Corso di Linguaggi di Programmazione — Parziale di fine modulo
Prova scritta **B** del 15 Dicembre 2016.

Tempo a disposizione: 2 ore e 30 minuti.

1. Per quali valori delle variabili $X, Y$ e $Z$ la seguente espressione
$$\mathcal{I}^{L_0}_{L_1}(\mathcal{C}^{L_1}_{X,L_3}, \mathcal{C}^{Y}_{Z,L_2})$$
ha senso? E cosa viene calcolato?

2. Si consideri il seguente linguaggio di programmazione, denominato *Silly*, definito dalla seguente sintassi astratta:
$$c \quad ::= \quad \mathtt{y} := 2 \mid c\,;c \mid \mathtt{while\ tt\ do}\, c$$
dove y è l'unica variabile utilizzabile. Definire le regole di semantica operazionale strutturata per *Silly*. Quali funzioni da input ad ouput (da store in store) può calcolare *Silly*? Questo linguaggio è Turing-completo?

3. Considerando la sintassi astratta di *Silly* al punto precedente, si verifichi che essa è ambigua. Si proponga una sintassi concreta, che può far uso di zucchero sintattico, che sia non ambigua.

4. Costruire una grammatica libera $G$ che generi il linguaggio $L = \{b^{n+1}c^{m+1}a^n \mid n, m \geq 0\}$ ed argomentare che effettivamente $G$ generi $L$.

5. Classificare il linguaggio $L$ del punto precedente, ovvero dire se $L$ è regolare, oppure libero ma non regolare, oppure non libero, giustificando adeguatamente la risposta.

6. Si consideri l'espressione regolare $(a|ab)^*a$. Si costruisca l'automa NFA $M$ associato, secondo la costruzione vista a lezione. Si trasformi l'NFA $M$ nell'equivalente DFA $M'$, secondo la costruzione per sottoinsiemi vista a lezione.

7. Preso il DFA $M'$ calcolato al punto precedente, si verifichi se è minimo; se non lo fosse, lo si minimizzi per ottenere un DFA $M''$; quindi si ricavi da $M''$ la grammatica regolare associata, seguendo la costruzione vista a lezione; quindi si semplifichi la grammatica ottenuta, eliminando i simboli inutili; infine, si ricavi da quella grammatica l'espressione regolare associata.

8. Il linguaggio $L = \{a^n b^m \mid n \geq m \geq 0\}$ può essere generato da una grammatica di classe LR(0)? Giustificare la risposta senza esibire alcuna grammatica.

9. Sapendo che $L_1 = \{a^n b^m \mid 0 \leq m \leq n\}$ ed $L_2 = \{a^n b^m \mid 0 \leq n \leq m\}$ sono liberi deterministici, è vero che $L_1 \cap L_2$ è un linguaggio libero deterministico?

10. Si consideri la grammatica $G$ con simbolo iniziale $V$:
$$\begin{array}{lcl} V & \to & A \mid V \wedge A \\ A & \to & (V) \mid b \end{array}$$
(*i*) Verificare che $G$ non è di classe LL(1). (*ii*) Manipolare la grammatica $G$ per renderla di classe LL(1). (*iii*) Costruire la tabella di parsing LL(1). (*iv*) Mostrare il funzionamento del parser LL(1) su input $(b) \wedge b$.

11. Si consideri la grammatica $G$ con simbolo iniziale $V$ del punto precedente. (*i*) Verificare se $G$ sia di classe LR(0), costruendo la tabella di parsing LR(0). (*ii*) Mostrare il funzionamento del parser LR(0) su input $(b \wedge b)$.

Corso di Linguaggi di Programmazione — Parziale di fine modulo
Prova scritta **B** del 15 Dicembre 2016.

Tempo a disposizione: 2 ore e 30 minuti.

1. Per quali valori delle variabili $X, Y$ e $Z$ la seguente espressione
$$\mathcal{I}^{L_0}_{L_1}(\mathcal{C}^{L_1}_{X,L_3}, \mathcal{C}^{Y}_{Z,L_2})$$
ha senso? E cosa viene calcolato?

2. Si consideri il seguente linguaggio di programmazione, denominato *Silly*, definito dalla seguente sintassi astratta:
$$c \quad ::= \quad \mathtt{y}:=2 \mid c;c \mid \mathtt{while\ tt\ do}\, c$$
dove y è l'unica variabile utilizzabile. Definire le regole di semantica operazionale strutturata per *Silly*. Quali funzioni da input ad ouput (da store in store) può calcolare *Silly*? Questo linguaggio è Turing-completo?

3. Considerando la sintassi astratta di *Silly* al punto precedente, si verifichi che essa è ambigua. Si proponga una sintassi concreta, che può far uso di zucchero sintattico, che sia non ambigua.

4. Costruire una grammatica libera $G$ che generi il linguaggio $L = \{b^{n+1}c^{m+1}a^n \mid n, m \geq 0\}$ ed argomentare che effettivamente $G$ generi $L$.

5. Classificare il linguaggio $L$ del punto precedente, ovvero dire se $L$ è regolare, oppure libero ma non regolare, oppure non libero, giustificando adeguatamente la risposta.

6. Si consideri l'espressione regolare $(a|ab)^*a$. Si costruisca l'automa NFA $M$ associato, secondo la costruzione vista a lezione. Si trasformi l'NFA $M$ nell'equivalente DFA $M'$, secondo la costruzione per sottoinsiemi vista a lezione.

7. Preso il DFA $M'$ calcolato al punto precedente, si verifichi se è minimo; se non lo fosse, lo si minimizzi per ottenere un DFA $M''$; quindi si ricavi da $M''$ la grammatica regolare associata, seguendo la costruzione vista a lezione; quindi si semplifichi la grammatica ottenuta, eliminando i simboli inutili; infine, si ricavi da quella grammatica l'espressione regolare associata.

8. Il linguaggio $L = \{a^n b^m \mid n \geq m \geq 0\}$ può essere generato da una grammatica di classe LR(0)? Giustificare la risposta senza esibire alcuna grammatica.

9. Sapendo che $L_1 = \{a^n b^m \mid 0 \leq m \leq n\}$ ed $L_2 = \{a^n b^m \mid 0 \leq n \leq m\}$ sono liberi deterministici, è vero che $L_1 \cap L_2$ è un linguaggio libero deterministico?

10. Si consideri la grammatica $G$ con simbolo iniziale $V$:
$$\begin{array}{lcl} V & \to & A \mid V \wedge A \\ A & \to & (V) \mid b \end{array}$$
(*i*) Verificare che $G$ non è di classe LL(1). (*ii*) Manipolare la grammatica $G$ per renderla di classe LL(1). (*iii*) Costruire la tabella di parsing LL(1). (*iv*) Mostrare il funzionamento del parser LL(1) su input $(b) \wedge b$.

11. Si consideri la grammatica $G$ con simbolo iniziale $V$ del punto precedente. (*i*) Verificare se $G$ sia di classe LR(0), costruendo la tabella di parsing LR(0). (*ii*) Mostrare il funzionamento del parser LR(0) su input $(b \wedge b)$.

1) $I^{L0}_{L1}\left(C^{L1}_{X,L3},\ C^{Y}_{Z,L2}\right)$

$X = Y\ (\neq L3)$
$Z =$ qualsiasi (ma $\neq L2$)

Calcola $C^{L3}_{Z,L2}$

---

2) Silly $\quad c ::= y := 2 \mid c;c \mid \text{while } tt \text{ do } c$

Ass $\quad \dfrac{}{\langle y:=2, \sigma\rangle \longrightarrow \sigma[y/2]}$

Seq1 $\quad \dfrac{\langle c_0, \sigma\rangle \longrightarrow \langle c_0', \sigma'\rangle}{\langle c_0; c_1, \sigma\rangle \longrightarrow \langle c_0'; c_1, \sigma'\rangle}$

Seq2 $\quad \dfrac{\langle c_0, \sigma\rangle \longrightarrow \sigma'}{\langle c_0; c_1, \sigma\rangle \longrightarrow \langle c_1, \sigma'\rangle}$

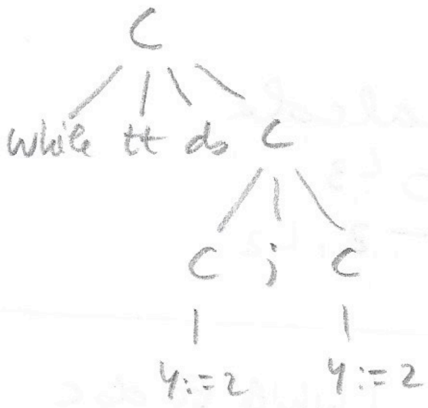
wh $\quad \dfrac{}{\langle \text{while } tt \text{ do } c, \sigma\rangle \longrightarrow \langle c; \text{while } tt \text{ do } c, \sigma\rangle}$

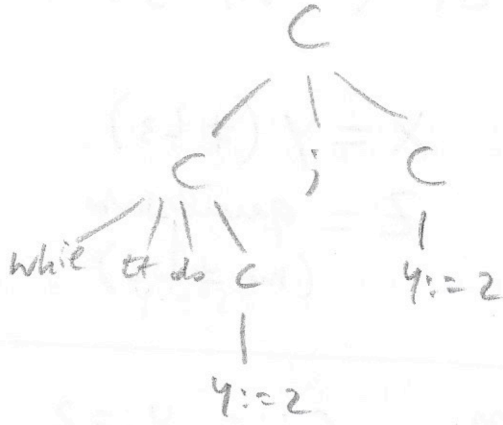
Le funzioni che può calcolare Silly sono

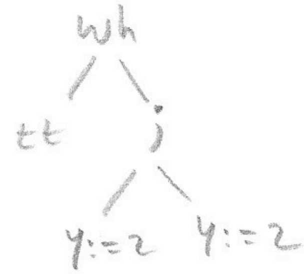
$f(\sigma) = \sigma[y/2]$ se non uso il while
$f(\sigma) =$ indefinita se uso il while

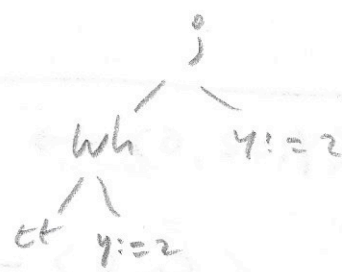
Il ling. Silly è non Turing-completo.
Ad es. $f(\sigma) = \sigma[y/1]$ non è calcolabile in Silly.

3) while tt do y:=2 ; y:=2




$\Longrightarrow$ ambigua

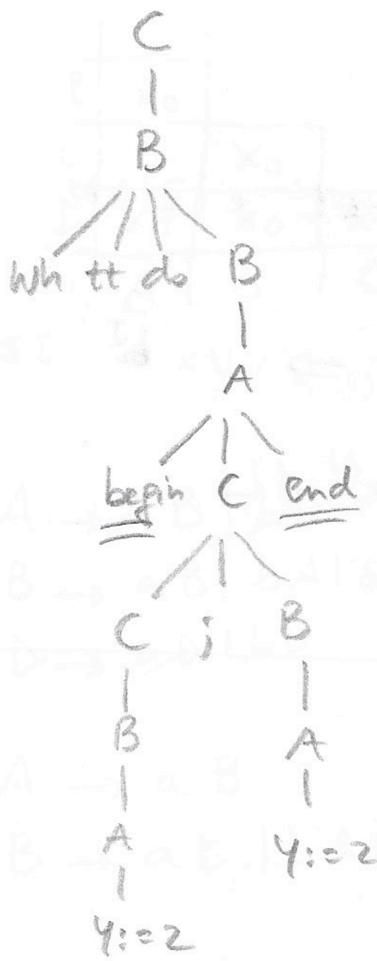
Per risolvere l'ambiguità dobbiamo:

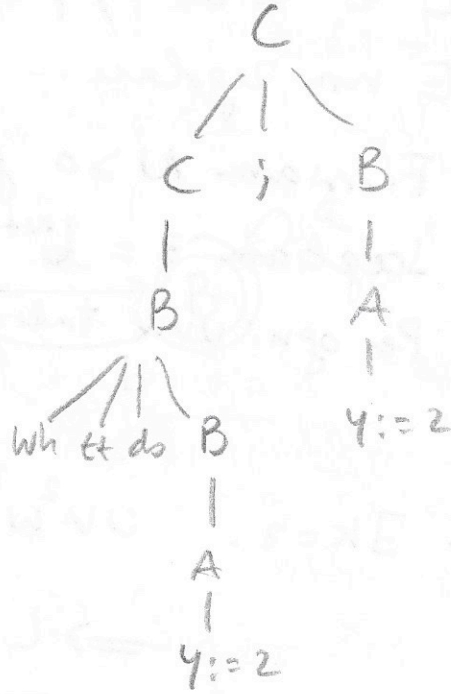
- fissare l'associatività di ; : ass sx
- decidere la precedenza tra ; e while: while > ;

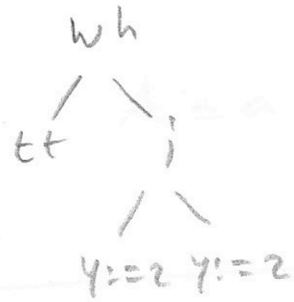
C ::= C ; B | B

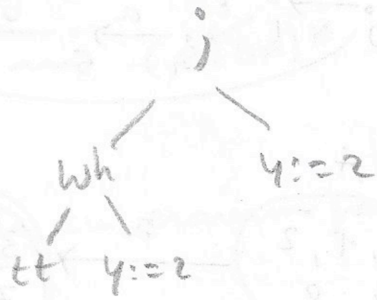
B ::= while tt do B | A

A ::= y:=2 | begin C end




4) $L = \{ b^{n+1} c^{m+1} a^{n} \mid n, m \geq 0 \}$

$S \to bA$ — $b^{n+1} c^{+} a^{n}$

$A \to bAa \mid C$ ~ $b^{n} c^{+} a^{n}$

$C \to c \mid cC$ ~ $c^{+}$

$S \to bSa \mid bC$

$C \to cC \mid c$

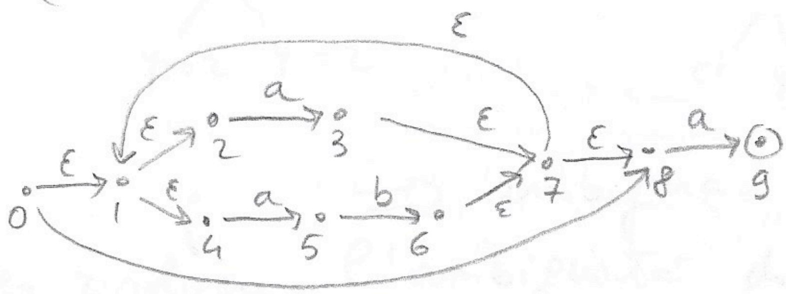
5) L è libero, perché generato da una gr. libera
È non regolare...

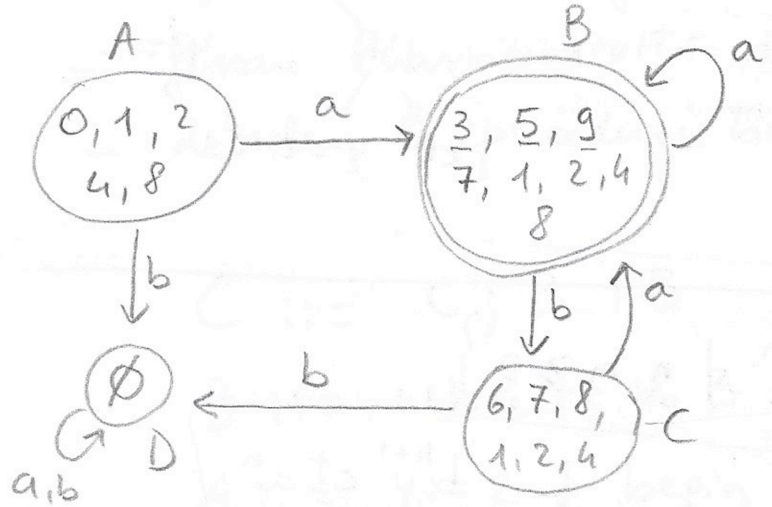
- Fissiamo $N > 0$ generico
- Scegliamo $z = b^{N+1} c^{N+1} a^{N}$ $\quad z \in L$ e $|z| > N$
- Per ogni $uvw$ tali che - $z = uvw$
  - $|uv| \leq N$
  - $|v| \geq 1$ $\Rightarrow v = b^{j}$ $j \geq 1$
- $\exists k = 2$. $uv^2w = b^{N+1+j} c^{N+1} a^{N} \notin L$

$\Longrightarrow$ L non è regolare

---

6) $(a|ab)^* a$

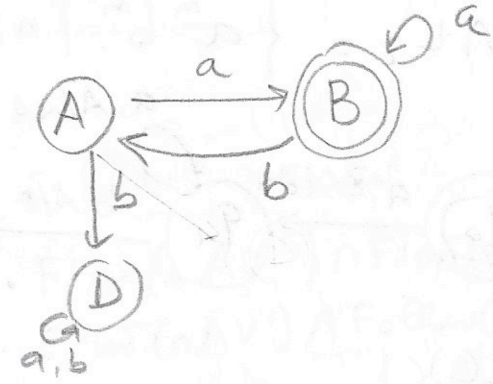


7)

| | A | B | C |
|---|---|---|---|
| B | $X_0$ | | |
| C | $0$ | $X_0$ | |
| D | $X_1$ | $X_0$ | $X_1$ |

$A \sim C$




$A \to aB \mid \cancel{bD}$
$B \to aB \mid bA \mid \varepsilon$
$\cancel{D \to aD \mid bD}$

$$\left.\begin{array}{l} A \to aB \\ B \to aB \mid bA \mid \varepsilon \end{array}\right] - \text{sostituzione}$$

$B \to aB \mid baB \mid \varepsilon$

$B = (a \mid ba) B \mid \varepsilon$

$\rightsquigarrow (a \mid ba)^*$

$A = aB = a(a \mid ba)^*$

$\cong (a \mid ab)^* a$ da cui eravamo partiti

---

8) $L = \{ a^n b^m \mid n \geq m \geq 0 \}$ non gode della prefix property. Infatti

$aabb, aab \in L$ ma $aab$ è prefisso di $aabb$

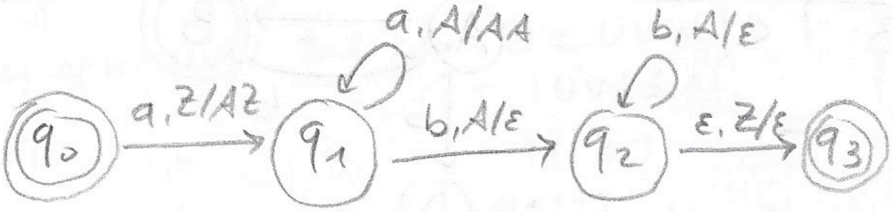
Poiché un linguaggio LR(0) gode della prefix property, deduciamo che L non è di classe LR(0).

9) $L_1 = \{a^n b^m \mid 0 \le m \le n\}$

$L_2 = \{a^n b^m \mid 0 \le n \le m\}$

$L_1 \cap L_2 = \{a^n b^n \mid n \ge 0\}$ lib. det.

1- DPDA




2- Gr. LL(1)

$S \to aSb \mid \varepsilon$ ] è LL(1)

$First(aSb) \cap First(\varepsilon) = \emptyset$

$First(aSb) \cap Follow(S) = \emptyset$

$\{a\} \cap \{b, \$\} = \emptyset$

Poiché i ling LL(1) sono lib. det., allora anche $L_1 \cap L_2$ è lib. det.

3) Gr. – SLR(1) ...

1. $V \to A \mid V \wedge A$
$A \to (V) \mid b$ ] G non è LL(1) perché ric sx

$First(A) \cap First(V \wedge A) \neq \emptyset$

$V \to AV'$
$V' \to \wedge AV' \mid \varepsilon$
$A \to (V) \mid b$ ] G' è LL(1) perché

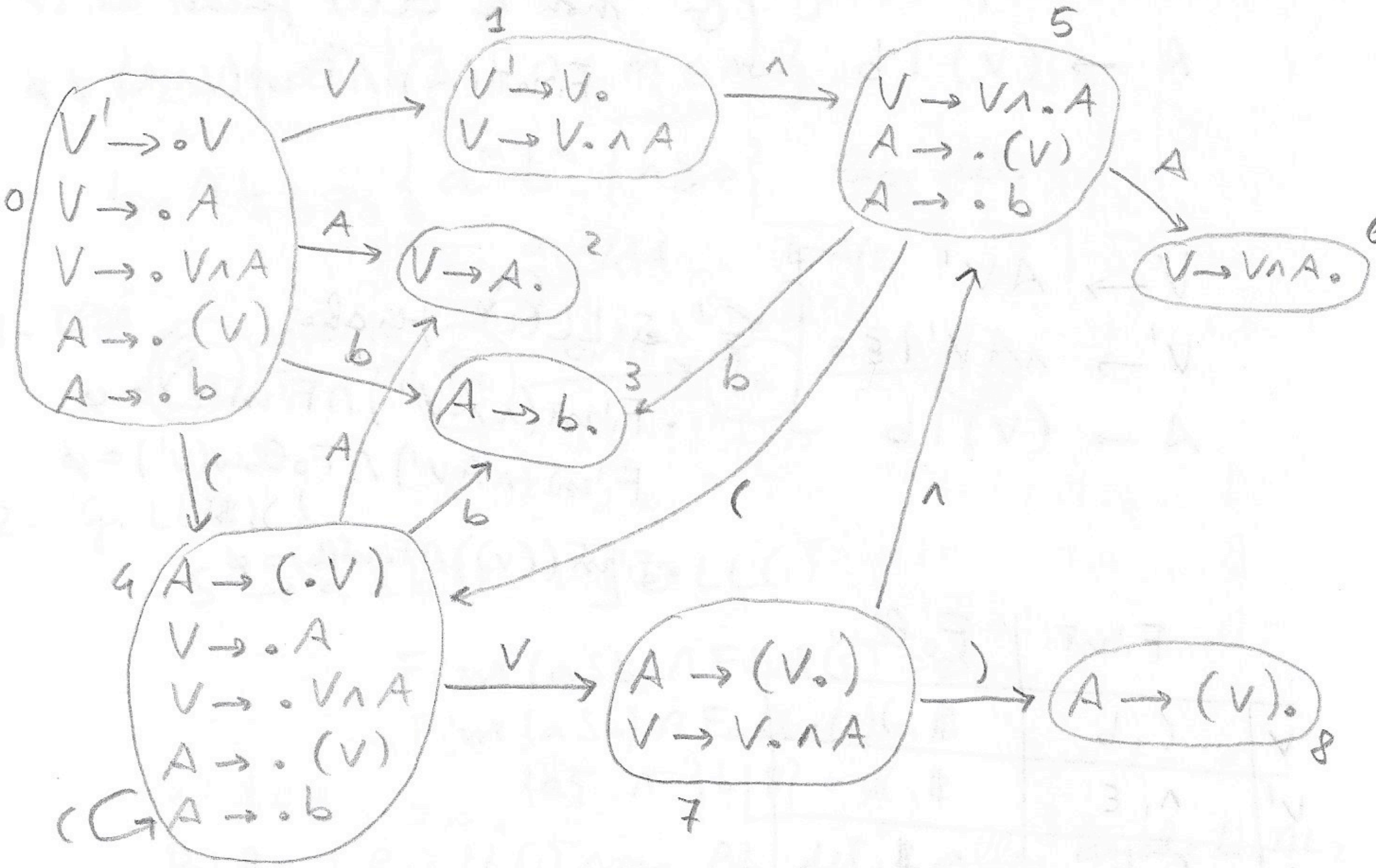
- $First(\wedge AV') \cap First(\varepsilon) = \emptyset$
  $First(\wedge AV') \cap Follow(V') = \emptyset$
  $\{ ), \$ \}$
- $First((V)) \cap First(b) = \emptyset$

| | First | Follow |
|---|---|---|
| V | (, b | \$, ) |
| V' | ∧, ε | \$, ) |
| A | (, b | ∧, \$, ) |

| | b | ( | ) | ∧ | \$ |
|---|---|---|---|---|---|
| V | $V \to AV'$ | $V \to AV'$ | | | |
| V' | | | $V' \to \varepsilon$ | $V' \to \wedge AV'$ | $V' \to \varepsilon$ |
| A | $A \to b$ | $A \to (V)$ | | | |

| | |
|---|---|
| (b)∧b\$ | V |
| | AV' |
| | (V)V' |
| b)∧b\$ | V)V' |
| | AV')V' |
| | bV')V' |
| )∧b\$ | V')V' |
| | )V' |
| ∧b\$ | V' |
| | ∧AV' |
| b\$ | AV' |
| | bV' |
| \$ | V' |
| | ε OK |

11) (0) $V' \to V$ (1) $V \to A$ (2) $V \to V \wedge A$ (3) $A \to (V)$ (4) $A \to b$




| | b | ( | ) | $\wedge$ | $ | V | A |
|---|---|---|---|---|---|---|---|
| 0 | s3 | s4 | | | | g1 | g2 |
| 1 | | | | s5 | acc | | |
| 2 | r1 | r1 | r1 | r1 | r1 | | |
| 3 | r4 | r4 | r4 | r4 | r4 | | |
| 4 | s3 | s4 | | | | g7 | g2 |
| 5 | s3 | s4 | | | | | g6 |
| 6 | r2 | r2 | r2 | r2 | r2 | | |
| 7 | | | s8 | s5 | | | |
| 8 | r3 | r3 | r3 | r3 | r3 | | |

(0, $\varepsilon$, (b$\wedge$b)$)

(04, (, b$\wedge$b)$)

(043, ($\underbrace{b}_{A}$, $\wedge$b)$)

(042, ($\underbrace{A}_{V}$, $\wedge$b)$)

(047, (V, $\wedge$b)$)

(0475, (V$\wedge$, b)$)

(04753, (V$\wedge\underbrace{b}_{A}$, )$)

(04756, ($\underbrace{V \wedge A}_{V}$, )$)

(047, (V, )$)

(0478, $\underbrace{(V)}_{A}$, $)

(02, $\underbrace{A}_{V}$, $)

(01, V, $) accept!