

## Hertentamen Vertalerbouw—14 juli 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.

$$\begin{aligned} S &\rightarrow AAB \\ , S &\rightarrow AB \\ , A &\rightarrow aA \\ , A &\rightarrow b \\ , B &\rightarrow bB \\ , B &\rightarrow \end{aligned}$$

2. (55 minuten)

Gegeven is het volgende Pascal(-achtige) programma:

```
PROGRAM tentamen;  
  
CONST N = 9  
TYPE vector = array [0..N] of integer;  
  
VAR v,w: vector;  
    n: integer;
```

```

PROCEDURE copy_vec (a: vector; VAR b: vector);
VAR n: integer;

    PROCEDURE copy_elem (i: integer);
    BEGIN b[i] := a[i];          (* 1 *)
    END;

BEGIN FOR n := 0 TO N DO BEGIN
    copy_elem(i);                (* 2 *)
    END
    ...
END (* copy_vec *)

PROCEDURE copy_one (i: integer);
BEGIN w[i] := v[i];
END; (* copy_one *)

PROCEDURE do_copy (PROCEDURE cp (n: integer);
BEGIN FOR n := 0 to N DO
    cp(n)                        (* 3 *)
END' (* do_copy *)

BEGIN ...
    do_copy (copy_one);          (* 4 *)
    copy(v,w)                    (* 5 *)
    ...
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_vec`;
- Geef de te genereren (pseudo-)instructies voor de **entry** en **exit** van `copy_elem`;
- Geef de te genereren (pseudo-)instructies voor de vijf genummerde regels. U hoeft geen arraybound checks te doen!

3. (50 minuten)

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

$$P = \left\{ \begin{array}{l} S \rightarrow E Stl \\ , \quad Stl \rightarrow \text{semicolonsym } S \\ , \quad Stl \rightarrow \\ , \quad E \rightarrow T Etl \\ , \quad Etl \rightarrow \text{plussym } E \\ , \quad Etl \rightarrow \\ , \quad T \rightarrow F Ttl \\ , \quad Ttl \rightarrow \text{maalsym } T \\ , \quad Ttl \rightarrow \\ , \quad F \rightarrow \text{number} \\ , \quad F \rightarrow (S) \\ \end{array} \right\}$$

Een voorbeeldinvoer:  $3+4 ; (7*2+1 ; 15)$

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

Geef de implementatie van deze procedures.

