**Per altra banda, si $\langle R', \# \rangle$ satisfà la condició de l'enunciat, llavors $\# \xrightarrow[R']{+} \#$. Escollim aquí la derivació més curta de la forma $\# \xrightarrow[R']{+} \#$, de manera que cap paraula intermedia és $\#$. Per les regles que tenim al sistema de rescriptura, aquesta derivació ha de ser de la forma $\# \xrightarrow[\# \to \$u\$]{} \$u\$ \xrightarrow[R']{*} \$v\$ \xrightarrow[\$v\$ \to \#]{} \#$. Es pot veure inductivament que en la derivació intermedia $\$u\$ \xrightarrow[R']{*} \$v\$$ totes les paraules són de la forma $\$w\$$, on $w$ no conté $\#$ ni $\$$: si en un cert punt abans del final es cumpleix la condició, llavors no es pot fer servir $\# \to \$u\$$ per a obtenir la següent paraula, però tampoc $\$v\$ \to \#$, ja que si no $\#$ seria paraula intermedia. De l'anterior també es dedueix que totes les regles que s'apliquen en aquesta subderivació són de $R$. Per tant $\$u\$ \xrightarrow[R]{*} \$v\$$, i com que les regles de $R$ no contenen $\$$, tambe tenim que $u \xrightarrow[R]{*} v$, de manera que $\langle R, u, v \rangle$ és del problema de `Mots`.**

(Examen de Juny-2007)

Tingueu present en tots els exercicis que les respostes s'han de justificar (a menys que es digui el contrari). La mera presentació d'una proposta de resultat no és suficient.

1. (4.5 punts en total) Els següents apartats, tot i estar relacionats, es poden contestar de forma independent, però recomanem començar pel primer, ja que dóna intuició per als tres darrers.

   a) (0.5) Considerem el llenguatge sobre $\{a, b\}^*$ dels mots que acaben en $aaaa$. Escriviu diréctament l'autòmat determinista mínim que el genera (en aquest cas no cal que poseu cap justificació).

   b) (1.5) Donat un nombre $k \geqslant 0$, el llenguatge dels mots que acaben en $k$ $a$'s, $\{a, b\}^* a^k$, és regular. Justifiqueu que hi ha un autòmat determinista amb $k + 1$ estats que el reconeix.

   c) (1.5) Demostreu que no hi ha cap autòmat determinista amb menys que $k+1$ estats que reconegui $\{a, b\}^* a^k$, o, equivalentment, que el de l'apartat anterior és mínim.

   d) (1) Demostreu que el llenguatge dels mots tals que la segona meitat d'ells conté només $a$'s, és a dir $\{wa^{|w|} | w \in \{a, b\}$, no és regular.

**Resposta:**

**a)**

| | $a$ | $b$ |
|---|---|---|
| $q_0$ | $q_1$ | $q_0$ |
| $q_1$ | $q_2$ | $q_0$ |
| $q_2$ | $q_3$ | $q_0$ |
| $q_3$ | $q_4$ | $q_0$ |
| $\dagger q_4$ | $q_4$ | $q_0$ |

**b) Generalitzant la idea de l'apartat anterior, només cal que recordem quantes $a$'s hem vist fins al moment. A l'estat $q_i$ hi arribem amb paraules que tenen $i$ $a$'s al final i no $i + 1$ $a$'s al final, excepte per al cas de $q_k$, que hi arribem amb paraules que tenen $k$ o més $a$'s al final. És fàcil demostrar inductivament que aquestes propietats es cumpleixen per a l'autòmat que descrivim.**

| | $a$ | $b$ |
|---|---|---|
| $q_0$ | $q_1$ | $q_0$ |
| $q_1$ | $q_2$ | $q_0$ |
| $\vdots$ | | |
| $q_i$ | $q_{i+1}$ | $q_0$ |
| $\vdots$ | | |
| $q_{k-1}$ | $q_k$ | $q_0$ |
| $\dagger q_k$ | $q_k$ | $q_0$ |

c) **Suposem que existeix un autòmat $A$ amb menys de $k+1$ estats i que reconeix $\{a,b\}^*a^k$. Sigui $q_0$ l'estat inicial de $A$. Donat que hi han menys de $k+1$ estats, dues de les $k+1$ paraules $\lambda$,$a$,$aa$, $\dots$ ,$a^k$ ens porten al mateix estat. Per tant, existeixen $0 \leqslant i < j \leqslant k$ tals que $q_0 a^i = q_0 a^j$. Donat que $a^k$ és del llenguatge, $q_0 a^k$ és acceptador. Però $q_0 a^k = q_0(a^{j+(k-j)}) = q_0(a^j a^{k-j}) = (q_0 a^j)a^{k-j} = (q_0 a^i)a^{k-j} = q_0(a^i a^{k-j}) = q_0(a^{k-j+i})$. Així doncs, la paraula $a^{k-j+i}$ també ens porta a estat acceptador. Però com que $i < j$, tenim que $k-j+i < k$, de manera que $a^{k-j+i}$ no acaba amb $k$ $a$'s, i per contra és acceptada: contradicció.**

**Com a resposta alternativa, si apliquem l'algorisme de minimització a l'autòmat de l'apartat anterior, veiem que es comprova que és mínim:**

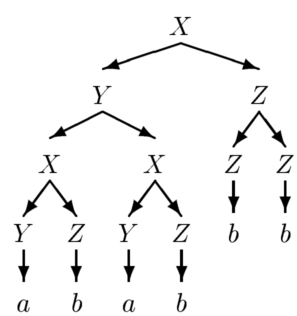
0-**distingibles:** $\{q_0, \dots, q_{k-1}\}$ $\{q_k\}$
1-**distingibles:** $\{q_0, \dots, q_{k-2}\}$ $\{q_{k-1}\}$ $\{q_k\}$
2-**distingibles:** $\{q_0, \dots, q_{k-3}\}$ $\{q_{k-2}\}$ $\{q_{k-1}\}$ $\{q_k\}$
$\vdots$
$k$-**distingibles:** $\{q_0\}$ $\{q_1\} \dots \{q_{k-1}\}$ $\{q_k\}$

d) **Ho fem pel lema de bombament. Fixat $N$, escollim la paraula $w = b^N a^N$, que és de $\{wa^{|w|} | w \in \{a,b\}$, i considerem una factorització qualsevol $w = xyz$ tal que $|xy| \leqslant N$ i $|y| \geqslant 1$. Sota aquestes condicions, $x$ és de la forma $b^i$ per a un cert $i \geqslant 0$, $y$ és de la forma $b^j$ per a un cert $j > 0$, i $z$ és de la forma $b^{N-i-j}a^N$. La paraula $xy^2z$ és $b^{N+j}a^N$, i donat que $j > 0$, també tenim que $N+j > N$, de manera que $xy^2z$ no és de $\{wa^{|w|} | w \in \{a,b\}$. Això conclou la prova.**

2. (2 punts) Considerem la següent gramàtica $G_{\texttt{exemple}}$:

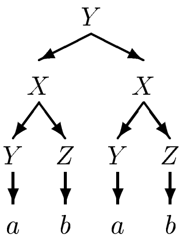
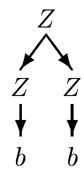
$$\begin{array}{lcl}
X & \to & YZ \\
Y & \to & XX \mid a \\
Z & \to & ZZ \mid b
\end{array}$$

I el següent arbre de derivació del mot terminal $ababbb$ a partir de $G_{\texttt{exemple}}$.



El recorregut en preordre d'aquest arbre és el mot: $XYXYaZbXYaZbZZbZb$.

En general, el recorregut en preordre d'un arbre, es defineix recursivament com el símbol de l'arrel, seguit del recorregut en preordre del primer fill, seguit del recorregut en preordre del segon fill, i així successivament fins al recorregut en preordre de l'últim fill.

Fixeu-vos que per al cas particular del recorregut en preordre de l'arbre de derivació anterior primer ens ve l'arrel $X$, seguida del recorregut en preordre de  que és $YXYaZbXYaZb$, seguit del recorregut en preordre de  que és $ZZbZb$.

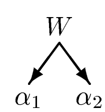
Donat un arbre de derivació $A$, anomenem $\texttt{preordre}(A)$ al mot que codifica el recorregut en preordre de $A$. Donada una gramàtica $G$, si anomenem $\mathbf{A}(G)$ al conjunt de tots els possibles arbres de derivació de $G$ en mots terminals, llavors $\{\texttt{preordre}(A)|A \in \mathbf{A}(G)\}$ defineix el conjunt dels mots que representen recorreguts en preordre de tots els arbres de derivació de $G$ en mots terminals. Anomenem $\texttt{preordre}(G)$ a aquest conjunt, és a dir, $\texttt{preordre}(G) = \{\texttt{preordre}(A)|A \in \mathbf{A}(G)\}$

Considerant l'exemple de gramàtica anterior $G_{\texttt{exemple}}$, construïu una nova gramàtica que generi $\texttt{preordre}(G_{\texttt{exemple}})$. Fixeu-vos que en la nova gramàtica caldrà que, no només $a$ i $b$ siguin símbols terminals, sinó que també $X$, $Y$ i $Z$ siguin terminals, quan de fet eren variables de la gramàtica original.

**Resposta:**

**Utilitzarem les variables $X'$, $Y'$ i $Z'$, que generaran $\texttt{Preordre}(L_{G_{\texttt{exemple}},X})$, $\texttt{Preordre}\ (L_{G_{\texttt{exemple}},Y})$ i $\texttt{Preordre}(L_{G_{\texttt{exemple}},X})$, respectivament; on $L_{G,W}$ representa el llenguatge generat per la gramàtica $G$ considerant $W$ com a símbol inicial. La gramàtica que construïm és:**

$$
\begin{array}{lcccc}
X' & \to & XY'Z' & & \\
Y' & \to & YX'X' & | & Ya \\
Z' & \to & ZZ'Z' & | & Zb
\end{array}
$$

**Tot i que es podria demostrar per inducció que es genera $\texttt{preordre}(G_{\texttt{exemple}})$, en aquest cas ens limitem a observar que la utilització d'una regla qualsevol $W \to W_1W_2$ de $G_{\texttt{exemple}}$ porta a un arbre de derivació de la forma**  **on $\alpha_1$ i $\alpha_2$ són àrbres de derivació generats desde $W_1$ i $W_2$, respectivament, i que el recorregut en preordre d'aquest arbre és $W\texttt{Preordre}(\alpha_1)\texttt{Preordre}(\alpha_2)$. Amb la regla $W' \to WW_1'W_2'$ podem assumir inductivament que $W_1'$ i $W_2'$ generen $\texttt{Preordre}(L_{G_{\texttt{exemple}},W_1'})$ i $\texttt{Preordre}(L_{G_{\texttt{exemple}},W_2'})$.**

**Com a resposta alternativa podriem haver donat:**

$$
\begin{array}{lcccc}
S & \to & XX' & & \\
X' & \to & YY'ZZ' & & \\
Y' & \to & XX'XX' & | & a \\
Z' & \to & ZZ'ZZ' & | & b
\end{array}
$$

3. (1.5 punts) Classifiqueu com a decidible, semi-decidible pero no decidible, o no semi-decidible, els següent problema.

$$\{x | \forall y : (M_x(y) \downarrow \Rightarrow M_x(y) < y)\}$$

(Les entrades i sortides de les màquines de turing s'interpreten com a nombres naturals.)

**Resposta:**

**Demostrarem que no és ni semi-decidible reduïnt desde $\overline{\mathtt{K}}$ de la següent manera. Si $x$ és l'element d'entrada, donem com a sortida de la reducció el programa $p$ següent.**

```
entrada y
   Simular Mx(x)
   sortida y
```

**Si $x$ pertany a $\overline{\mathtt{K}}$, llavors $\mathbf{M}_x(x)\uparrow$, de manera que el programa $p$ no s'atura per cap entrada, i llavors compleix la condició $\forall y : (M_p(y) \downarrow \Rightarrow M_p(y) < y)$, concloent així que $p$ pertany al nostre llenguatge.**

**En canvi, si $x$ no pertany a $\overline{\mathtt{K}}$, llavors $\mathbf{M}_x(x)\downarrow$, de manera que el programa $p$ s'atura amb totes les entrades, i de fet es comporta com la funció identitat, de manera que no compleix la condició $\forall y : (M_p(y) \downarrow \Rightarrow M_p(y) < y)$, concloent així que $p$ no pertany al nostre llenguatge.**

**Com a resposta alternativa, pel teorema de Rice, donat que $L$ és un conjunt d'índexos, que no és ni el total ni el buit, i que conté els nombres que representen la funció totalment indefinida, resulta que $L$ no és ni semi-decidible.**

**Més en detall, $L$ és un conjunt d'indexos perque en la condició sempre que apareix $x$, ho fa amb el contexte $M_x$, és a dir, una condició només sobre el comportament del programa $M_x$, independentment de la seva implementació. $L$ no és buit perque conté, per exemple, a qualsevol nombre de programa que es penja sempre. $L$ no és el total perque no conté per exemple als nombres de programes que donen com a sortida la pròpia entrada, per a totes les entrades.**

4. (2 punts en total) **En aquest cas heu d'escollir qué contestar d'entre dos problemes alternatius**

**Opció 1:**

En els exercicis-control del curs hem vist que és indecidible el problema de, donades dues gramàtiques, saber si tenen intersecció no buida.

$$\{\langle G_1, G_2\rangle \mid \mathbf{L}(G_1) \cap \mathbf{L}(G_2) \neq \emptyset\}$$

Ara volem demostrar que també ho és el problema de, donades dues gramàtiques, saber si tenen com a intersecció un nombre infinit de mots; és a dir, si hi ha infinits mots generats per totes dues gramàtiques.

$$\{\langle G_1, G_2\rangle \mid |\mathbf{L}(G_1) \cap \mathbf{L}(G_2)| = \infty\}$$

a) (1) Demostreu amb un contraexemple que el següent intent de reducció desde intersecció no buida no funciona. (És a dir, que no és cert que una entrada del problema del intersecció no buida té resposta afirmativa si i només si la seva transformada en una entrada del problema que hem enunciat també en té).

   **Donada la entrada $\langle G_1, G_2\rangle$ de intersecció no buida, construïm una nova entrada $\langle G_1', G_2'\rangle$ per a intersecció infinita així:**

   **Si $S_1$ és el símbol inicial de $G_1$, llavors $G_1'$ té una nova variable $S_1'$ com a símbol inicial, i les seves regles són les de $G_1$ més $S_1' \to S_1S_1'|\lambda$.**

   **Si $S_2$ és el símbol inicial de $G_2$, llavors $G_2'$ té una nova variable $S_2'$ com a símbol inicial, i les seves regles són les de $G_2$ més $S_2' \to S_2S_2'|\lambda$.**

b) (1) Doneu una reducció des del problema de intersecció no buida i justifiqueu que és correcta. Pista: és recomanable utilitzar algún símbol nou per a aconseguir-ho.

**Opció 2:**

Expliqueu formalment, amb generalitat i claretat, cóm es pot obtenir un autòmat amb pila indeterminista que reconegui el llenguatge generat per una gramàtica donada.

**Resposta:**

**Opció 1:**

a) **Podem tenir com a entrada les gramàtiques $S_1 \to a$ i $S_2 \to aa$. La intersecció és buida, i en canvi les corresponents**

$$\begin{array}{lcl} S_1' & \to & S_1S_1' \mid \lambda \\ S_1 & \to & a \end{array}$$

   **i**

$$\begin{array}{lcl} S_2' & \to & S_2S_2' \mid \lambda \\ S_2 & \to & aa \end{array}$$

   **Generen infinites paraules comunes: $\{a^{2n}\}$.**

b) **Donada la entrada $\langle G_1, G_2\rangle$ de intersecció no buida, agafem un símbol nou # que no aparegui ni a $G_1$ ni a $G_2$, i construïm una nova entrada $\langle G_1', G_2'\rangle$ per a intersecció infinita així:**

   **Si $S_1$ és el símbol inicial de $G_1$, llavors $G_1'$ té una nova variable $S_1'$ com a símbol inicial, i les seves regles són les de $G_1$ més $S_1' \to S_1\#S_1'|S_1$.**

   **Si $S_2$ és el símbol inicial de $G_2$, llavors $G_2'$ té una nova variable $S_2'$ com a símbol inicial, i les seves regles són les de $G_2$ més $S_2' \to S_2\#S_2'|S_2$.**

   **Si $G_1$ i $G_2$ generen una paraula comuna $w$, llavors $G_1'$ i $G_2'$ generen infinites paraules comunes de la forma $w\#w\#\dots\#w\#w$.**

   **Per la direcció contraria, si $G_1'$ i $G_2'$ generen infinites paraules comunes, agafem qualsevol d'elles, que serà de la forma $w$ o $w\#\dots$, on $w$ no conté #. Aquesta $w$ és generable tant desde $G_1'$ com desde $G_2'$.**

   **Com a resposta alternativa també podiem haver creat com a $G_1'$ les regles de $G_1$ més $S_1' \to \#S_1'|S_1$, i com a $G_2'$ les regles de $G_2$ més $S_2' \to \#S_2'|S_2$.**

**Opció 2:**

**A partir d'una CFG $G = \langle V, \Sigma, P, S\rangle$ podem construir de la manera següent un NPDA $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$ tal que $L(M) = L(G)$:**

- **el conjunt d'estats $Q = \{q_0, p, f\}$ (estat inicial, estat de procés i estat acceptador);**
- **l'alfabet de pila $\Gamma = \Sigma \cup V \cup \{Z_0\}$ on $Z_0 \notin \Sigma \cup V$ és el símbol de fons de pila;**
- **el conjunt d'estats acceptadors $F = \{f\}$;**
- **la funció de transició $\delta$ està format per:**

  $Z_0 q_0 \vdash Z_0 S p$ **(una $\lambda$-transició que posa la variable inicial al cim de la pila en l'estat de procés);**

  $\{Zp \vdash \alpha^R p |\ Z \to \alpha \in P\}$ **(una $\lambda$-transició per a cada producció de la variable $Z$ al cim de la pila);**

  $\{apa \vdash p |\ a \in \Sigma\}$ **(si el cim de la pila és un terminal, l'autòmat ha de verificar que sigui igual al símbol en curs del mot d'entrada i avançar el capçal);**

  $Z_0 p \vdash f$ **(una $\lambda$-transició que permet passar a l'estat acceptador quan tot el mot ha estat processat).**