Prüfung aus Übersetzerbau 30.11.2000 Musterlösung

1. 25 % Quadrupel-Code

Erzeugen Sie für das rechte Programmstück Quadrupel-Code nach der Kontrollfluss-Methode. Ein `INTEGER` ist 4 Byte groß. Die Untergrenze aller Indexbereiche ist 0.

```
VAR
  a:  ARRAY[10,20] OF INTEGER;
  i,j:  INTEGER;
⋮
WHILE ((i+5)<=9) AND NOT ((j-7)<0) THEN
  i := a[i,j];
  j := a[i+5,a[i,j-2]-7];
END
```

2. 25 % Heap-Verwaltung

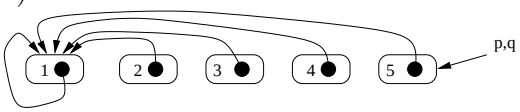
Gegeben sei folgendes Programm, das fünf Zellen alloziert.

```
TYPE
  dyn:  RECORD
    link:  POINTER TO dyn;
  END;
VAR
  i:  INTEGER;
  p,q:  POINTER TO dyn;
⋮
p:= NEW dyn;
p.link:= p;
<------------- t1 ------------>
FOR i:= 1 TO 4 DO
  q := NEW dyn;
  q.link := p.link;
  p := q;
END
<------------- t2 ------------>
```

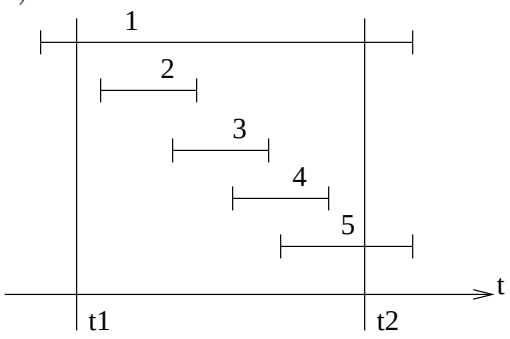
a) (15 %) Stellen Sie die erzeugte dynamische Datenstruktur zum Zeitpunkt `t2` graphisch dar und numerieren Sie jede Zelle in der Reihenfolge ihrer Erzeugung. Die Graphik soll jede allozierte Zelle und die enthaltenen Referenzen darstellen. Zusätzlich sollen die Variablen `p` und `q` mit ihren Referenzen eingezeichnet werden.

b) (10 %) Zeichnen Sie die Klassifizierung der Zellen bezüglich des Zeitintervals `(t1,t2)`.

a)



b)



3. 25 % Grammatik

Gegeben sei folgende Grammatik (Kleinbuchstaben und Sonderzeichen sind Terminalsymbole; Startsymbol ist S):

S → L $
L_1 → L a L
L_2 → b L
L_3 → c L d
L_4 → e

a) (5 %) Bestimmen Sie die first- und follow-Menge des Nonterminals L.

$first(L) = \{b, c, e\}$
$follow(L) = \{a, d, \$\}$

b) (5 %) Begründen Sie, warum die Grammatik nicht LL(1) ist.

Die Grammatik ist nicht LL(1), da in der Produktion L_1 eine Linksrekursion auftritt.

c) (15 %) Geben Sie eine neue Grammatik an, die die selbe Sprache erzeugt, und eine LL(1) Grammatik ist.

S → L $
L → F L_r
L → b F L_r
L_r → ε
L_r → a L
F → c L d
F → e

4. 25 % Attributierte Grammatik

Diese Grammatik berechnet die Online-Zeit **Oz** für eine Liste von Computern **Cl** die einen Provider **P** verwenden. Die Online-Zeiten eines Computers **C** (PC **pc**, Powerbook **pb** oder Palm III **p3**) stehen in einer Verbindungsliste **Vl**. Je nach Tageszeit oder Wochentag ergibt sich die Art **A** (Tag, Nacht oder Feiertag) einer Verbindung **v**.
An einem Feiertag werden die gleichen Gebühren verrechnet wie in der Nacht. Der PC nutzt die Kanalbündelung von ISDN und hat daher *doppelt so hohe* Verbindungskosten.

```
Oz → Cl P
Cl → C Vl | Cl C Vl
C → pc | pb | p3
Vl → v A | Vl v A
A → tag | nacht | feiertag
P → tiger | atu | itwo
```

	Gebühren/min.	
Provider	Tag	Nacht
tiger	0,50	0,20
atu	0,40	0,10
itwo	0,80	0,30

Erweitern Sie die Grammatik um Attribute zur Berechnung der Gesamtkosten. Die Verbindungsgebühren stehen in den synthetisierten Attributen `tag` bzw. `nacht` der jeweiligen Provider. Die Verbindungsdauer (in Minuten) steht im syn. Attribut `v.min`. Die Summe der angefallenen Kosten soll im syn. Attribut `Oz.s` geliefert werden. Anmerkung: Eindeutige Kopierregeln von Attributen können weg gelassen werden.