*)

PROCEDURE initscanner;

(\* Initialisatie van de scanner \*)

PROCEDURE nextsym;

(\* Levert bij aanroep de tokenwaarde op (in de variabele sym) van het eerstvolgende symbool in de invoer \*)

PROCEDURE error;

(\* Genereert een primitieve foutmelding en stopt de parser \*)

Verder mag U uitgaan van de beschikbaarheid van de ADT stapel (met elementen van type symbol); een variabele van dit type is impliciet gedeclareerd (maar niet geïnitieerd). Beschikbare operaties op deze stack variabele:

```
mkEmpty, isEmpty, push, pop, top
```