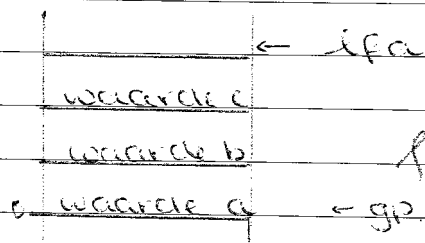
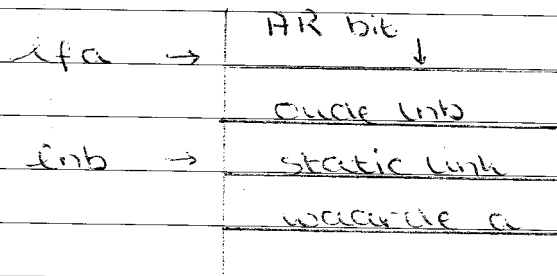
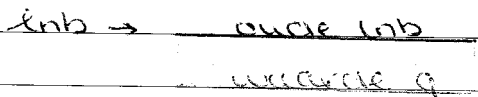
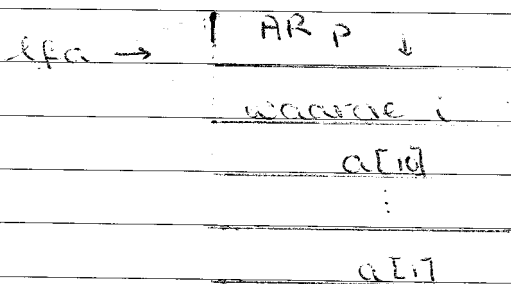
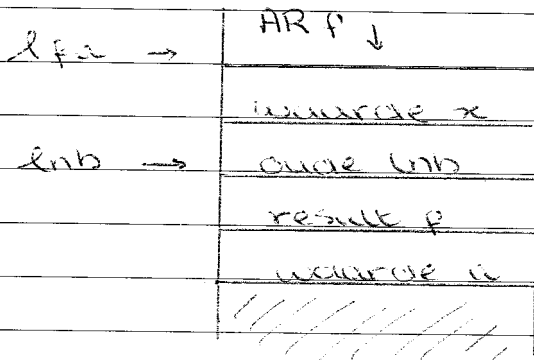


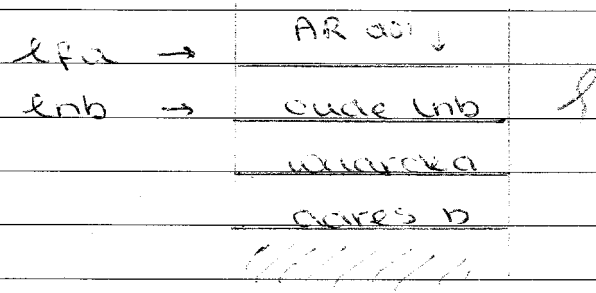
3a



b bit_lab : $M(\&fa) := \&nb$
 $M(\&fa+1) := \&nb$
 $\&fa := \&fa + 2$
 $\&nb := \&fa - 2$

$\text{Lnb} := M(\text{Lnb} + 1)$ }
 return

c $b := -a * c :$



$R_0 := -M(\text{Lnb} - 1)$ // waarde -a
 $R_1 := M(\text{gp} + 2)$ // waarde c
 $R_2 := M(\text{Lnb} - 2)$ // adres b
 $M(R_2) := R_0 * R_1$ }

$b := (a + b) :$

$R_0 := M(\text{Lnb} - 1)$ // waarde a
 $R_1 := M(\text{gp} + 1)$ // waarde b
 $M(\text{gp} + 1) := R_0 + R_1$ }

doi ($2 * a, x$) :

$R_0 := \text{Lnb} + 1$ // adres x
 $M(\text{Lpa}) := R_0$
 $\text{Lpa} := \text{Lpa} + 1$
 $R_1 := M(\text{Lnb} - 2)$ // waarde a
 $M(\text{Lpa}) := 2 * R_1$
 $\text{Lpa} := \text{Lpa} + 1$
 call doi
 $\text{Lpa} := \text{Lpa} - 2$ }

bit(a) :

$R_0 := M(\text{Lnb} - 2)$ // waarde a
 $M(\text{Lpa}) := R_0$
 $\text{Lpa} := \text{Lpa} + 1$ ← env := ... bit(0) = 2!
 call bit

```

4. b procedure pTLL (keys: tsymbolset)
begin
  delete (keys + first(S) + semicolon);
  if sym = semicolon then
    nextsym;
  pS(keys)
end (* dit skip *)
end;

```

```

procedure pELL (keys: tsymbolset)
begin
  delete (keys + first(E) + plus + minus);
  if sym ∈ {plus, minus} then
    nextsym;
  pE(keys);
end
end;

```

↑ hier zou ik splitsen (om momentik) + en = open

```

procedure pE (keys: tsymbolset)
begin
  delete pF (keys + first(ELL));
  pELL(keys)
end;

```

```

procedure delete (keys: tsymbolset)
begin
  while sym ≠ keys do
    begin
      next printf ("error, %s deleted", sym);
      nextsym
    end
  end
end;

```

~~pE~~

vervolg 4b

~~if function
processes first (s: symbol)
begin
if s ∈ symbolset then
return s;
else~~

$\text{first}(E1) = \{\text{plus, minus, } \epsilon\}$

$\text{first}(E) = \text{first}(F) = \{a, (\}$

$\text{first}(D) = \text{first}(E) = \{a, (\}$

2	symbol	attribuut	inh/syn
	R	val	syn
	S	val	syn
	S	radix	syn
	# dig	val	syn
	E	val	syn
	E	radix	inh

T, F en N hebben dezelfde synthesized en inherited attributen als E

de symbolen #, +, *, (en) hebben geen attributen

V

En nu de rekenvoorschriften...

Vervolg 3c

RETURN x :

$$R_0 := M(\text{lenb} + 1)$$

// waarde x

R[X] :=

$$M(\text{lenb} - 1) := R_0$$

en vermen met exit_code f

$$\text{a[i]} := f(i) * q :$$

$$R_0 := M(\text{lena} - 1)$$

// waarde i

$$M(\text{lena}) := R_0$$

$$\text{lena} := \text{lena} + 2$$

call - f

$$R_1 := M(\text{lena} - 1)$$

// waarde f(i)

$$\text{lena} := \text{lena} - 2$$

$$R_2 := M(\text{lenb} - 1)$$

// waarde q

$$M(\text{lenb} + R_0) := R_1 * R_2$$

X f gebouwd R0 mk!

$$1a \text{ first}(C) = \{c\}$$

$$\text{first}(A) = \{a\}$$

$$\text{first}(B) = \{b\}$$

$$\text{first}(S) = \text{first}(A) \cup \text{first}(B) \cup \text{first}(C) = \{a, b, c\}$$

$$\text{first}(D) = \{d\} \cup \text{first}(S) = \{a, b, c, d\}$$

in weg de productieregel $Z \rightarrow S \perp$ toe in de grammatica (zonder deze regel zijn de follow-sets in dit geval gelijk aan onderstaande, met daartuit \perp weggelaten)

$$\text{follow}(S) = \{\perp\} \cup \text{follow}(D)$$

$$\text{follow}(D) = \text{follow}(S) \cup \text{follow}(C)$$

$$\text{follow}(C) = \text{first}(D) = \{a, b, c, d\}$$

$$\Rightarrow \text{follow}(D) = \{\perp\} \cup \text{follow}(D) \cup \{a, b, c, d\} = \{a, b, c, d, \perp\}$$

$$\Rightarrow \text{follow}(S) = \{\perp\} \cup \{a, b, c, d, \perp\} = \{a, b, c, d, \perp\}$$

$$\text{follow}(A) = \text{first}(B) \cup \text{first}(C) = \{b, c\}$$

$$\text{follow}(B) = \text{first}(C) \cup \text{follow}(A) \cup \{b\} = \{b, c\}$$

$$b \quad B \rightarrow Bb \quad \leftarrow \text{link-recursie, dus niet LL(1)}$$

$S \rightarrow \cdot ABCD$

$A \rightarrow \cdot aAB$

$A \rightarrow \cdot$

} shift/reduce-conflict,

dus niet LR(0) tussen w en w?

X

$follow(A) = \{b, c\}$

~~$first(S) = \{a, b, c\}$~~

$follow(A) \cap first(S) \neq \emptyset = \{b, c\} \neq \emptyset$

\Rightarrow wanneer je in deze situatie (toestand) een b of c leest, weet je ~~niet~~ (met follow-informatie) niet of je shift of reduce moet doen

\Rightarrow niet SLR(1)

ⓐ $Z \rightarrow \cdot SL, \emptyset$

$S \rightarrow \cdot ABCD, \{L\}$

$A \rightarrow \cdot aAB, first(BCD L) = \{b\} \cup first(CD L) = \{b, c\}$

$A \rightarrow \cdot, \{b, c\}$

ⓑ

weer een s/r-conflict

lees je een b of c, dan weet je met de

extra voor informatie nog niet wat te doen

\Rightarrow niet LR(1) a shift, or reduce ?!

X

ⓐ

✓

~~LA~~ $LA(S \rightarrow \text{semicolon } S) = \{\text{semicolon}\}$

$LA(S \rightarrow \cdot) = follow(S) = follow(S) = \{ \}$

$LA(S \rightarrow \text{semicolon } S) \cap LA(S \rightarrow \cdot) = \emptyset$

$LA(E \rightarrow \text{plus } E) = \{\text{plus}\}$

$LA(E \rightarrow \text{minus } E) = \{\text{minus}\}$

$LA(E \rightarrow \cdot) = follow(E) = follow(E)$

$= first(S) \setminus \{E\} \cup follow(S)$

$= \{\text{semicolon}\} \cup follow(S)$

$= \{\text{semicolon}, \cdot\}, L$

✓

pragmatische
doorsnedes

van deze drie verzamelingen zijn leeg

$LA(F \rightarrow a) = \{a\}$ } lege doorsnede

$LA(F \rightarrow (S)) = \{($ }

dus de grammatica is LL(1)