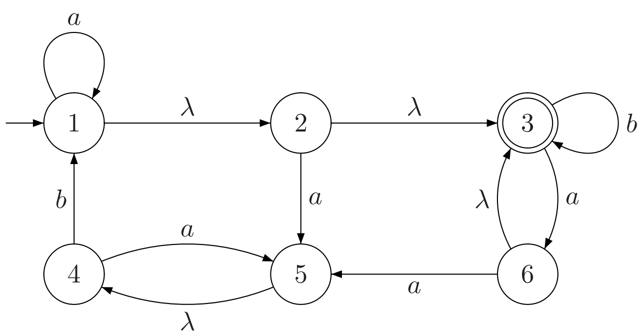
Teoría de Autómatas y Lenguajes Formales
Ingeniería Técnica en Informática de Sistemas.
Facultad de Informática, UCM. Examen final, Junio 2005

Primer parcial:

1. **(0.9 puntos)** Dado el siguiente $AFN-\lambda$,



aplica un método sistemático para obtener un AFN equivalente. No es necesario que muestres los pasos de tu método ni que muestres el diagrama del autómata obtenido. Como respuesta a este ejercicio, basta con que muestres la función de transición de estados del AFN obtenido, indicando además los estados inicial y de aceptación.

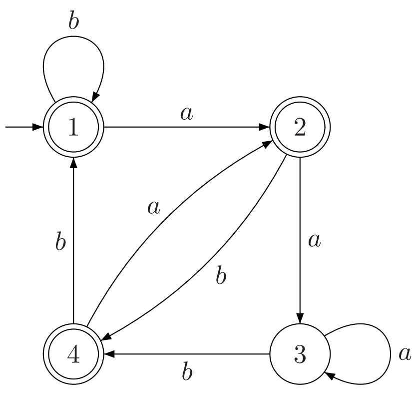
2. **(2.1 puntos)** Para cada uno de los siguientes lenguajes, indica si es regular o no. En caso de serlo, muestra una expresión regular <u>y</u> un autómata finito para ese lenguaje. Indica el tipo de autómata finito que has utilizado. En caso de no ser regular, demuéstralo por el lema del bombeo para lenguajes regulares <u>y</u> a través de un conjunto infinito de cadenas distinguibles dos a dos (y, como es habitual, mostrando por qué esas cadenas dos a dos son distinguibles).

   (a) $L_1 = \{a^i b^j | j \text{ es el resto de dividir } i \text{ entre } 2\}$

   (b) $L_2 = \{a^i b a^j | 2 \cdot j = i\}$

   (c) $L_3 = \{a^i b^j a^j | i \geq 0, j \geq 0\}$

3. **(1.5 puntos)** Aplica algún método automatizable para hallar la expresión regular que caracteriza el lenguaje que reconoce el siguiente AFD.



4. **(0.5 puntos)** Inventa un método sistemático y automatizable que, dadas dos expresiones regulares er_1 y er_2, te permita obtener una expresión er_3 tal que el lenguaje representado por er_3 sea la intersección de los dos lenguajes representados por er_1 y er_2. Para ello, haz que los pasos de tu algoritmo consistan en utilizar otros algoritmos conocidos.

Segundo parcial:

1. **(1.5 puntos)** Considera el siguiente problema, donde el alfabeto de entrada es $\Sigma = \{a, b\}$: Dada una secuencia de símbolos, devolver la secuencia que resulta de tomar por orden un símbolo de cada tres, comenzando por el primero.

   Por ejemplo, si la entrada es *abaabbbab* entonces la salida debe ser *aab*, pues los símbolos en posiciones primera, cuarta y séptima son, por este orden, *aab*. Otro ejemplo: Si la entrada es *bbbaabaababbaaa* entonces la salida debe ser *baaaa*, pues los símbolos en posiciones 1, 4, 7, 10 y 13 son, por este orden, *baaaa*. Un último ejemplo: Si la entrada es *ab*, la salida es *a*.

   Crea una máquina de Turing que resuelva este problema. Como es habitual, el resultado final deberá estar rodeado por infinitos símbolos blancos por ambos lados, y el cursor deberá apuntar al símbolo más a la izquierda del resultado cuando la máquina se pare. Recuerda que puedes usar tantos símbolos que no pertenezcan a Σ como quieras: X, Y, $\#$, @, etc.

2. **(2 puntos)** Para cada uno de los siguientes lenguajes, indica si es incontextual o no. En caso de serlo, muestra una gramática incontextual <u>*y*</u> un autómata con pila para el lenguaje. Indica el tipo de autómata con pila utilizado así como el símbolo inicial de su pila. En caso de no ser incontextual, demuéstralo por el lema del bombeo para lenguajes incontextuales.

   (a) $L_1 = \{a^i b^j a^{i'} b^{j'} | i \leq i', j \leq j'\}$

   (b) $L_2 = \{a^i b^j c^k | i > j \text{ ó } i \leq k\}$

3. **(0.5 puntos)** Considera las GI tales que en la parte derecha de cada regla aparece *como mucho una variable* (aunque el número de terminales no está limitado). Por ejemplo, las reglas $I \to aaAa$ y $B \to \lambda$ son reglas válidas, pero $I \to aAbbBa$ y $A \to aaAA$ no lo son. Sea LH_1 el conjunto de lenguajes que puede representarse por alguna de esas gramáticas. ¿Se cumple $LH_1 = LI$? En caso afirmativo, explica por qué. En caso contrario, muestra un lenguaje L tal que $L \in LI$ pero $L \notin LH_1$.

   Considera las GI tales que en la parte derecha de cada regla aparecen *como mucho dos variables* (iguales o distintas). Por ejemplo, las cuatro reglas consideradas antes serían válidas. Sea LH_2 el conjunto de lenguajes que puede representarse por alguna de esas gramáticas. ¿Se cumple $LH_2 = LI$? En caso afirmativo, explica por qué. En caso contrario, muestra un lenguaje L tal que $L \in LI$ pero $L \notin LH_2$.

   Considera las GI tales que en la parte derecha de cada regla aparecen *como mucho tres variables* (iguales o distintas). Sea LH_3 el conjunto de lenguajes que puede representarse por alguna de esas gramáticas. ¿Se cumple $LH_3 = LI$? En caso afirmativo, explica por qué. En caso contrario, muestra un lenguaje L tal que $L \in LI$ pero $L \notin LH_3$.

4. **(1 punto)** Aplica un método sistemático para pasar la siguiente gramática incontextual a forma normal de Chomsky. El símbolo inicial de la gramática es I. No es necesario que muestres cómo aplicas cada paso de tu método. Sólo se pide que muestres la gramática que resulta *después* de aplicar *cada uno* de los pasos, así como la gramática final.

$$\begin{array}{lcl} I & \to & aAB \\ A & \to & BAb \mid \lambda \mid B \\ B & \to & a \mid b \mid CD \\ C & \to & ba \\ D & \to & DD \end{array}$$

Solución

Primer parcial:

1. Función de transición de estados del AFN:

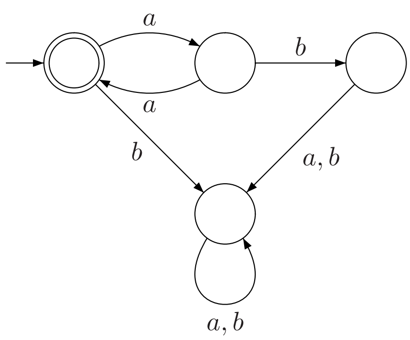
	a	b
1	$\{1,2,3,4,5,6\}$	$\{3\}$
2	$\{3,4,5,6\}$	$\{3\}$
3	$\{3,6\}$	$\{3\}$
4	$\{4,5\}$	$\{1,2,3\}$
5	$\{4,5\}$	$\{1,2,3\}$
6	$\{4,5,6\}$	$\{3\}$

- El estado inicial del AFN es el 1.
- Los estados de aceptación del AFN son el 3 (porque ya lo era en el AFN-λ) y el 1 (porque es el inicial y, *a su vez*, algún estado de aceptación pertenecía a su cierre-λ en el AFN-λ).

2. (a $L_1 = \{a^i b^j | j \text{ es el resto de dividir } i \text{ entre } 2\}$
   L_1 *es regular.*
   - Una posible expresión regular es:

$$(aa)^* + (aa)^* ab$$

   - Para el AF, escogemos el siguiente AFD:



   (b $L_2 = \{a^i b a^j | 2 \cdot j = i\}$
   L_2 *no es regular.*
   - Lo probamos con el lema del bombeo:
     - Fijada una cierta longitud n, escogemos la palabra $w = a^{2n}ba^n$ que cumple $|w| \geq n$ y $w \in L_2$. Veamos si podemos hacer cumplir a la vez las siguientes condiciones, donde $w = xyz$:
     (1) $|xy| \leq n$
     (2) $y \neq \lambda$
     (3) $\forall i \geq 0 : xy^iz \in L_2$
     - Por (1), xy sólo puede abarcar a's del grupo a^{2n}. Por (2), y tiene al menos una de dichas a's. Por (3), debería cumplirse $xy^2z \in L_2$. En dicha palabra, el nº de a's antes de la b es $2n + |y| \geq 2n + 1$, y el nº de a's después de la b es n. Dado que $2n + |y| \neq 2n$, $xy^2z \notin L_2$ y la condición (3) no se cumple. Por tanto, no es posible partir la palabra w en 3 pedazos xyz tal que se cumplan (1), (2) y (3) a la vez.

- Lo probamos ahora por medio del conjunto infinito:
  - Consideremos el conjunto $A = \{b, aab, aaaab, aaaaaab, \ldots\}$. Cualesquiera dos cadenas pertenecientes a este conjunto infinito son distinguibles en el lenguaje L_2. Dos cadenas del conjunto w_1 y w_2 seguirán la forma $w_1 = a^{2i}b$ y $w_2 = a^{2j}b$ con $i \neq j$. Busquemos una misma continuación para ambas tal que una de las palabras resultantes se acepte y la otra no. Esta será a^i: se cumple $w_1a^i = a^{2i}ba^i \in L_2$, pero $w_2a^i = a^{2j}ba^i \notin L_2$, pues $i \neq j$. Por tanto, A es un conjunto infinito de cadenas distinguibles dos a dos $\Rightarrow L_2$ no es regular.

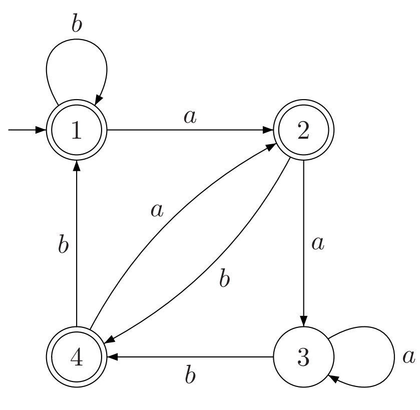
(c $L_3 = \{a^ib^ja^j | i \geq 0, j \geq 0\}$

L_3 *no es regular.*

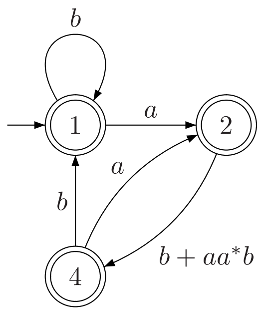
- Lo probamos con el lema del bombeo:
  - Fijada una cierta longitud n, escogemos la palabra $w = a^nb^na^n$, entonces después sucederá que y sólo tendrá *a's del 1º grupo* a^n. En tal caso, cualquier cantidad de veces que bombeemos y nos dará una palabra que pertenece a L_3, y no conseguiremos refutar nada. Debemos escoger, por tanto, otra palabra w tal que $w \in L_3$, $|w| \geq n$, y la parte a bombear llegue, al menos, al grupo de b's.
  - Tomamos $w = b^na^n \in L_3$. Veamos si podemos partir w en 3 pedazos xyz tales que se cumplan las condiciones (1), (2) y (3). (ver apartado (b))
  - Por (1), xy sólo puede abarcar b's del grupo b^n. Por (2), y tiene al menos una de dichas b's. Por (3), debería cumplirse $xy^2z \in L_3$. En dicha palabra, el nº de b's del 1º grupo será $n + |y| \geq n + 1$, y el nº de a's del 2º grupo será n. Dado que $n + |y| \neq n$, $xy^2z \notin L_3$ y la condición (3) no se cumple. Por tanto, no es posible partir la palabra w en 3 pedazos xyz tal que se cumplan (1), (2) y (3) a la vez.
- Lo probamos ahora por medio del conjunto infinito:
  - Consideremos el conjunto $A = \{\lambda, b, bb, bbb, bbbb, \ldots\}$ (también valdría, por ejemplo, $A = \{\lambda, ab, aabb, aaabbb, aaaabbbb, \ldots\}$ con unos argumentos similares). Dos cadenas del conjunto w_1 y w_2 seguirán la forma $w_1 = b^i$ y $w_2 = b^j$ con $i \neq j$. Para ambas, usaremos la continuación a^i. Se cumple $w_1a^i = b^ia^i \in L_3$, pero $w_2a^i = b^ja^i \notin L_3$, pues $i \neq j$. Por tanto, A es un conjunto infinito de cadenas distinguibles dos a dos $\Rightarrow L_3$ no es regular.

3. Lo haremos por el método de eliminación de estados. Nos quedaremos con un autómata por cada estado de aceptación, en el que estarán *dicho estado* y el *estado inicial* únicamente.

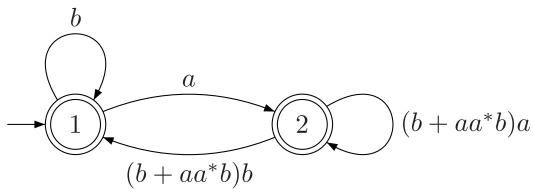
- Autómata inicial:



- Eliminamos el estado 3:



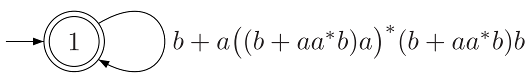
- Ahora eliminamos el estado 4:



- De ahí sacamos la ER del estado 2:

$$ER_2 = b^*a\big((b+aa^*b)a\big)^*\Big((b+aa^*b)bb^*a\big((b+aa^*b)a\big)^*\Big)^*$$

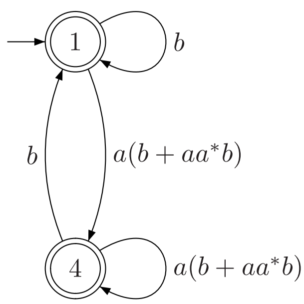
- Sobre el autómata anterior eliminamos el estado 2:



- Esto nos da la ER del estado 1:

$$ER_1 = \Big(b + a\big((b+aa^*b)a\big)^*(b+aa^*b)b\Big)^*$$

- Sobre el autómata con 1,2 y 4 eliminamos el estado 2:



- De ahí obtenemos la ER del estado 4:

$$ER_4 = b^*a(b + aa^*b)\big(a(b + aa^*b)\big)^*\Big(bb^*a(b + aa^*b)\big(a(b + aa^*b)\big)^*\Big)^*$$

- La expresión regular del AFD propuesto en el ejercicio será la suma de las tres ER obtenidas:

$$ER = ER_1 + ER_2 + ER_4$$

4. Proponemos el siguiente método

    - Para cada una de las dos ER dadas, calculamos su correspondiente AFN-λ.
    - Para cada uno de ellos, obtenemos sus respectivos AFN.
    - Para cada uno de ellos, obtenemos sus correspondientes AFD.
    - Dados esos dos AFD's, creamos un AFD nuevo donde los estados representan parejas de estados de cada uno de los AFD's dados. Desde cada estado, cada transición conduce a un nuevo estado donde cada estado de su pareja es aquél que habríamos alcanzado en su correspondiente AFD. Dado que queremos obtener la *intersección*, los estados de aceptación del nuevo AFD serán aquellos en los que cada estado de la pareja era de aceptación en su correspondiente AFD(*ambos* deberán serlo a la vez).
    - A partir del nuevo AFD obtenemos su expresión regular. Esta expresión representa un lenguaje que es la intersección de los lenguajes representados por er_1 y er_2.

Segundo parcial:

1. El plan de nuestra MT será el siguiente:

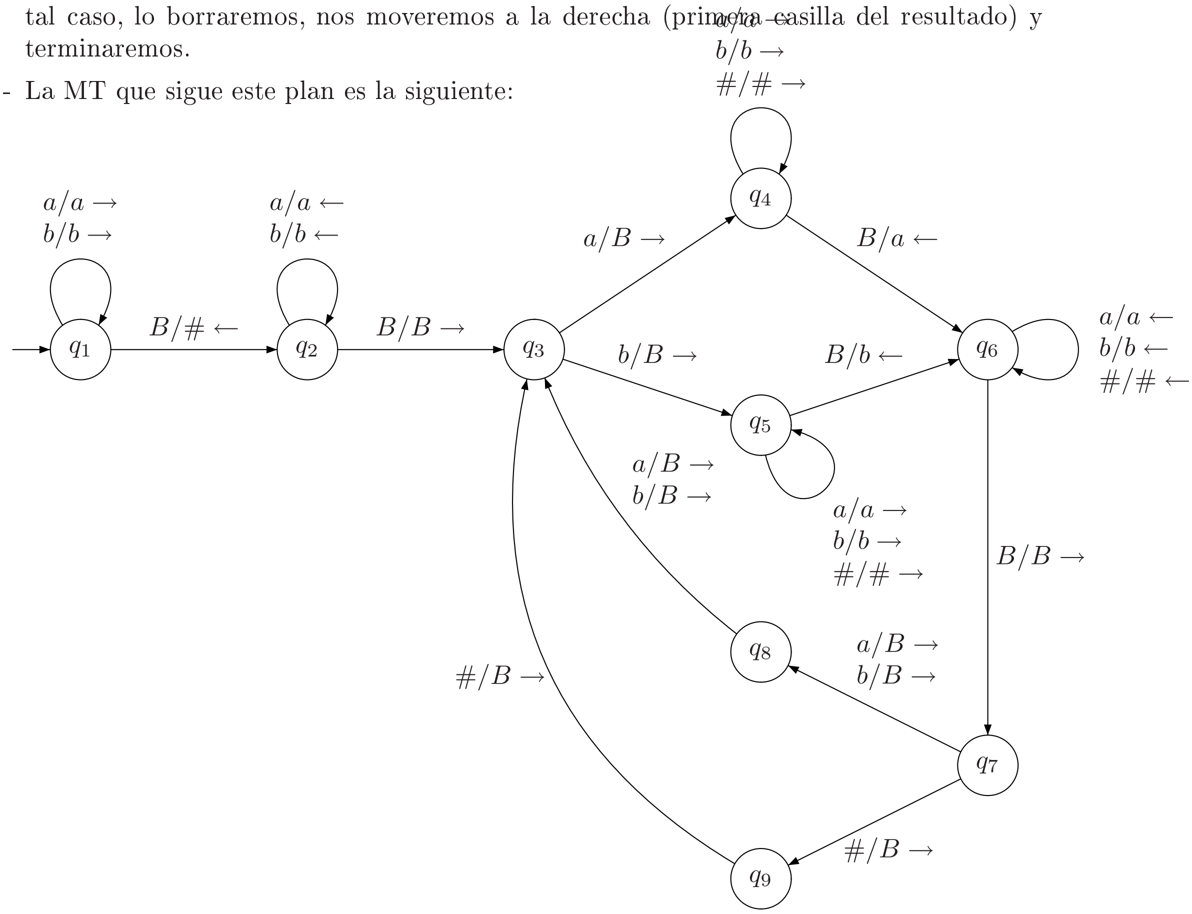
   - Iremos al extremo derecho para poner un separador (#) y volveremos al principio. A la derecha del separadoriremos construyendo el resultado.
   - Recorreremos la cinta para tomar un símbolo de cada tres. Con el objetivo de simplificar el borrado final de los símbolos que no forman parte del resultado, iremos borrando los símbolos a medida que los recorremos. (Esto podemos hacerlo porque después no necesitamos la entrada original para nada)

     (1). Tomaremos el primer símbolo que encontremos y lo copiaremos al final de la cadena de resultado. Volveremos para continuar con el siguiente.

     (2). Este símbolo y su siguiente se borrarán sin más.

     (3). Para el siguiente símbolo, volvemos al punto (1).

     El bucle anterior se cortará en el momento en que el símbolo encontrado sea #. En tal caso, lo borraremos, nos moveremos a la derecha (primera casilla del resultado) y terminaremos.
   - La MT que sigue este plan es la siguiente:



   - Nótese que si la entrada es λ (cinta con sólo blancos), el resultado de tomar un símbolo de cada 3 es también λ (cinta con sólo blancos).

2. (a) $L_1 = \{a^i b^j a^{i'} b^{j'} | i \leq i', j \leq j'\}$

   Este lenguaje *no es incontextual*. Lo probamos por el lema del bombeo para lenguajes incontextuales:

   - Fijamos una cierta n y tomamos la palabra $z = a^n b^n a^n b^n$. Se cumple $z \in L_1$ y $|z| \geq n$. Probemos que no es posible partir la palabra z en 5 pedazos $uvwxy$ de tal forma que se cumplan a la vez las tres condiciones siguientes:
     (1) $|uwx| \leq n$
     (2) $vx \neq \lambda$
     (3) $\forall i \geq 0 : uv^i wx^i y \in L_1$
   - Por (1), pueden darse los siguientes casos:
     (I) vwx tiene a's del 1º grupo y/o b's del 2º grupo.
     (II) vwx tiene b's del 2º grupo y/o a's del 3º grupo.
     (III) vwx tiene a's del 3º grupo y/o b's del 4º grupo.
   - Consideremos cada caso:
     (I) Por (2) tenemos que $uwy = a^{n-l} b^{n-k} a^n b^n$ con $l > 0$ ó $k > 0$. Distinguimos dos subcasos:
       - Si $l = 0$ entonces $k > 0$. En ese caso, en la palabra $uv^2wx^2y = a^n b^{n+k} a^n b^n$ hay más b's en el 2º grupo que en el 4º grupo $\Rightarrow uv^2wx^2y \notin L_1$. Por tanto, en este subcaso no es posible hacer cumplir las condiciones (1),(2) y (3) a la vez.
       - Si $l > 0$ entonces la palabra $uv^2wx^2y = a^{n+l} b^{n+k} a^n b^n$ tiene más a's en el 1º grupo que en el 3º grupo $\Rightarrow uv^2wx^2y \notin L_1$
     (II) Por (2) tenemos que $uwy = a^n b^{n-l} a^{n-k} b^n$ con $l > 0$ y $k > 0$. Entonces uwy tiene menos a's en la 3ª parte que en la 1ª $\Rightarrow uwy \notin L_1$ (es decir, $uv^0wx^0y \notin L_1$). También podríamos ver que la condición (3) no se cumple bombeando *de más*, pues $uv^2wx^2y \notin L_1$ (al tener $a^n b^{n+l} a^{n+k} b^n$ más b's en el 2º grupo que en el 4º grupo). Cualquiera de las dos opciones es válida en este caso.
     (III) Por (2) tenemos que $uwy = a^n b^n a^{n-l} b^{n-k}$ con $l > 0$ y $k > 0$. Distinguimos dos subcasos:
       - Si $l = 0$ entonces $k > 0$. En ese caso, la palabra uwy(es decir, $a^n b^n a^n b^{n-k}$) tiene más b's que el 2º grupo que en el 4º grupo $\Rightarrow uwy \notin L_1$.
       - Si $l > 0$ entonces la palabra $uwy = a^n b^n a^{n-l} b^{n-k}$ tiene más a's en el 1º grupo que en el 3º grupo $\Rightarrow uwy \notin L_1$.

     Por tanto, queda probado que no puede partirse la palabra z en 5 pedazos $uvwxy$ de forma que se cumplan las condiciones (1), (2) y (3).

   (b) $L_2 = \{a^i b^j c^k | i > j \text{ ó } i \leq k\}$

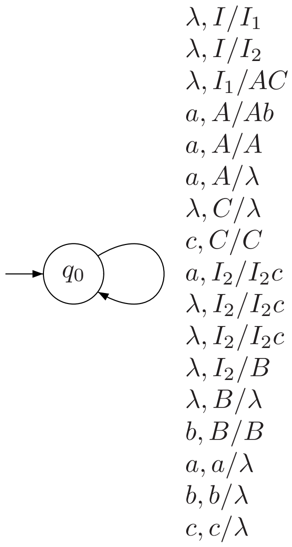
   L_2 es *incontextual*.

$$L_2 = \underbrace{\{a^i b^j c^k | i > j\}}_{I_1} \cup \underbrace{\{a^i b^j c^k | i \leq k\}}_{I_2}$$

   En base a la división anterior, proponemos la siguiente gramática incontextual:

$$\begin{array}{lcl} I & \rightarrow & I_1 \mid I_2 \\ I_1 & \rightarrow & AC \\ A & \rightarrow & aAb \mid aA \mid a \\ C & \rightarrow & \lambda \mid cC \\ I_2 & \rightarrow & aI_2C \mid I_2C \mid B \\ B & \rightarrow & \lambda \mid bB \end{array}$$

   El lenguaje L_2 es reconocido por el siguiente autómata con pila que reconoce por pila vacía, cuyo símbolo inicial es I:



3. (a) $LH_1 \neq LI$.

Como ejemplo de lenguaje que está en LI pero no está en LH_1, consideremos el lenguaje $\{a^n b^n a^m b^m | n, m \geq 0\}$.

- L está en LI:

$$\begin{array}{rcl} I & \rightarrow & AA \\ A & \rightarrow & aAb \mid \lambda \end{array}$$

- Sin embargo, L no está en LH_1. Sólo damos un motivo intuitivo: la parte $\{a^n b^n\}$ y la parte $\{a^m b^m\}$ necesitarán, en alguna regla de la gramática, una variable cada una (p.e., $I \rightarrow AA$. Si esta regla no existe, todos los terminales deberán surgir a los lados de una sola variable en reglas de la forma $X \rightarrow w_1 Y w_2$, siendo w_1 y w_2 cadenas de terminales. De esa forma podría crecer *sólo una de las dos partes*, nunca las dos a la vez. Dado que en la cadena desarrollada no podrá haber otra variable que desarrolle la otra parte, nunca podremos obtener todas las palabras del lenguaje L.

(b) $LH_2 = LI$.

Dada cualquier GI que no cumple los requisitos impuestos, siempre podremos crear su FNC, que representa el mismo lenguaje pero nunca utiliza más de dos variables en la parte derecha de una regla. (Para que represente *siempre* el mismo lenguaje, podríamos ignorar el paso de eliminación de producciones λ).

4.

$$\begin{array}{lcl} I & \rightarrow & aAB \\ A & \rightarrow & BAb \mid \lambda \mid B \\ B & \rightarrow & a \mid b \mid CD \\ C & \rightarrow & ba \\ D & \rightarrow & DD \end{array}$$

- Tras llevar a cabo todos los pasos necesarios para eliminar las producciones λ, obtenemos:

$$\begin{array}{lcl} I & \rightarrow & aAB \mid aB \\ A & \rightarrow & BAb \mid Bb \mid B \\ B & \rightarrow & a \mid b \mid CD \\ C & \rightarrow & ba \\ D & \rightarrow & DD \end{array}$$

- Tras hacer lo necesario para quitar las producciones unitarias tenemos:

$$\begin{array}{lcl} I & \rightarrow & aAB \mid aB \\ A & \rightarrow & BAb \mid Bb \mid a \mid b \mid CD \\ B & \rightarrow & a \mid b \mid CD \\ C & \rightarrow & ba \\ D & \rightarrow & DD \end{array}$$

- Tras eliminar los símbolos inútiles obtenemos:

$$\begin{array}{lcl} I & \rightarrow & aAB \mid aB \\ A & \rightarrow & BAb \mid Bb \mid a \mid b \\ B & \rightarrow & a \mid b \end{array}$$

- Tras hacer que en todas las partes derechas de las reglas con 2 símbolos o más sólo haya variables, obtenemos:

$$\begin{array}{lcl} I & \rightarrow & A'AB \mid A'B \\ A & \rightarrow & BAB' \mid BB' \mid a \mid b \\ B & \rightarrow & a \mid b \\ A' & \rightarrow & a \\ B' & \rightarrow & b \end{array}$$

- Finalmente, tras descomponer todas las reglas con 3 ó más variables en la parte derecha en reglas con 2 variables, obtenemos:

$$\begin{array}{lcl} I & \rightarrow & A'P \mid A'B \\ P & \rightarrow & AB \\ A & \rightarrow & BQ \mid BB' \mid a \mid b \\ Q & \rightarrow & AB' \\ B & \rightarrow & a \mid b \\ A' & \rightarrow & a \\ B' & \rightarrow & b \end{array}$$