

## Tentamen Vertalerbouw—8 april 2009

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 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.

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  $S$ ,  $LL(1)$ ,  $LR(0)$ ,  $SLR(1)$ ,  $LR(1)$ ?

Geef in geval van conflicten deze duidelijk aan. Geef, ingeval de conflicten volgens U oplosbaar zijn, aan hoe de oplossing verloopt.

```
S → AbC
, A → Ab
, A → c
, A → S
, C → bC
, C → c
```

2. (50 minuten)

Gegeven is het volgende Pascal(-achtige) programma:

```
PROGRAM tentamen;

TYPE paar = array [1..2] of integer;

VAR p: paar;
    i: integer;
```

```

PROCEDURE copy (VAR a: paar; b: paar);
VAR i: integer;

    PROCEDURE one (i: integer);
    BEGIN IF (i<2) THEN
            a[i+1] := b[i];           (* 1 *)
        ELSE a[1] := b[i];
    END;

BEGIN FOR i := 1 TO 2 DO BEGIN
        one(i);                       (* 2 *)
        a[i] := a[i] + p[i];         (* 3 *)
    END
    ...
END (* copy *)

BEGIN ...
    copy(p,p)                         (* 4 *)
    ...
END (* tentamen *).

```

Voor het geheugenbeheer 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

**lfa** het adres van de eerste vrije stack locatie

(We gaan ervan uit dat het return adres op een aparte stack wordt bewaard, zodat U daarmee geen rekening hoeft te houden.)

Voor het overdragen van de omgeving van een aan te roepen procedure kan het register **env** worden gebruikt. Verder zijn er voldoende registers (R0, R1, . . .) voor het opslaan van tussenresultaten. Merk op dat het (Pascal) keyword **VAR** betekent dat het argument *by reference* wordt doorgegeven.

- Teken de AR (activation record) voor procedure copy;
- Geef de te genereren (pseudo-)instructies voor de **entry** en **exit** van **one**;
- Geef de te genereren (pseudo-)instructies voor de vier genummerde regels. U hoeft geen arraybound checks te doen!

3. (55 minuten)

Gegeven is een eenvoudig taaltje, dat syntactisch gespecificeerd wordt door de productieregels  $P$ :

$$P = \left\{ \begin{array}{ll} S & \rightarrow E \text{ Stl} \\ , & \text{Stl} \rightarrow \text{semicolon } S \\ , & \text{Stl} \rightarrow \\ , & E \rightarrow T \text{ Etl} \\ , & \text{Etl} \rightarrow \text{addopsym } E \\ , & \text{Etl} \rightarrow \\ , & T \rightarrow F \text{ Ttl} \\ , & \text{Ttl} \rightarrow \text{mulopsym } T \\ , & \text{Ttl} \rightarrow \\ , & F \rightarrow \text{intdenotation} \\ , & F \rightarrow (S) \\ & \end{array} \right\}$$

- Laat zien dat deze grammatica  $LL(1)$  is.
- Voor een recursive descent parser met error-recovery hebben we o.a. de volgende procedures nodig: `pStl`, `pT`, `pTtl`, `match` en `delete`.

Geef de implementatie van deze procedures.

