TENTAMEN COMPILERCONSTRUCTIE
Vrijdag 21 december 2012, 10:00 - 13:00 uur

Dit tentamen bestaat uit 5 opgaven.

Als je het antwoord op een onderdeel niet weet, en je hebt dat antwoord nodig bij een later onderdeel, dan kun je het antwoord 'kopen' bij de docent.

Als er bij een opgave gevraagd wordt om uitleg of motivatie van je antwoord, is het belangrijk dat je die ook geeft.

---

1. (a) Noem drie soorten informatie over een variabele die in de *symbol table* worden bijgehouden. In welke fase van het compileerproces wordt elk van deze soorten informatie in de symbol table gezet? Motiveer je antwoorden.

   (b) Waarom houdt een compiler niet één algemene symbol table bij, maar een complete hierarchie van symbol tables?

---

2. Beschouw de context-vrije grammatica G met startvariabele S en de volgende producties:

$$\begin{aligned} S &\rightarrow \textbf{if } B\ S \mid \textbf{id} = \textbf{id} \\ B &\rightarrow B\ \textbf{boolop}\ E \mid E \\ E &\rightarrow \textbf{id relop id} \mid \textbf{id} \end{aligned}$$

   G is dus een eenvoudige grammatica voor een instructie in een programmeertaal. S, B, E zijn de variabelen en $\textbf{if}, \textbf{id}, =, \textbf{boolop}, \textbf{relop}$ zijn de terminalen in G.

   (a) Bepaal voor elke variabele in G zowel de FIRST- als de FOLLOW-verzameling.

   (b) Construeer de LR(0)-automaat bij grammatica G.

   (c) Construeer de SLR *parsing table* bij grammatica G.

   (d) Is G een SLR grammatica? Motiveer je antwoord.

Beschouw nu de context-vrije grammatica H met startvariabele S en de volgende producties:

(1) $S \to S - A$
(2) $S \to A$
(3) $A \to AB$
(4) $A \to B$
(5) $B \to B+$
(6) $B \to p$
(7) $B \to q$

(e) Geef (ad hoc) een afleidingsboom in H voor de string $q + p$.

(f) Gegeven is dat de SLR parsing table bij grammatica H er als volgt uit ziet:

State	Action					Goto		
	$-$	$+$	p	q	\$	S	A	B
0			s6	s7		1	9	8
1	s2				acc			
2			s6	s7			3	8
3	r1		s6	s7	r1			4
4	r3	s5	r3	r3	r3			
5	r5	r5	r5	r5	r5			
6	r6	r6	r6	r6	r6			
7	r7	r7	r7	r7	r7			
8	r4	s5	r4	r4	r4			
9	r2		s6	s7	r2			4

Parse de string $q+p$ met deze tabel. Laat bij iedere stap duidelijk zien wat je doet, bijvoorbeeld met behulp van een tabel van de volgende vorm:

States on stack	Corresponding Symbols on stack	Input	Action
...	...	...	...

---

3. Verschillende typen in een programmeertaal kunnen naar elkaar geconverteerd worden.

   (a) Wanneer spreken we van een *widening conversion* en wanneer van een *narrowing conversion* ? Geef van elk van deze twee typen conversie een voorbeeld.

   (b) Wanneer spreken we van een *coercion* ?

---

4. Tijdens het genereren van drie-adres code voor boolese expressies en *flow-of-control* instructies weten we bij Goto-instructies vaak niet onmiddellijk waar we naartoe moeten springen. We kunnen *backpatching* gebruiken om dit achteraf op te lossen. Hierbij krijgt de variabele B in de grammatica (overeenkomend met een boolese expressie) attributen *truelist* en *falselist*. De variabele S (overeenkomend met een instructie) krijgt een attribuut *nextlist*.

   (a) Leg voor elk van deze drie lijsten uit wat de lijst bevat.

   (b) Een deel van het *translation scheme* voor het werken met *truelist* en *falselist* ziet er als volgt uit:

   $$\begin{array}{ll} B \rightarrow B_1 || M B_2 & \{ \; backpatch(B_1.falselist, M.instr); \\ & \quad B.truelist = merge(B_1.truelist, B_2.truelist); \\ & \quad B.falselist = B_2.falselist; \} \\ M \rightarrow \epsilon & \{ \; M.instr = nextinstr; \} \end{array}$$

   Hierbij is M een hulpvariabele.

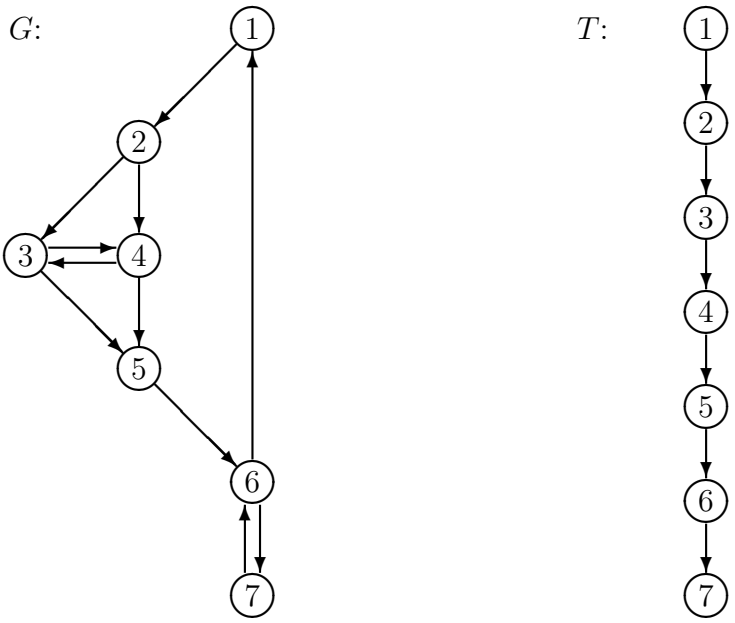
   Leg uit wat er gebeurt (de *semantic action*) bij de productie $B \rightarrow B_1 || M B_2$. Leg ook uit waarom dat gebeurt.

   (c) Geef de semantic action voor de productie $B \rightarrow B_1 \&\& M B_2$. Motiveer je antwoord.

---

5. (a) Wanneer noemen we een knoop d in een *flow graph* een *dominator* van een knoop n ?

   Beschouw nu de volgende *flow graph* G met beginknoop 1 (het plaatje links):



   (b) Geef de *dominator tree* voor flow graph G.

   De boom T hierboven (het plaatje rechts) is een voorbeeld van een *depth-first spanning tree* voor G.

   (c) Wat zijn de *retreating edges* van flow graph G bij boom T ? En wat zijn de *back edges* ? Motiveer je antwoorden.

   (d) Geef bij elke back edge van G de bijbehorende *natural loop*. Leg uit hoe je aan je antwoord(en) komt.
