

③ Grammaires non contextuelles déterministes

① Soit $G = (V, \Sigma, S, R)$ une grammaire quelconque.
 on crée un nouveau symbole initial $S_0 \notin V$ et on pose
 $G' = (V \cup \{S_0\}, \Sigma, S_0, R \cup \{S_0 \rightarrow S\})$ G' ne fait jamais
 apparaitre son symbole initial dans un membre de droite
 et $\mathcal{L}(G) = \mathcal{L}(G')$

② On mène par récurrence sur $k \geq 1$ que tout mot
 généré par la grammaire $G_1 = (V_1, \Sigma_1, R_1, R_2)$ avec une
 dérivation de longueur k est ab^k .

init: $k=1$ une seule étape de dérivation $R \rightarrow ab$ de $\frac{k}{1}$ m.
hén: Soit m généré à partir de R en $k-1$ étapes $R \rightarrow m$, $\frac{k}{1}$ m.
 on a alors $m_1 = aRb$ car doit être dérivé ensuite
 donc $\exists u \in \Sigma_1^+ t \cdot q \cdot m_1 = a u b$ et $R \rightarrow^k u$ par HR
 donc $u = a^k b^k$ et donc $m = a^{k+1} b^{k+1}$.

On mène de même par récurrence sur $k \geq 1$ que
 le seul mot généré par $G_1'' = (V_1, \Sigma_1, \bar{I}, R_1)$ avec une
 dérivation de longueur k est $a^k b^k$.

Soit $m \in \mathcal{L}(G_1)$ alors il existe une dérivation $S \Rightarrow^k m$.
 La première étape de cette dérivation est $S \Rightarrow R$ ou $S \Rightarrow T$
 dans le premier cas, on aura $m \in \mathcal{L}(G_1')$ et donc $\exists k \geq 1, q$
 $m = a^k b^k$ et dans le deuxième cas, $m \in \mathcal{L}(G_1'')$ donc $\exists k \geq 1$
 $t \cdot q \cdot m = a^k b^k$.

On a bien $m \cdot q \quad \mathcal{L}(G_1) \subseteq \{a^n b^n : n \in \mathbb{N}^+\} \cup \{a^n b^{2n} : n \in \mathbb{N}\}$

③ Réciproquement, soit $m \in \{a^n b^n : n \in \mathbb{N}^+\} \cup \{a^n b^{2n} : n \in \mathbb{N}^+\}$.
 Distinguons les deux cas.
 • $m = a^n b^n$ avec $n \geq 1$ alors on mène par
 récurrence sur n que $R \Rightarrow^* m$.
 Si $n=1$, on a bien $R \Rightarrow ab$ donc $R \Rightarrow^1 a^1 b^1$.
 Si $n > 1$, $R \Rightarrow^* a^{n-1} b^{n-1}$ alors $R \Rightarrow^* a^{n-1} b^{n-1} a^{n-1} b^{n-1}$
 et donc $R \Rightarrow^* a^n b^n$.
 • Si $m = a^n b^{2n}$ avec $n \geq 1$ on mène de même
 par récurrence sur n que $T \Rightarrow^* m$.

Ainsi si $m = a^n b^n$ avec $n \geq 1$ $S \Rightarrow R \Rightarrow^* m$ et donc $m \in \mathcal{L}(G)$
 si $m = a^n b^{2n}$ avec $n \geq 1$ $S \Rightarrow T \Rightarrow^* m$ et donc $m \in \mathcal{L}(G)$

On a bien $\mathcal{L}(G_1) \supseteq \{a^n b^n : n \in \mathbb{N}^+\} \cup \{a^n b^{2n} : n \in \mathbb{N}^+\}$
 et donc par double inclusion:

$$\mathcal{L}(G_1) = \{a^n b^n : n \in \mathbb{N}^+\} \cup \{a^n b^{2n} : n \in \mathbb{N}^+\}$$

$$\begin{aligned} S &\Rightarrow R \Rightarrow aRb \\ &\Rightarrow aARbb \\ &\Rightarrow aAabbb \end{aligned}$$

$$\begin{aligned} S &\Rightarrow T \Rightarrow aTbb \\ &\Rightarrow aabbbb \end{aligned}$$

(6)

begin - 1 (chaq' m) }

int $l = \text{stack}(m)$

int $i = 0$, (compte les 'a')

while ($i < l$ && $m[i] == 'a'$) }

$i++$;

}

int $j = 0$, (compte les 'b')

while ($i+j < l$ && $m[i+j] == 'b'$) }

$j++$;

}

return ($i+j == l$ && $((i == j) || (2+i == j))$);

}

(7)

$aabbba \hookrightarrow aTbb \hookrightarrow T \hookrightarrow S$

(8)

soit $m \hookrightarrow u_1 \hookrightarrow u_2 \hookrightarrow \dots \hookrightarrow u_k = S$ une réduction gauche alors on a $S = u_k \Rightarrow u_{k-1} \Rightarrow \dots \Rightarrow u_1 \Rightarrow u_0 = m$.

Si cette dérivation n'est pas droite alors il existe

$i \in \llbracket 1, k \rrbracket$ t.q. $u_i = v_1 v_2 v_3 v_4$ et

$u_{i+1} = v_1 h v_2 v_3 v_4$ c'est-à-dire

qu'on dérive la non terminal v_2 avant v_1 .

Ici v_1, v_2, v_3 et h sont dans $(V \cup \Sigma)^+$ et v_1, v_2 de V

on a alors $u_{i+1} = v_1 h v_2 v_3 v_4 \hookrightarrow v_1 v_1 v_2 v_3 v_4 = u_i$

mais alors cela signifie que v_2 est obtenu précédemment lors d'une réduction $u_i \hookrightarrow u_{i+1}$ avec $j < i-1$ alors que la réduction de h était possible, la réduction

ne revait alors pas gauche.

(9)

Si une grammaire est non ambigu alors elle admet une unique dérivation droite et donc d'après la question précédente, elle admet une unique réduction gauche.

(10)

$u \hookrightarrow v$ signifie $v \Rightarrow u$ et par définition de la dérivation cela signifie qu'il existe une règle $T \rightarrow h$ et deux mots x et y dans $(V \cup \Sigma)^+$ tels que $v = xTy$ et $u = xhy$.

(11)

Si y contient un symbole non terminal alors celui-ci est apparu lors d'une étape précédente comme résultat d'une réduction. Mais alors la réduction ne peut pas être gauche car elle a réduit un facteur à droite d'un facteur réductible. Il lui même n'était déjà présent au moment de la réduction considérée ou d'un de ses facteurs non terminal. Sinon.

(16)

réduct° droite:

$S_2 \Rightarrow E$
 $\Rightarrow E + T$
 $\Rightarrow E + a$
 $\Rightarrow E + T + a$
 $\Rightarrow E + T + a + a$
 $\Rightarrow E + a + a + a$
 $\Rightarrow T + a + a + a$
 $\Rightarrow a + a + a + a$

$S_2 \Rightarrow E$
 $\Rightarrow E + T$
 $\Rightarrow E + a$
 $\Rightarrow E + T + a$
 $\Rightarrow E + T + a + a$
 $\Rightarrow E + a + a + a$
 $\Rightarrow T + a + a + a$
 $\Rightarrow a + a + a + a$

(d). $a_0 a_1 b b b b b b \in \mathcal{L}(G) \Sigma^4$ donc c'est un mot valide

(47) $\mathcal{L}(G_2')$ est dérivé par l'expression $a(*a)^*$

(18) $\mathcal{L}(G_n)$ est décrit par l'expression régulière suivante

$a(a^*)^* (a(a^*)^*)^*$ ou plus simplement $a((+1^*)a)^*$

(19) $\mathcal{L}(G_2) = \mathcal{L}(G_1)$ est donc décrit par l'expression régulière de la question précédente. Il s'agit de tous les mots composés de a avec un + ou un $\cdot$ entre chaque a . Soit $m \in \mathcal{L}(G_2)$ alors m est de la forme :

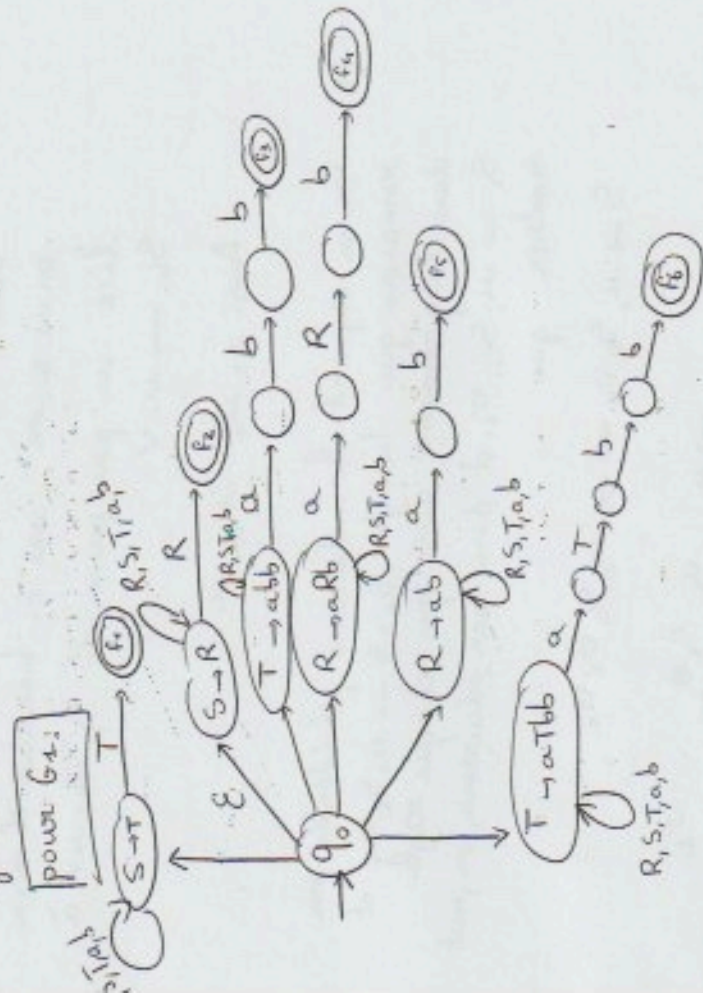
$$a + a + \dots + a + a + \dots + a + a + \dots + a$$

ter
symbole
+ du mot

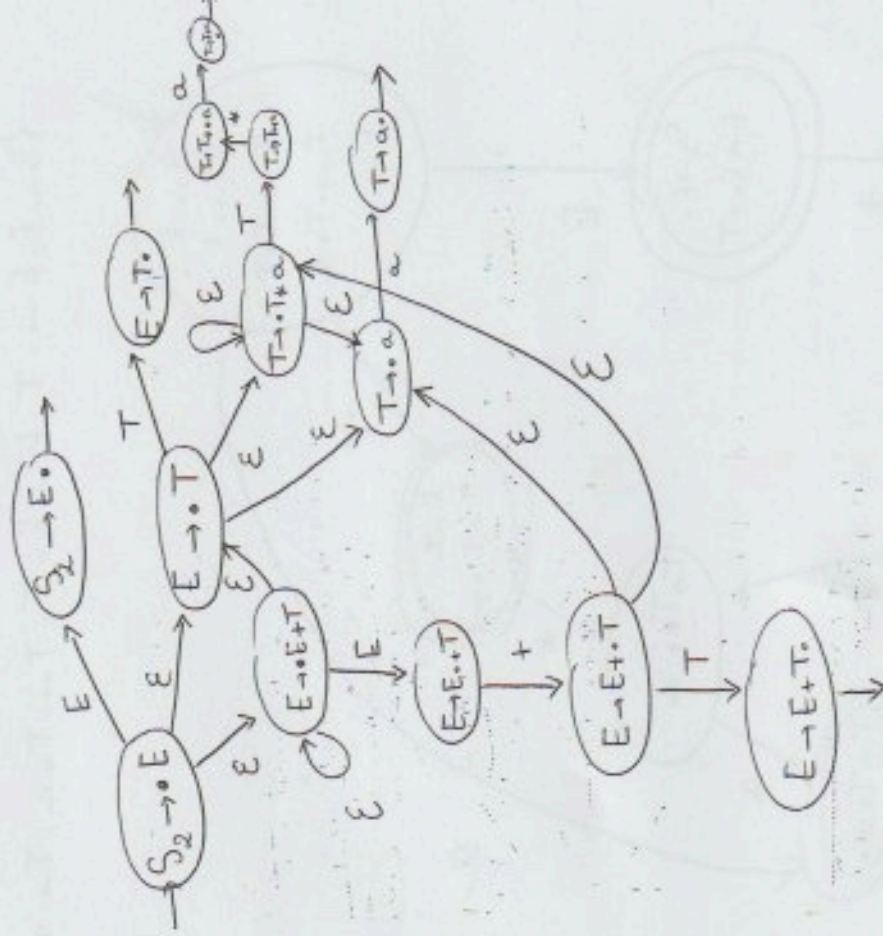
Supposons qu'il y a k occurrences de α dans m
alors l'arbre est de la forme suivante

10

- 22) on crée un état initial suivi d'une E -transition vers un sommet pour chaque règle de la grammaire, chacun de ces sommets admet une boucle sur lui-même pour chaque lettre de $V \cup \Sigma$ et admet une succession de transitions étiquetées par le membre droit de la règle et aboutissant dans un état final.



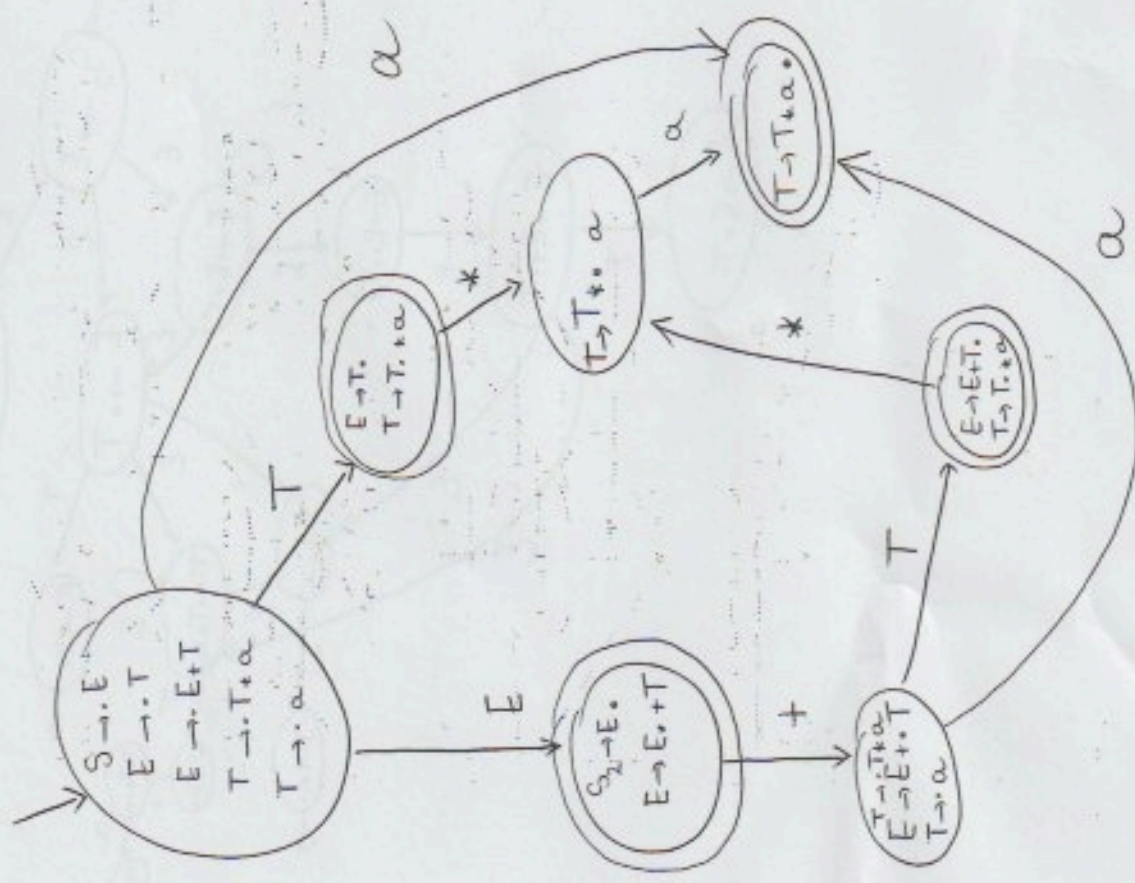
cf page suivante.



24

La clôture transitive par ε -transit° de l'état initial est:

$\{S \rightarrow \cdot E, E \rightarrow \cdot T, E \rightarrow \cdot E+T, T \rightarrow \cdot T+a, T \rightarrow \cdot a\}$



11

25

(a)

Pour arriver dans l'état $S_1 \rightarrow u_1 \cdot S_2 \cdot a$, on a lu le mot u_1 puis on passe dans un état $S_2 \rightarrow \cdot c$ avec une ε -transition et on va devoir lire u_2 pour aller jusqu'à l'état $S_2 \rightarrow u_2 \cdot S_3 \cdot a$.

On va ainsi lire $u_1 \dots u_{k-1}$ pour arriver dans l'état $S_{k-1} \rightarrow u_{k-1} \cdot S_k \cdot a$ puis on va suivre une ε -transition et on va lire u pour passer de $S_k \rightarrow \cdot u \cdot a$ à $S_k \rightarrow u \cdot \checkmark$.

Ainsi $z = u_1 \dots u_{k-1} u$.

(b) Si on reprend les notations précédentes on remarque que la règle $S_1 \rightarrow u_1 S_2 a$ et dans la grammaire ainsi que chaque règle $S_i \rightarrow u_i S_{i+1} a_i$ et donc par récurrence, on peut montrer que

$$S \Rightarrow u_1 S_2 a_1 \Rightarrow u_1 u_2 S_3 a_2 a_1$$

$$\begin{aligned} &\Rightarrow u_1 \dots u_{k-1} S_k a_{k-1} \dots a_1 \\ &\Rightarrow \underbrace{u_1 \dots u_{k-1} u}_{= z} \underbrace{a_{k-1} \dots a_1}_y \end{aligned}$$

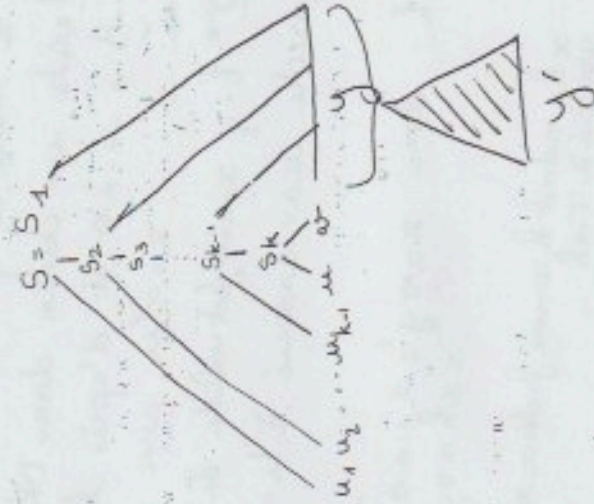
en prenant $y = a_{k-1} \dots a_1$ on a bien $z a y$ dérivable depuis S et en réduisant le facteur u on obtient $z a y \hookrightarrow u_1 \dots u_{k-1} T y = x T y$

12

(c) le y de la question précédente n'est pas forcément dans Σ^+ mais on peut dériver complètement chacun de ses non terminaux afin d'obtenir $y \Rightarrow^* y' \in \Sigma^+$.

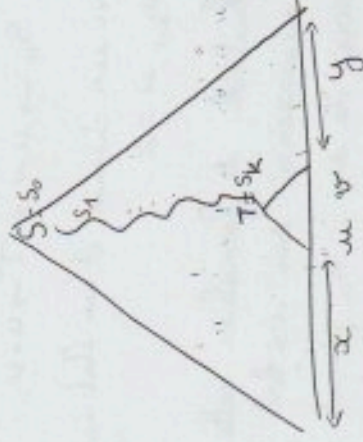
On a alors $S \Rightarrow^* x u v y \Rightarrow^* x u v y'$ donc $x u v y'$ valide et l'existence de la règle $T \rightarrow u v$ donc $u v$ est un facteur réductible de $x u v y'$.

D'après la 1^{re} précédente, l'autre de dérivat^{on} pour $x u v y'$ ressemble à cela :



Ainsi aucun facteur à gauche de $u v$ n'est réductible et il n'est bien un facteur gauche.

(a)



(b) Si on avait un non terminal à gauche du chemin $S \Rightarrow^* T$ qui n'est pas une feuille et donc un facteur de x alors cela impliquerait qu'un facteur de x serait réductible et que $u v$ ne serait pas le facteur réductible gauche.

(c) On a donc $x = u_0 \dots u_{k-1}$ et l'existence des règles $S_i \rightarrow u_i S_{i+1} v_i$.

Lisons le mot z dans $DK(G)$ en suivant le chemin suivant qui existe :

on démarre en $S_0 \rightarrow u_0 S_1 v_0$ et on lit u_0 jusqu'à arriver en $S_1 \rightarrow u_1 S_2 v_1$ on suit alors une E-transit^{ion} vers $S_1 \rightarrow u_1 S_2 v_1$ d'où on lit u_1 pour arriver en $S_2 \rightarrow u_2 S_3 v_2$ d'où on suit une E-transit^{ion}.

En continuant ainsi, on arrive en $S_k \rightarrow u_k S_{k+1} v_{k+1}$ à la lecture de u_k d'où on suit une

(28)

(a) q est acceptant dans l'automate des parties donc doit contenir un état acceptant de $DK(\mathcal{G})$ c-à-d de la forme $T \rightarrow h$. D'après la q 27, si z mène à q alors il existe un chemin dans $DK(\mathcal{G})$ vers l'état $T \rightarrow h$ et donc $z = xh$ avec $x \in (00\Sigma)^*$ et $\exists y \in \Sigma^* t q$ h soit un facteur réductible gauche de xhy associé à la règle $T \rightarrow h$.

(b) Puisque z mène à q contenant $B \rightarrow u.a.v$ alors il existe un chemin dans $DK(\mathcal{G})$ le menant à $B \rightarrow u.a.v$ et d'après la q 25 on sait que $z = x'ue$ avec $x' \in (00\Sigma)^*$ et $\exists y' \in \Sigma^* t q$ $x = x'ue$ est valide de facteur réductible gauche uav associé à la règle $B \rightarrow u.a.v$.

(c) $z = x'u = xh$ donc $zay' = x'uaoy' = xhaoy'$

on remarque que: xhy admet h comme facteur réductible gauche associé à $T \rightarrow h$

et $xhaoy' = x'uaoy'$

admet uav comme facteur réductible gauche associé à $B \rightarrow u.a.v$.

On a mis en évidence un mot valide xhy avec une réduction et un mot valide xhy ("y" = aoy') avec un effet h et strictement avec une réduct° gauche $\neq$ [à gauche de uav].
Cependant, on ne peut

(15)

ε -transition vers $S_k \rightarrow u.v$ d'où l'on lit u pour arriver en $S_k \rightarrow u.v = T \rightarrow u.v$.

On a donc suivi un chemin d'un état initial à $T \rightarrow u.v$ en lisant $u_1 \dots u_{k-1} u = z$.

Il serait souhaitable de formaliser cette idée en faisant une récurrence sur $i \leq k-2$ garantissant que la lecture de $u_1 \dots u_i$ permet d'accéder à l'état $S_i \rightarrow u_i.v_i$ et donc $S_{i+1} \rightarrow u_{i+1}.v_{i+1}$ (avec ε -transition).

$\hookrightarrow$ $i=0$ est fait et si en lisant $u_1 \dots u_i$ on arrive en $S_{i+1} \rightarrow u_{i+1}.v_{i+1}$ alors en lisant u_{i+1} depuis cet état on peut accéder à $S_{i+2} \rightarrow u_{i+2}.v_{i+2}$ puis à $S_{i+3} \rightarrow u_{i+3}.v_{i+3}$ par ε -transition.

Il suffit alors de conclure comme précédemment.

(27) On applique les résultats des questions 25 et 26 avec $u = h$ et $v = \varepsilon$.

On obtient donc: 25 $\Rightarrow$ Si on arrive en $T \rightarrow h$ à la lecture de z alors $\exists x \in (00\Sigma)^* t q$ $z = xh$ et $\exists y \in \Sigma^* t q$ xhy est valide de facteur réductible gauche h , la règle associée est alors $T \rightarrow h$.

26 $\Rightarrow$ Si xhy est valide de facteur réductible gauche h associé à $T \rightarrow h$ et $y \in \Sigma^*$ alors la lecture de $z = xh$ permet d'accéder à l'état $T \rightarrow h$ de $DK(\mathcal{G})$.

(16)

pas encore conclure que la grammaire est déterministe car $y'' \notin \Sigma^*$

(d) II faut donc dériver v en $v' \in \Sigma^*$ et poser le mot $m \cdot \overline{x' a v' y'}$ qui se réduit de la manière suivante et est donc valide :

$$\overline{x' a v' y'} = \overline{x' a v' y'} \hookrightarrow \overline{x' a v' y' \hookrightarrow x' b y' z' s}$$

Ce mot est valide
 de la forme xhy''' avec $y'' \in \Sigma^+$
 et n'admet pas h comme facteur réductible gauche.
 car son facteur réductible gauche doit contenir a sinon ua ou ma serait
 un facteur réductible gauche de $x'ua'y''$ qui serait le u qui est celui de
 $x'ua'y''$.

(29) On reprend l'automate de la

$q^0 24$ et on observe qu'un état acceptant contient par exemple $E \rightarrow E \cdot + T$ et donc d'après la $q^0 28$, la grammaire G_1 considérée n'est pas déterministe.

3.4 Grammaire

(30) Ce sont les expressions bien parenthésées suivies de 1

```

31) bool genre-3(char* m) {
    int n = strlen(m);
    int cpt = 0;
    for (int i = 0; i < m[i]; i++) {
        if (m[i] == '(') {
            cpt++;
        }
    }
}

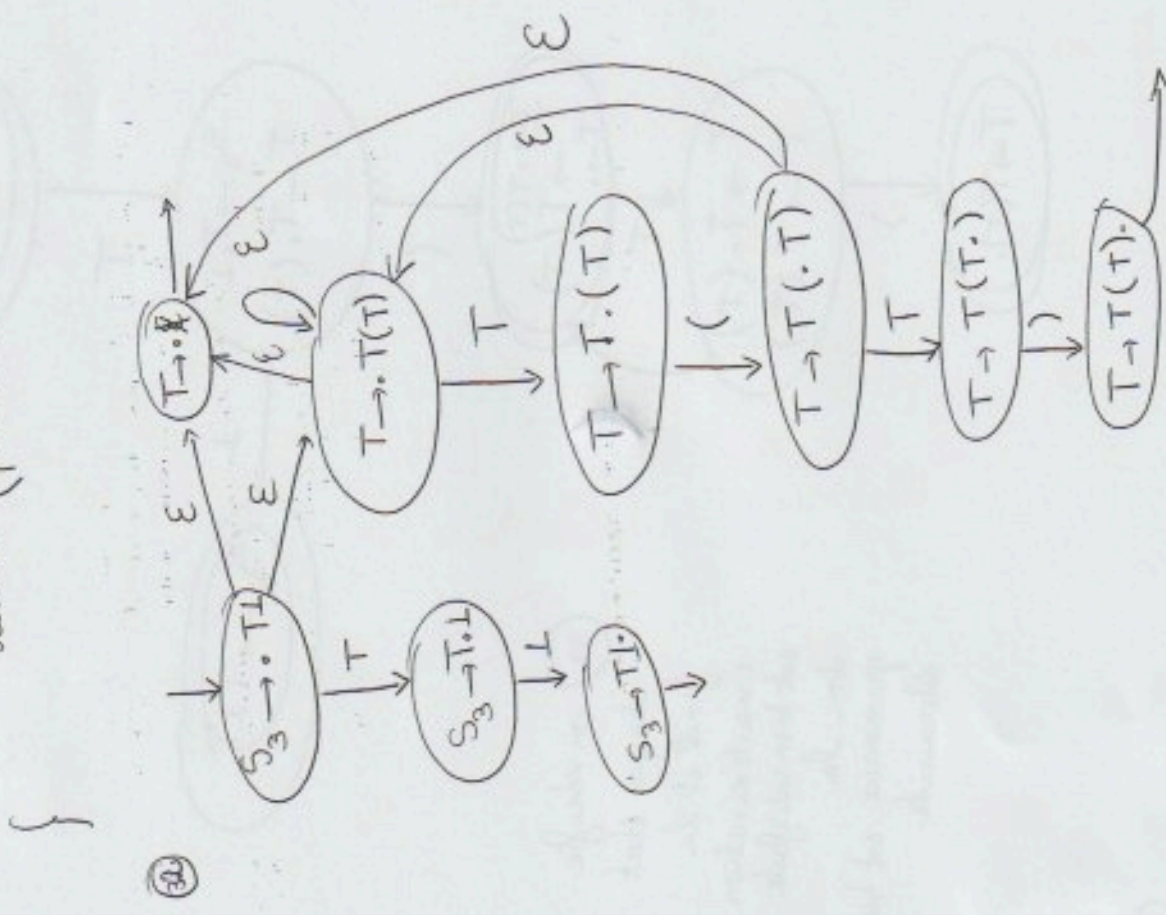
```

```
else if (m[i] = '\') {
    cpt--;
    if (cpt < 0) return false;
}
```

```

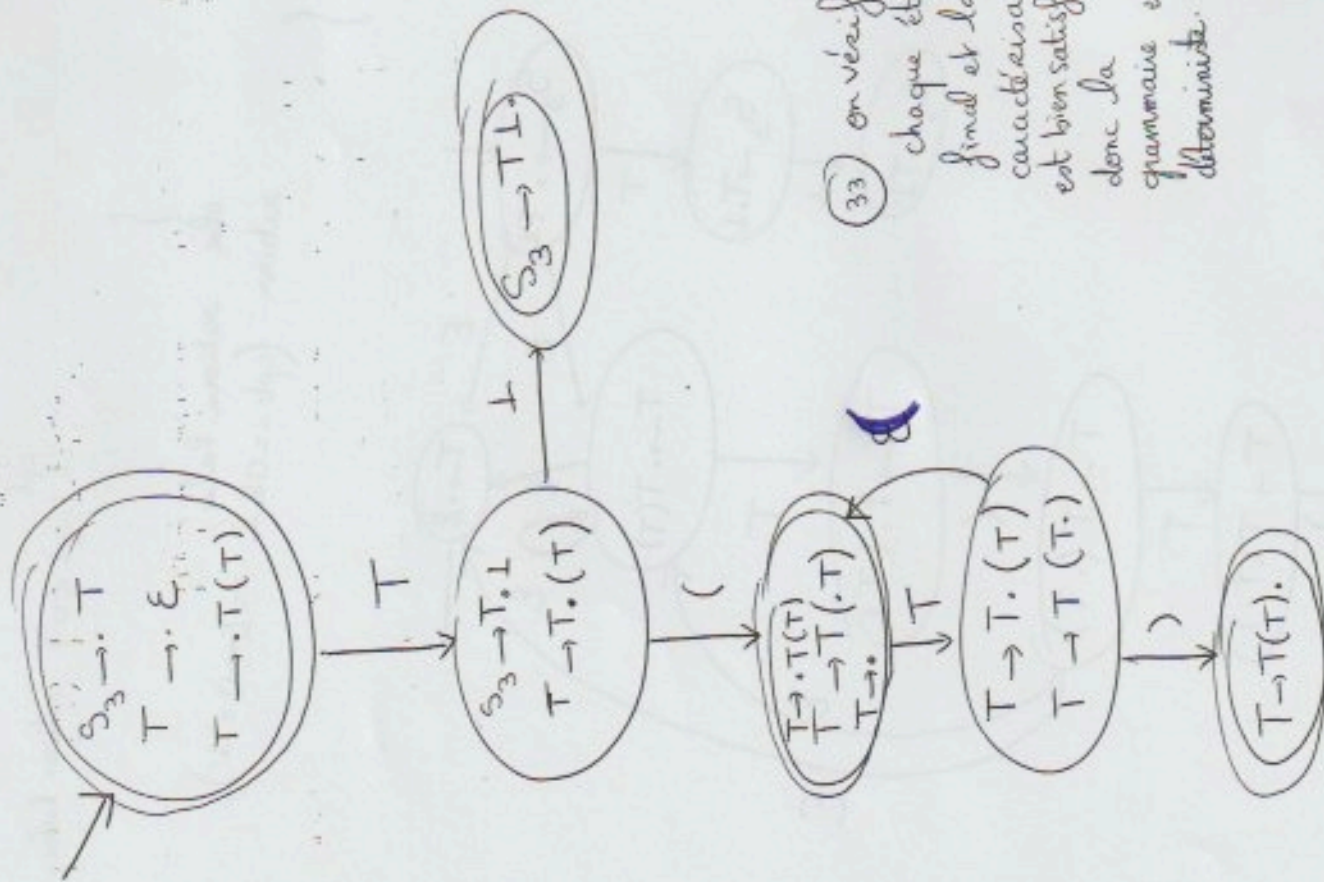
    }
    else return false;
    return (cpt == 0) & m[m-1] == '1';
}

```



(5)

D'etermined



(33) on vérifie
chaque état
final et la
caractérisation
est bien satisfaisante
donc la
chaine est bien
déterministe.