

Tentamen Vertalerbouw— 1 juli 2008

- Schrijf netjes en duidelijk, met zwarte of blauwe pen.
- Zet op het eerste blad alle gegevens als naam, etc., en het totaal aantal ingeleverde bladen, en nummer de ingeleverde bladen.
- Lees de opgaven eerst goed door.
- Motiveer uw antwoorden.
- De opgaven tellen niet even zwaar.

1. (40 punten)

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

b) Is de grammatica (gegeven door de volgende produkties) *LL(1)*, *LR(0)*, *SLR(1)* en/of *LR(1)*? Geef in geval van conflicten deze duidelijk aan.

$$\begin{aligned} S &\rightarrow AB \mid AC \\ , A &\rightarrow Aa \mid C \mid \varepsilon \\ , B &\rightarrow Bb \mid CS \mid \varepsilon \\ , C &\rightarrow Cc \mid c \end{aligned}$$

2. (50 punten)

Gegeven is het volgende (Pascal-achtige) programma:

```
PROGRAM tentamen;

TYPE rij = ARRAY [5..8] OF integer;

VAR a: rij;
    m,r: integer;

FUNCTION f (r: integer): integer;
BEGIN ...
    f := a[r]+a[m] (* 1 *)
END;

PROCEDURE p (m: integer; VAR r: integer);
VAR j: integer;
PROCEDURE q (VAR s: integer);
BEGIN r := s*f(m) (* 2 *)
END;
BEGIN r := a[j]+m; (* 3 *)
    ...
    q(r) (* 4 *)
END;
```

```

BEGIN ...
    p(6,r);                                (* 5 *)
    ...
END (* tentamen *).

```

Voor het geheugenbeheer en de adresberekeningen worden de volgende registers gebruikt:

GP het base address van het activation record van het hoofdprogramma,  
LNB het base address van het huidige activation record, en  
LFA het adres van de eerste vrije geheugenlokatie.

Voor het overdragen van de omgeving van een aan te roepen procedure kan het register ENV worden gebruikt.

In de machineinstructies CALL 1able en RETURN van de doelmachine wordt impliciet gebruik gemaakt van een (aparte) return stack. U hoeft zich dus niet druk te maken over terugkeer-adressen!

Er zijn voldoende registers (R0, R1, R2, ...) voor het opslaan van de tussenresultaten.

- a) Geef de layout van de activation records van  $f$  en  $q$ .
- b) Geef de te genereren (pseudo-)instructies voor de procedure-entry en exit van  $q$ .
- c) Geef de te genereren (pseudo-)instructies voor de 5 gemarkeerde statements. Controle op index-waarden, die al dan niet buiten array grenzen zouden kunnen gaan, mag je weglaten!

3. (50 punten)

Zij  $G$  een uitgebreide contextvrije grammatica met de set  $P$  van productieregels:

$$P = \{ \begin{array}{l} Z \rightarrow S \text{ eofs} \\ , S \rightarrow Si \\ , S \rightarrow ifs E \text{ thens } Si \text{ Ep } fis \\ , E \rightarrow T Xt \\ , T \rightarrow ident \\ , T \rightarrow lpar E rpar \\ , Xt \rightarrow op T Xt \\ , Xt \rightarrow \\ , Ep \rightarrow elses S \\ , Ep \rightarrow \\ , Si \rightarrow assignment \\ , Si \rightarrow begs Sl \text{ ends} \\ , Sl \rightarrow S St \\ , St \rightarrow sc S St \\ , St \rightarrow \end{array} \}$$

(De terminal *sc* staat voor de semicolon ';'.)

- Geef voor elke nonterminal in *G* een default productie aan.
- Geef de functiewaarden van de functie *follow* voor de nonterminals *Xt* en *St*.
- Toon aan dat de richters van de producties voor de nonterminal *Xt* disjunct zijn. Evenzo voor de nonterminal *St*.
- Geef de implementatie van het hoofdprogramma en van de procedures voor de nonterminals *Si*, *Sl* en *St* in een recursive descent parser voor *G*, inclusief syntactische error recovery.

Onderstaande declaraties mogen worden gebruikt in de implementatie van de recursive descent parser. Alle overige gebruikte procedures e.d. moeten gedeclareerd worden.

```
TYPE
  tsymbol      = (assignment, begs, elses, ends, fis, ident,
                 ifs, lpar, op, rpar, sc, thens, eofs);
  tsymbolset = SET OF tsymbol;

VAR
  sym: tsymbol;

PROCEDURE initscanner;
  (* Initialisatie van de scanner *)
  BEGIN ... END (* initscanner *);

PROCEDURE nextsym;
  (* Levert bij aanroep de tokenwaarde op (in de variabele
     sym) van het eerstvolgende symbool in de invoer *)
  BEGIN ... END (* nextsym *);

PROCEDURE delete (keys: tsymbolset);
  (* gooit alle symbols weg die niet in keys zitten *)

PROCEDURE insertion (sy: tsymbol);
  (* verzorgt foutmelding dat symbol sy is ingevoegd *)
```