eman ta zabal zazu

Universidad del País Vasco | Euskal Herriko Unibertsitatea

Lenguajes, Computación y Sistemas Inteligentes

Grado en Ingeniería Informática de Gestión y Sistemas de Información

Escuela de Ingeniería de Bilbao (UPV/EHU)

2^{o} curso

Curso académico 2023-2024

Tema 7: Autómatas finitos: Equivalencias

JOSÉ GAINTZARAIN IBARMIA

Departamento de Lenguajes y Sistemas Informáticos

Última actualización: 09 - 11 - 2023

Índice general

Índice de figuras

Índice de tablas

Tema 7

Autómatas finitos: Equivalencias

7.1. Introducción

En este tema hemos clasificado los autómatas finitos teniendo en cuenta algunas propiedades:

- Tener transiciones λ o no tenerlas.
- Ser determinista o no serlo.
- Ser completo o no serlo.
- Ser de opción binaria o no serlo.

Cada AF cumplirá algunas de esas propiedades.

A la hora de diseñar un autómata finito que defina un lenguaje, habitualmente la opción más cómoda es obtener un λ-AF porque los λ-AF hacen posible expresar las propiedades y estructuras de las palabras de manera más fácil, permitiendo un alto grado de modularidad y dando la opción de analizar propiedades disyuntivas por separado (por ejemplo, mediante el desarrollo de ramas distintas en árboles de configuraciones deterministas). Si consideramos los AF que no tienen transiciones λ, los AF no deterministas son los más cómodos para utilizar, teniendo en cuenta el grado de facilidad para expresar lenguajes, puesto que también permiten un cierto grado de modularidad y la posibilidad de analizar propiedades disyuntivas por separado (aunque dicha separación no es tan clara como en los λ-AF). Por último, como en los AF deterministas no es posible desarrollar árboles en los que distintas propiedades sean analizadas en distintas ramas y puesto que se han de controlar todas las propiedades mediante un único camino (o una única rama o secuencia de configuraciones deterministas), a veces resulta díficil diseñar un AF determinista para algunos lenguajes.

Como el grado de facilidad para expresar propiedades es diferente dependiendo de esas propiedades de los autómatas finitos, se puede pensar que tal vez su capacidad para definir o expresar lenguajes es también distinto. Dicho de otra forma, ¿es posible que lenguajes que no sean definibles mediante un AF sin transiciones λ sean definibles mediante un λ-AF? De la misma forma, ¿es posible que lenguajes que no son definibles mediante un AF determinista sean definibles mediante un AF no determinista?

La respuesta es no, en ambos casos. Esas características de los autómatas finitos no afectan a la capacidad de definir lenguajes. Por tanto, cualquier lenguaje definible mediante un λ-AF es definible mediante un AF sin transiciones λ. En la misma línea, cualquier lenguaje definible mediante un AF no determinista es definible mediante un AF determinista. La característica de opción binaria, formalizada mediante los β-AF, tampoco añade capacidad adicional, tampoco limita la capacidad. Ocurre lo mismo con la característica de ser completo o no serlo: ni añade ni quita capacidad.

Nos centraremos en la eliminación de las transiciones λ y en la eliminación del no determinismo. Por tanto, en primer lugar, se mostrará un método que, a partir de un λ-AF obtiene un AF sin transiciones λ. En segundo lugar, se mostrará un método que, a partir de un AF no determinista, obtiene un AF determinista.

7.2.

Sustitución de transiciones λ por transiciones no vacías

Cada transición λ (transición vacía) que aparezca en un AF puede ser simulada y, por tanto, sustituida, por una o más transiciones no vacías.

7.2.1 Descripción del AF sin transiciones λ correspondiente a un λ-AF

Dado un λ-AF $\mathcal{F}$ de la forma $(Q, \mathbb{A}, \tau_{\mathbb{A}}, \tau_\lambda, \sigma, Y)$ tal que $\tau_\lambda \neq \varnothing$, se obtendrá un AF $\mathcal{F}'$ de la forma $(Q', \mathbb{A}', \tau'_{\mathbb{A}'}, \tau'_\lambda, \sigma', Y')$ donde:

- $Q' = Q$
- $\mathbb{A}' = \mathbb{A}$
- $\tau'_\lambda = \varnothing$
- $\sigma' = \sigma$

Por tanto, en el nuevo AF $\mathcal{F}'$ no habrá transiciones vacías, los estados serán los mismos, el alfabeto también y el estado inicial también.

En cuanto a Y', será o bien igual a Y o bien igual a $Y \cup \{\sigma\}$. Para saber cuál de esas dos opciones es la adecuada, hay que tener en cuenta el siguiente criterio:

- Si en el AF original $\mathcal{F}$, partiendo desde el estado inicial σ es posible llegar a un estado de Y por medio de transiciones λ, entonces $Y' = Y \cup \{\sigma\}$.
- En caso contrario, $Y' = Y$.

Finalmente, el criterio para calcular elconjunto $\tau'_{\mathbb{A}'}$ es el siguiente:

- $\tau_{\mathbb{A}} \subseteq \tau'_{\mathbb{A}'}$
- Por cada $q_i \in Q$ y cada $\alpha \in \mathbb{A}$: si existe una computación de la forma $\langle(q_i, \alpha\varepsilon), \ldots, (q_h, \varepsilon)\rangle$, entonces, $(q_i, \alpha, q_h) \in \tau'_{\mathbb{A}'}$

Es decir, todas las transiciones de $\tau_{\mathbb{A}}$ van también en $\tau'_{\mathbb{A}'}$ y, además, por cada estado q_i perteneciente a Q y cada símbolo α perteneciente a $\mathbb{A}$, si desde q_i se puede llegar a q_h habiendo consumido la palabra $\alpha\varepsilon$, entonces habrá una transición de la forma (q_i, α, q_h) en $\tau'_{\mathbb{A}'}$.

7.2.2 Construcción del AF sin transiciones λ correspondiente a un λ-AF

Puesto que Q', $\mathbb{A}'$ y σ' son Q, $\mathbb{A}$ y σ respectivamente y el conjunto τ'_λ es vacío, la construcción del nuevo AF se centra en el cálculo de $\tau'_{\mathbb{A}'}$ y el cálculo de Y'.

7.2.2.1 Cálculo de Y'

Si pretendemos calcular Y' a mano, podemos valernos del diagrama de transiciones y averiguar si es posible llegar desde σ hasta un estado de Y atravesando solo transiciones λ.

Si se quiere automatizar el cálculo (por ejemplo, mediante un programa) una opción es la de ir costruyendo el árbol de computaciones para la configuración (σ, ε). Si en ese árbol surge alguna configuración de la forma (ϱ, ε) donde $\varrho \in Y$, entonces $Y' = Y \cup \{\sigma\}$. En cambio, si se ha construido el árbol de computaciones completo para la configuración (σ, ε) y no ha surgido ninguna configuración de la forma (ϱ, ε) donde $\varrho \in Y$, entonces $Y' = Y$.

7.2.2.2 Cálculo de $\tau'_{\mathbb{A}'}$

Una manera sistemática de calcular $\tau'_{\mathbb{A}'}$ es la de construir, para cada $q_i \in Q$ y cada $\alpha \in \mathbb{A}$, el árbol de computaciones para la configuración (q_i, α). Una vez que se tenga el árbol completo para (q_i, α), habrá una transición (q_i, α, q_h) para cada configuración de la forma (q_h, ε) que aparezca en el árbol. Si no aparece ninguna configuración de la forma (q_h, ε), entonces desde q_i no habrá ninguna transición con el símbolo α.

7.2.3 Coste computacional de la construcción del AF sin transiciones λ correspondiente a un λ-AF

Puesto que el nuevo AF sin transiciones λ tiene el mismo número de estados que el λ-AF inicial, el coste es polinómico y, por tanto, aceptable.

7.2.4 Ejemplos de construcción del AF sin transiciones λ correspondiente a un λ-AF

7.2.4.1 Ejemplo 1

En la figura 7.2.1 de la página 21, se muestra un AF $(Q, \mathbb{A}, \tau_{\mathbb{A}}, \tau_{\lambda}, \sigma, Y)$ que contiene dos transiciones λ. En ese AF, los componentes son los siguientes:

- $Q = \{q_0, q_1, q_2\}$
- $\mathbb{A} = \{a, b, c\}$
- $\tau_{\mathbb{A}} = \{(q_0, a, q_0), (q_1, b, q_1), (q_2, a, q_2)\}$
- $\tau_{\lambda} = \{(q_0, q_1), (q_1, q_2)\}$
- $\sigma = q_0$
- $Y = \{q_2\}$

El objetivo es construir un AF $(Q', \mathbb{A}', \tau'_{\mathbb{A}'}, \tau'_{\lambda}, \sigma', Y')$ que defina el mismo lenguaje pero en el que τ'_{λ} sea $\varnothing$.

Sabemos que se cumplirá $Q' = Q$, $\mathbb{A}' = \mathbb{A}$ y $\sigma' = \sigma$. Quedan por determinar Y' y $\tau'_{\mathbb{A}'}$.

En el árbol de computaciones correspondiente a (q_0, ε) —mostrado en la figura 7.2.2 de la página 21— aparece una configuración cuyo primer componente es un estado que pertenece a Y: (q_2, ε). Esa configuración está marcada en amarillo. Debido a la aparición de esa configuración, se sabe que en el nuevo AF el estado inicial ha de pertenecer a Y', es decir, ha de tener dos círculos. Consecuentemente, $Y' = Y \cup \{q_0\}$.

En la figura 7.2.3 de la página 21 se muestran los árboles para las configuraciones $(q_0, a\varepsilon)$, $(q_0, b\varepsilon)$ y $(q_0, c\varepsilon)$. En cada árbol se han marcado en amarillo las configuraciones que dan lugar a una transición desde q_0. De esta manera, habrá transiciones desde q_0 a q_0, q_1 y q_2 con a; habrá transiciones desde q_0 a q_1 y q_2 con b; y no habrá ninguna transición desde q_0 con c.

En la figura 7.2.4 de la página 21 se muestran los árboles para las configuraciones $(q_1, a\varepsilon)$, $(q_1, b\varepsilon)$ y $(q_1, c\varepsilon)$. En cada árbol se han marcado en amarillo las configuraciones que dan lugar a una transición desde q_1. De esta manera, habrá transiciones desde q_1 a q_2 con a; habrá transiciones desde q_1 a q_1 y q_2 con b; y no habrá ninguna transición desde q_1 con c.

En la figura 7.2.5 de la página 23 se muestran los árboles para las configuraciones $(q_2, a\varepsilon)$, $(q_2, b\varepsilon)$ y $(q_2, c\varepsilon)$. En cada árbol se han marcado en amarillo las configuraciones que dan lugar a una transición desde q_2. De esta manera, habrá transición desde q_2 a q_2 con a; pero no habrá ninguna transición desde q_2 ni con b ni con c.

En la figura 7.2.6 de la página 23 se muestran el AF obtenido. Este AF es equivalente al AF de la figura 7.2.1 de la página 21 pero no tiene transiciones λ.

7.2.4.2 Ejemplo 2

En la figura 7.2.7 de la página 23, se muestra un AF $(R, \mathbb{A}, \tau_{\mathbb{A}}, \tau_{\lambda}, \sigma, Y)$ que contiene dos transiciones λ. En ese AF, los componentes son los siguientes:

- $R = \{r_0, r_1, r_2\}$
- $\mathbb{A} = \{a, b, c\}$
- $\tau_{\mathbb{A}} = \{(r_0, a, r_0), (r_0, a, r_1), (r_1, b, r_1), (r_2, a, r_2)\}$
- $\tau_{\lambda} = \{(r_1, r_2)\}$
- $\sigma = r_0$
- $Y = \{r_2\}$

El objetivo es construir un AF $(R', \mathbb{A}', \tau'_{\mathbb{A}'}, \tau'_{\lambda}, \sigma', Y')$ que defina el mismo lenguaje pero en el que τ'_{λ} sea $\varnothing$.

Sabemos que se cumplirá $R' = R$, $\mathbb{A}' = \mathbb{A}$ y $\sigma' = \sigma$. Quedan por determinar Y' y $\tau'_{\mathbb{A}'}$.

En el árbol de computaciones correspondiente a (r_0, ε) —mostrado en la figura 7.2.8 de la página 23— no aparece ninguna configuración cuyo primer componente sea un estado que pertenece a Y. Debido a ello, se sabe que en el nuevo AF el estado inicial no ha de pertenecer a Y', es decir, ha de tener solo un círculo. Consecuentemente, $Y' = Y$.

En la figura 7.2.9 de la página 23 se muestran los árboles para las configuraciones $(r_0, a\varepsilon)$, $(r_0, b\varepsilon)$ y $(r_0, c\varepsilon)$. En cada árbol se han marcado en amarillo las configuraciones que dan lugar a una transición desde r_0. De esta manera, habrá transiciones desde r_0 a r_0, r_1 y r_2 con a; y no habrá ninguna transición desde r_0 ni con b ni con c.

En la figura 7.2.10 de la página 24 se muestran los árboles para las configuraciones $(r_1, a\varepsilon)$, $(r_1, b\varepsilon)$ y $(r_1, c\varepsilon)$. En cada árbol se han marcado en amarillo las configuraciones que dan lugar a una transición desde r_1. De esta manera, habrá una transición desde r_1 a r_2 con a; habrá transiciones desde r_1 a r_1 y r_2 con b; y no habrá ninguna transición desde r_1 con c.

En la figura 7.2.11 de la página 24 se muestran los árboles para las configuraciones $(r_2, a\varepsilon)$, $(r_2, b\varepsilon)$ y $(r_2, c\varepsilon)$. En cada árbol se han marcado en amarillo las configuraciones que dan lugar a una transición desde r_2. De esta manera, habrá transición desde r_2 a r_2 con a; pero no habrá ninguna transición desde r_2 ni con b ni con c.

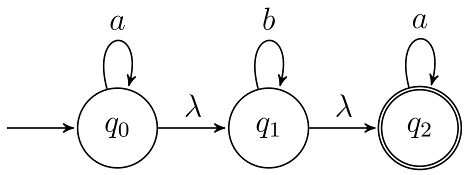


Figura 7.2.1. Sustitución de transiciones λ: ejemplo 1.

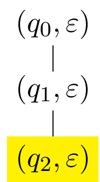


Figura 7.2.2. Sustitución de transiciones λ: árbol de computaciones correspondiente a la configuración (q_0, ε) en el AF de la figura 7.2.1 (página 21).

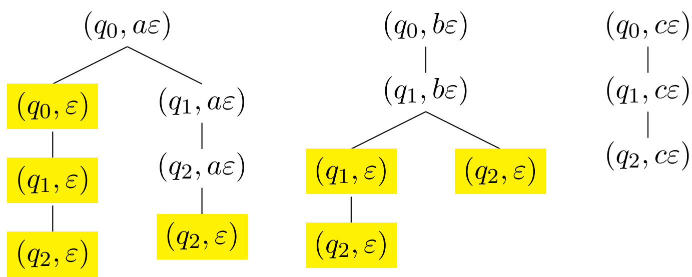


Figura 7.2.3. Sustitución de transiciones λ: árboles de computaciones correspondientes a las configuraciones en las que el primer componente es q_0. AF de la figura 7.2.1 (página 21).

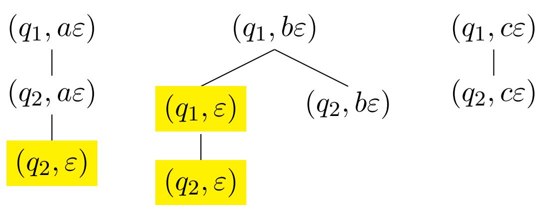


Figura 7.2.4. Sustitución de transiciones λ: árboles de computaciones correspondientes a las configuraciones en las que el primer componente es q_1. AF de la figura 7.2.1 (página 21).

En la figura 7.2.12 de la página 24 se muestran el AF obtenido. Este AF es equivalente al AF de la figura 7.2.7 de la página 23 pero no tiene transiciones λ.

7.2.4.3 Ejemplo 3

En la figura 7.2.13 de la página 26, se muestra un AF $(S, \mathbb{A}, \tau_{\mathbb{A}}, \tau_{\lambda}, \sigma, Y)$ que contiene dos transiciones λ. En ese AF, los componentes son los siguientes:

- $S = \{s_0, s_1, s_2\}$
- $\mathbb{A} = \{a, b, c\}$
- $\tau_{\mathbb{A}} = \{(s_2, a, s_3), (s_2, a, s_4), (s_3, a, s_3), (s_4, c, s_4)\}$
- $\tau_{\lambda} = \{(s_0, s_1), (s_1, s_2)\}$
- $\sigma = s_0$
- $Y = \{s_1, s_3, s_4\}$

El objetivo es construir un AF $(S', \mathbb{A}', \tau'_{\mathbb{A}'}, \tau'_{\lambda}, \sigma', Y')$ que defina el mismo lenguaje pero en el que τ'_{λ} sea $\varnothing$.

Sabemos que se cumplirá $S' = S$, $\mathbb{A}' = \mathbb{A}$ y $\sigma' = \sigma$. Quedan por determinar Y' y $\tau'_{\mathbb{A}'}$.

En el árbol de computaciones correspondiente a (s_0, ε) —mostrado en la figura 7.2.14 de la página 26— no aparece ninguna configuración cuyo primer componente sea un estado que pertenece a Y. Debido a ello, se sabe que en el nuevo AF el estado inicial no ha de pertenecer a Y', es decir, ha de tener solo un círculo. Consecuentemente, $Y' = Y$.

En la figura 7.2.15 de la página 26 se muestran los árboles para las configuraciones $(s_0, a\varepsilon)$, $(s_0, b\varepsilon)$ y $(s_0, c\varepsilon)$. En cada árbol se han marcado en amarillo las configuraciones que dan lugar a una transición desde s_0. De esta manera, habrá una transición desde s_0 a s_3 con a, otra transición desde s_0 a s_4 con c y no habrá ninguna transición desde s_0 con b.

En la figura 7.2.16 de la página 26 se muestran los árboles para las configuraciones $(s_1, a\varepsilon)$, $(s_1, b\varepsilon)$ y $(s_1, c\varepsilon)$. En cada árbol se han marcado en amarillo las configuraciones que dan lugar a una transición desde s_1. De esta manera, habrá una transición desde s_1 a s_3 con a y habrá una transición desde s_1 a s_4 con c; pero no habrá ninguna transición desde s_1 con b.

En la figura 7.2.17 de la página 27 se muestran los árboles para las configuraciones $(s_2, a\varepsilon)$, $(s_2, b\varepsilon)$ y $(s_2, c\varepsilon)$. En cada árbol se han marcado en amarillo las configuraciones que dan lugar a una transición desde s_2. De esta manera, habrá una transición desde s_2 a s_3 con a y habrá una transición desde s_2 a s_4 con c; pero no habrá ninguna transición desde s_2 con b.

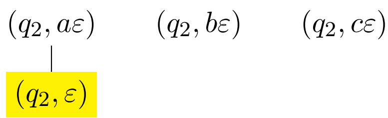


Figura 7.2.5. Sustitución de transiciones λ: árboles de computaciones correspondientes a las configuraciones en las que el primer componente es q_2. AF de la figura 7.2.1 (página 21).

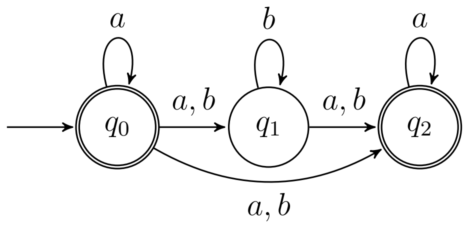


Figura 7.2.6. Sustitución de transiciones λ: ejemplo 1. AF resultante

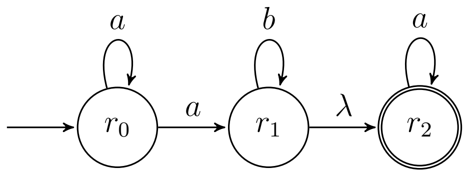


Figura 7.2.7. Sustitución de transiciones λ: ejemplo 2.

$$(r_0, \varepsilon)$$

Figura 7.2.8. Sustitución de transiciones λ: árbol de computaciones correspondiente a la configuración (r_0, ε) en el AF de la figura 7.2.7 (página 23).

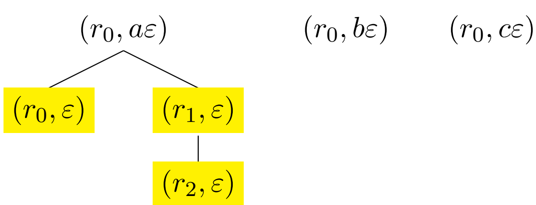


Figura 7.2.9. Sustitución de transiciones λ: árboles de computaciones correspondientes a las configuraciones en las que el primer componente es r_0. AF de la figura 7.2.7 (página 23).

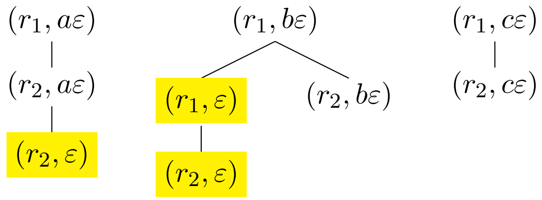


Figura 7.2.10. Sustitución de transiciones λ: árboles de computaciones correspondientes a las configuraciones en las que el primer componente es r_1. AF de la figura 7.2.7 (página 23).

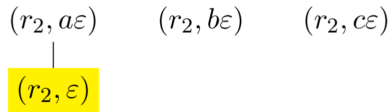


Figura 7.2.11. Sustitución de transiciones λ: árboles de computaciones correspondientes a las configuraciones en las que el primer componente es r_2. AF de la figura 7.2.7 (página 23).

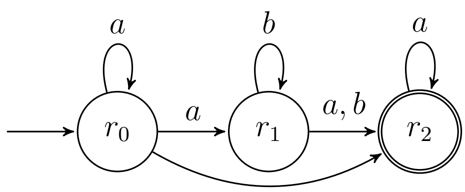


Figura 7.2.12. Sustitución de transiciones λ: ejemplo 2. AF resultante. AF de la figura 7.2.7 (página 23)

En la figura 7.2.18 de la página 27 se muestran los árboles para las configuraciones $(s_3, a\varepsilon)$, $(s_3, b\varepsilon)$ y $(s_3, c\varepsilon)$. En cada árbol se han marcado en amarillo las configuraciones que dan lugar a una transición desde s_3. De esta manera, habrá una transición desde s_3 a s_3 con a y no habrá ninguna transición desde s_3 ni con b ni con c.

Por último, en la figura 7.2.19 de la página 27 se muestran los árboles para las configuraciones $(s_4, a\varepsilon)$, $(s_4, b\varepsilon)$ y $(s_4, c\varepsilon)$. En cada árbol se han marcado en amarillo las configuraciones que dan lugar a una transición desde s_4. De esta manera, habrá una transición desde s_4 a s_4 con c y no habrá ninguna transición desde s_4 ni con a ni con b.

En la figura 7.2.20 de la página 27 se muestra el AF obtenido. Este AF es equivalente al AF de la figura 7.2.13 de la página 26 pero no tiene transiciones λ.

Nótese que en el AF presentado en la figura 7.2.20 de la página 27, los estados s_1 y s_2 son inalcanzables desde el estado inicial s_0. Es decir, no hay manera de llegar desde s_0 a los estados s_1 y s_2, ni directamente ni pasando por otros estados. Recordemos que nosotros utilizamos los AF para averiguar si una palabra pertenece a un lenguaje concreto y para realizar esa averiguación siempre hay que partir desde el estado s_0. Por tanto, podríamos simplificar el AF eliminando los estados s_1 y s_2. En la figura 7.2.21 de la página 28 se muestra el AF simplificado.

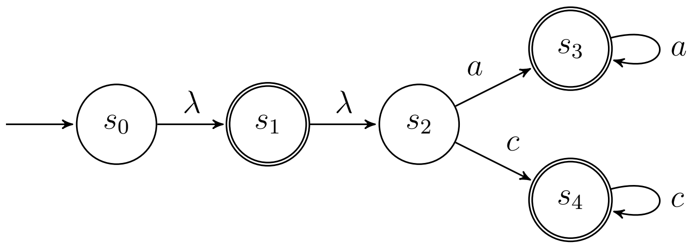


Figura 7.2.13. Sustitución de transiciones λ: ejemplo 3.

$$
\begin{array}{c}
(s_0, \varepsilon) \\
| \\
\boxed{(s_1, \varepsilon)} \\
| \\
(s_2, \varepsilon)
\end{array}
$$

Figura 7.2.14. Sustitución de transiciones λ: árbol de computaciones correspondiente a la configuración (s_0, ε) en el AF de la figura 7.2.13 (página 26).

$$
\begin{array}{ccccc}
(s_0, a\varepsilon) & & (s_0, b\varepsilon) & & (s_0, c\varepsilon) \\
| & & | & & | \\
(s_1, a\varepsilon) & & (s_1, b\varepsilon) & & (s_1, c\varepsilon) \\
| & & | & & | \\
(s_2, a\varepsilon) & & (s_2, b\varepsilon) & & (s_2, c\varepsilon) \\
| & & & & | \\
\boxed{(s_3, \varepsilon)} & & & & \boxed{(s_4, \varepsilon)}
\end{array}
$$

Figura 7.2.15. Sustitución de transiciones λ: árboles de computaciones correspondientes a las configuraciones en las que el primer componente es s_0. AF de la figura 7.2.13 (página 26).

$$
\begin{array}{ccccc}
(s_1, a\varepsilon) & & (s_1, b\varepsilon) & & (s_1, c\varepsilon) \\
| & & | & & | \\
(s_2, a\varepsilon) & & (s_2, b\varepsilon) & & (s_2, c\varepsilon) \\
| & & & & | \\
\boxed{(s_3, \varepsilon)} & & & & \boxed{(s_4, \varepsilon)}
\end{array}
$$

Figura 7.2.16. Sustitución de transiciones λ: árboles de computaciones correspondientes a las configuraciones en las que el primer componente es s_1. AF de la figura 7.2.13 (página 26).

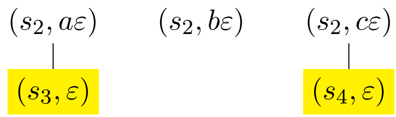


Figura 7.2.17. Sustitución de transiciones λ: árboles de computaciones correspondientes a las configuraciones en las que el primer componente es s_2. AF de la figura 7.2.13 (página 26).

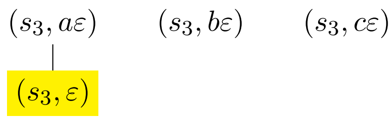


Figura 7.2.18. Sustitución de transiciones λ: árboles de computaciones correspondientes a las configuraciones en las que el primer componente es s_3. AF de la figura 7.2.13 (página 26).

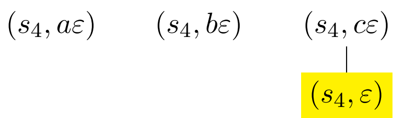


Figura 7.2.19. Sustitución de transiciones λ: árboles de computaciones correspondientes a las configuraciones en las que el primer componente es s_4. AF de la figura 7.2.13 (página 26).

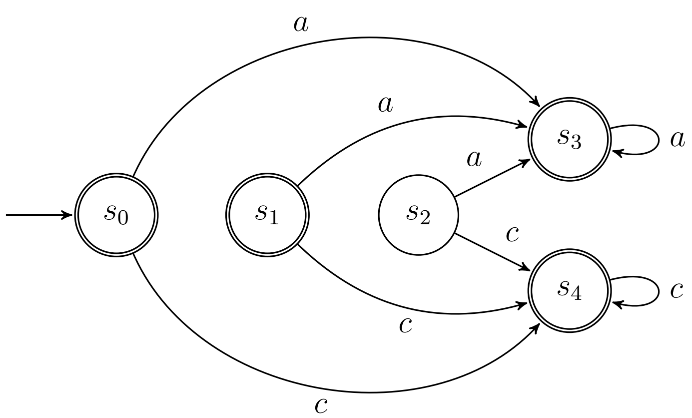


Figura 7.2.20. Sustitución de transiciones λ: ejemplo 3. AF resultante. AF de la figura 7.2.13 (página 26)

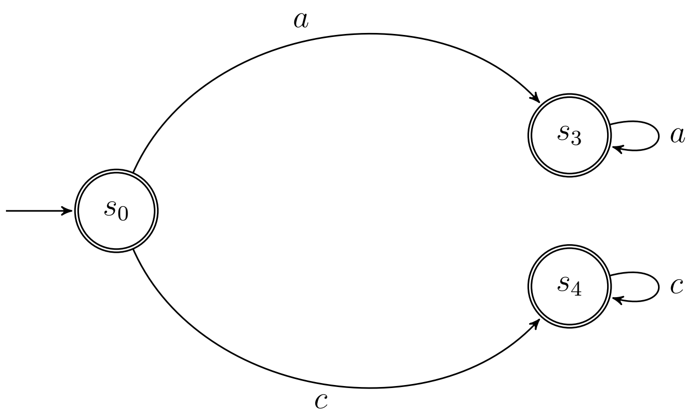


Figura 7.2.21. Sustitución de transiciones λ: ejemplo 3. AF resultante simplificado. AF de la figura 7.2.13 (página 26)

7.3. Transformación de no determinista a determinista y completo

En este apartado, se presenta un método para obtener un AF determinista completo (AFDC) sin transiciones λ a partir de un AF no determinista sin transiciones λ. El hecho de considerar que el AF no determinista que se nos da para ser transformado no tiene transiciones λ, no supone una pérdida de generalidad. Si se quiere obtener un AF determinista sin transiciones λ a partir de un AF $\mathcal{F}$ que contiene transiciones λ, primero aplicaremos el método presentado en el apartado anterior y obtendremos un AF $\mathcal{F}'$ sin transiciones λ. Si $\mathcal{F}'$ no es determinista, se aplicará a $\mathcal{F}'$ el método que se presenta en este apartado para construir un AF $\mathcal{F}''$ determinista. El AF $\mathcal{F}''$ no tendrá transiciones λ.

En el resto de este apartado, todo AF será sin transiciones λ.

7.3.1 Descripción del AFDC sin transiciones λ correspondiente a un AFND sin transiciones λ

Dado un AFND $\mathcal{F}$ de la forma $(Q, \mathbb{A}, \tau_{\mathbb{A}}, \tau_\lambda, \sigma, Y)$ tal que $\tau_\lambda = \varnothing$, se obtendrá un AFDC $\mathcal{F}'$ de la forma $(Q', \mathbb{A}', \tau'_{\mathbb{A}'}, \tau'_\lambda, \sigma', Y')$ donde:

- $Q' \subseteq 2^Q$
- $\mathbb{A}' = \mathbb{A}$
- $\tau'_\lambda = \varnothing$
- $\sigma' = \{\sigma\}$

Por tanto, en el nuevo AF $\mathcal{F}'$ no habrá transiciones vacías, el alfabeto será el mismo y el estado inicial será el conjunto unitario formado por σ.

El conjunto formado por todos los subconjuntos de Q lo representaremos como 2^Q. El conjunto de estados Q' estará formado por algunos subconjuntos de Q. En general, no todos los subconjuntos de Q estarán en Q'. Solo han de estar en Q' el conjunto unitario $\{\sigma\}$ y aquellos subconjuntos de Q que son alcanzables desde $\{\sigma\}$ en uno o más pasos. Por tanto, por lo menos estará $\{\sigma\}$.

El conjunto Y', estará formado por aquellos elementos de Q' que contengan al menos un elemento de Y.

$$Y' = \{\varrho \mid (\varrho \in Q') \wedge (\varrho \cap Y \neq \varnothing)\}$$

Nótese que decir $\varrho \in Q'$ es equivalente a decir $\varrho \subseteq Q$. [1]

En cuanto a $\tau'_{\mathbb{A}'}$, lo primero es tener en cuenta que por cada $\varrho \in Q'$ y cada $\alpha \in \mathbb{A}$, tiene que existir una transición de la forma $(\varrho, \alpha, \varrho')$ donde ϱ' es un elemento de Q'. Además, solo puede haber una única transición con ϱ y α como primeros dos elementos. Recordemos que se quiere diseñar un AFDC.

El elemento ϱ' correspondiente a un estado ϱ y un símbolo α es el siguiente:

$$\varrho' = \{q_h \mid (q_h \in Q) \wedge \exists q_i (q_i \in \varrho \wedge (q_i, \alpha, q_h) \in \tau_{\mathbb{A}})\}$$

Es decir, ϱ' estará formado por todos aquellos estados que son accesibles en un solo paso con α desde algún estado de ϱ.

7.3.2 Construcción del AFDC sin transiciones λ correspondiente a un AFND sin transiciones λ

Puesto que $\mathbb{A}'$ es $\mathbb{A}$, el conjunto τ'_λ es vacío y σ' es el conjunto unitario $\{\sigma\}$, la construcción del nuevo AFDC se centra en el cálculo de $\tau'_{\mathbb{A}'}$, Q' eY'.

7.3.2.1 Cálculo de $\tau'_{\mathbb{A}'}$ y Q'

Se calcularán $\tau'_{\mathbb{A}'}$ y Q' de manera progresiva.

- **Inicialización de Q' y de $\tau'_{\mathbb{A}'}$:**
  Para empezar, sabemos que el estado inicial σ' es $\{\sigma\}$ y que $\{\sigma\}$ ha de pertenecer a Q'. Por tanto, inicializamos Q' como el conjunto de estados formado por el estado inicial $\{\sigma\}$.

  Por otra parte, inicializamos $\tau'_{\mathbb{A}'}$ como conjunto vacío.

[1] ϱ es la letra griega *qoppa*, que ya no está en el alfabeto griego moderno.

- **Actualización de Q' y de $\tau'_{\mathbb{A}'}$:**
  Si para algún estado ϱ de Q' no se han calculado las transiciones correspondientes (una por cada símbolo del alfabeto), se han de calcular dichas transiciones. Para cada símbolo α de $\mathbb{A}$ se ha de calcular el conjunto ϱ' de estados a los que se puede llegar con α desde los estado de ϱ. Por una parte, se añade la transición $(\varrho, \alpha, \varrho')$ a $\tau'_{\mathbb{A}'}$. Por otra parte, si ϱ' no está de antes en Q', se añade a Q'.

  El proceso de actualización se ha de repetir hasta que se cumpla la condición para finalizar el proceso.

- **Finalización del cálculo de Q' y de $\tau'_{\mathbb{A}'}$:**
  Si para todo estado ϱ de Q' ya se han calculado las transiciones correspondientes, se termina el proceso de cálculo de Q' y de $\tau'_{\mathbb{A}'}$.

En general, el conjunto Q' final no tendrá todos los elementos de $\mathbb{2}^Q$. Los elementos de $\mathbb{2}^Q$ que no aparecen en Q' son estados inaccesibles desde el estado inicial $\sigma' = \{\sigma\}$.

7.3.2.2 Cálculo de Y'

Una vez calculado el conjunto Q' final, el conjunto Y' estará formado por los elementos de Q' que contengan algún elemento de Y.

7.3.3 Coste computacional de la construcción del AFDC correspondiente a un AFND

Si el AFND inicial tiene n estados, el nuevo AFDC puede llegar a tener 2^n estados. Por tanto, el coste es exponencial y esto quiere decir que el método es muy costoso desde el punto de vista computacional. Pero no se conoce ningún método que sea menos costoso.

7.3.4 Ejemplos de construcción del AFDC sin transiciones λ correspondiente a un AFND sin transiciones λ

7.3.4.1 Ejemplo 1 AFND – AFDC

7.3.4.1.1 Ejemplo 1: AFND original

En la figura 7.3.1 de la página 33, se muestra un AF $(Q, \mathbb{A}, \tau_{\mathbb{A}}, \tau_{\lambda}, \sigma, Y)$ que no es determinista: en q_0 y q_2 para a hay más de una opción y en q_0 para b y c hay más de una opción.

En ese AF, los componentes son los siguientes:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\mathbb{A} = \{a, b, c\}$
- $\tau_{\mathbb{A}} = \{(q_0, a, q_1), (q_0, a, q_2), (q_0, b, q_0), (q_0, b, q_2), (q_0, c, q_0), (q_0, c, q_2), (q_1, b, q_1), (q_1, c, q_1), (q_2, a, q_2), (q_2, a, q_3), (q_2, b, q_2), (q_2, c, q_2)\}$
- $\tau_\lambda = \varnothing$
- $\sigma = q_0$
- $Y = \{q_1, q_3\}$

7.3.4.1.2 Ejemplo 1: Pasos para construir un AFDC a partir del AFND original

El objetivo es construir un AF $(Q', \mathbb{A}', \tau'_{\mathbb{A}'}, \tau'_\lambda, \sigma', Y')$ que defina el mismo lenguaje pero que sea determinista y completo y en el que el conjunto τ'_λ sea vacío.

Para empezar, sabemos que $\mathbb{A}' = \mathbb{A}$, $\tau'_\lambda = \varnothing$ y $\sigma' = \{\sigma\} = \{q_0\}$. Quedan por determinar Q', $\tau'_{\mathbb{A}'}$ y Y'.

- **Paso 1:**
  Sabemos que el estado inicial σ' es $\{\sigma\}$ —es decir, $\{q_0\}$— y que $\{q_0\}$ pertenece a Q'. En la figura 7.3.2 de la página 33, se muestra este punto de partida.

- **Paso 2:**
  A continuación, se calculan las transiciones correspondientes a $\{q_0\}$. En este caso, solo tenemos que considerar el estado q_0:

  - Con a, desde q_0 se llega a $\{q_1, q_2\}$
  - Con b, desde q_0 se llega a $\{q_0, q_2\}$
  - Con c, desde q_0 se llega a $\{q_0, q_2\}$

  Por tanto, se han de añadir esos dos estados nuevos —$\{q_1, q_2\}$ y $\{q_0, q_2\}$— y las transiciones desde $\{q_0\}$ a esos dos estados.

  En la figura 7.3.3 de la página 33, se muestra lo obtenido en el paso 2.

- **Paso 3:**
  A continuación, se calculan las transiciones correspondientes a $\{q_1, q_2\}$. En este caso, tenemos que considerar los estados q_1 y q_2:

  - Con a, desde q_1 se llega a $\varnothing$ y desde q_2 se llega a $\{q_2, q_3\}$. Juntando los dos casos, queda $\varnothing \cup \{q_2, q_3\} = \{q_2, q_3\}$.

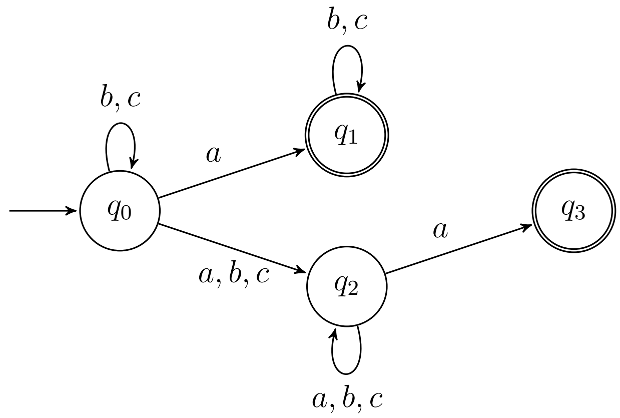


Figura 7.3.1. De AFND a AFDC: ejemplo 1. AFND de partida.

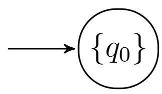


Figura 7.3.2. De AFND a AFD: ejemplo 1, paso 1.

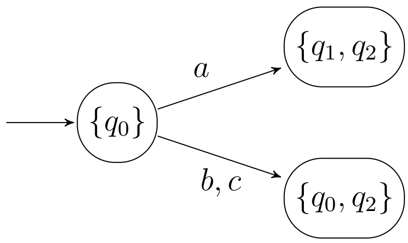


Figura 7.3.3. De AFND a AFD: ejemplo 1, paso 2.

- Con b, desde q_1 se llega a $\{q_1\}$ y desde q_2 se llega a $\{q_2\}$. Juntando los dos casos, queda $\{q_1\} \cup \{q_2\} = \{q_1, q_2\}$.
- Con c, desde q_1 se llega a $\{q_1\}$ y desde q_2 se llega a $\{q_2\}$. Juntando los dos casos, queda $\{q_1\} \cup \{q_2\} = \{q_1, q_2\}$.

Por tanto, se ha de añadir el estado nuevo $\{q_2, q_3\}$ y las transiciones desde $\{q_1, q_2\}$ que se han calculado.

En la figura 7.3.4 de la página 35, se muestra lo obtenido en el paso 3.

- **Paso 4:**
  A continuación, se calculan las transiciones correspondientes a $\{q_0, q_2\}$. En este caso, tenemos que considerar los estados q_0 y q_2:

  - Con a, desde q_0 se llega a $\{q_1, q_2\}$ y desde q_2 se llega a $\{q_2, q_3\}$. Juntando los dos casos, queda $\{q_1, q_2\} \cup \{q_2, q_3\} = \{q_1, q_2, q_3\}$.
  - Con b, desde q_0 se llega a $\{q_0, q_2\}$ y desde q_2 se llega a $\{q_2\}$. Juntando los dos casos, queda $\{q_0, q_2\} \cup \{q_2\} = \{q_0, q_2\}$.
  - Con c, desde q_0 se llega a $\{q_0, q_2\}$ y desde q_2 se llega a $\{q_2\}$. Juntando los dos casos, queda $\{q_0, q_2\} \cup \{q_2\} = \{q_0, q_2\}$.

  Por tanto, se ha de añadir el estado $\{q_1, q_2, q_3\}$ y las transiciones desde $\{q_0, q_2\}$ que se han calculado.

  En la figura 7.3.5 de la página 35, se muestra lo obtenido en el paso 4.

- **Paso 5:**
  A continuación, se calculan las transiciones correspondientes a $\{q_2, q_3\}$. En este caso, tenemos que considerar los estados q_2 y q_3:

  - Con a, desde q_2 se llega a $\{q_2, q_3\}$ y desde q_3 se llega a $\varnothing$. Juntando los dos casos, queda $\{q_2, q_3\} \cup \varnothing = \{q_2, q_3\}$.
  - Con b, desde q_2 se llega a $\{q_2\}$ y desde q_3 se llega a $\varnothing$. Juntando los dos casos, queda $\{q_2\} \cup \varnothing = \{q_2\}$.
  - Con c, desde q_2 se llega a $\{q_2\}$ y desde q_3 se llega a $\varnothing$. Juntando los dos casos, queda $\{q_2\} \cup \varnothing = \{q_2\}$.

  Por tanto, se ha de añadir el estado nuevo $\{q_2\}$ y las transiciones desde $\{q_2, q_3\}$ que se han calculado.

  En la figura 7.3.6 de la página 35, se muestra lo obtenido en el paso 5.

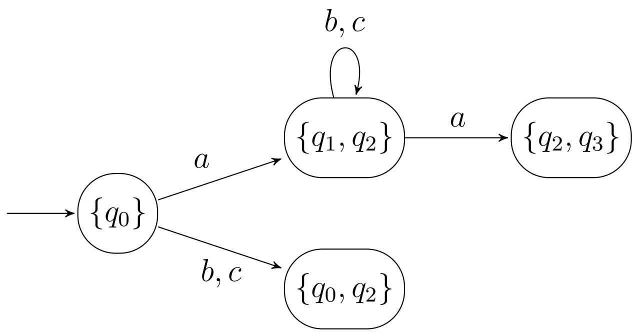


Figura 7.3.4. De AFND a AFD: ejemplo 1, paso 3.

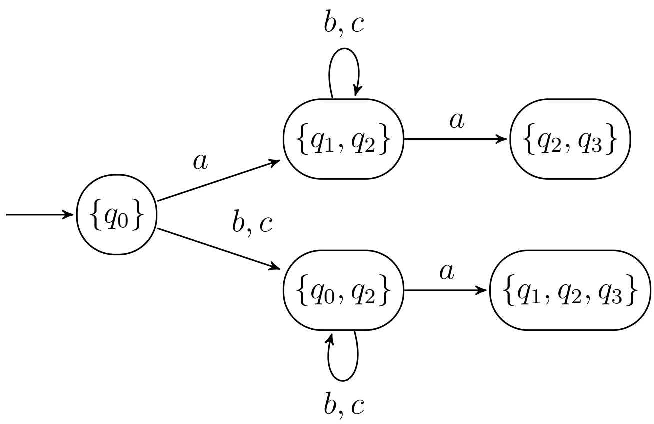


Figura 7.3.5. De AFND a AFD: ejemplo 1, paso 4.

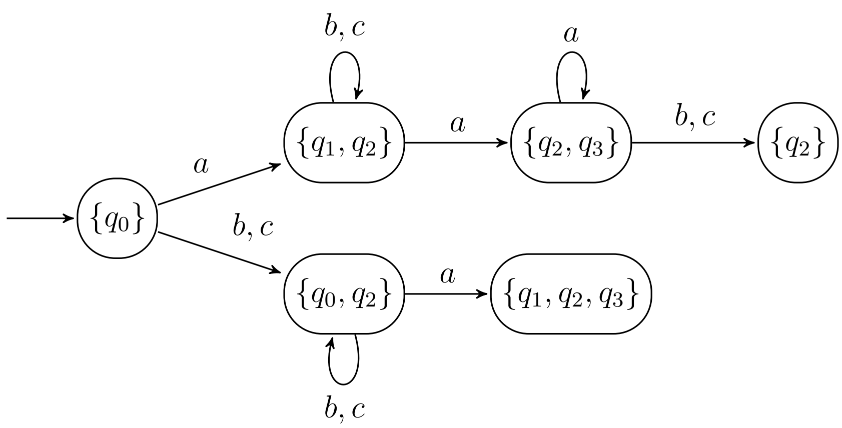


Figura 7.3.6. De AFND a AFD: ejemplo 1, paso 5.

- **Paso 6:**
A continuación, se calculan las transiciones correspondientes a $\{q_1, q_2, q_3\}$. En este caso, tenemos que considerar los estados q_1, q_2 y q_3:

  - Con a, desde q_1 se llega a $\varnothing$, desde q_2 se llega a $\{q_2, q_3\}$ y desde q_3 se llega a $\varnothing$. Juntando los tres casos, queda $\varnothing \cup \{q_2, q_3\} \cup \varnothing = \{q_2, q_3\}$.
  - Con b, desde q_1 se llega a $\{q_1\}$, desde q_2 se llega a $\{q_2\}$ y desde q_3 se llega a $\varnothing$. Juntando los tres casos, queda $\{q_1\} \cup \{q_2\} \cup \varnothing = \{q_1, q_2\}$.
  - Con c, desde q_1 se llega a $\{q_1\}$, desde q_2 se llega a $\{q_2\}$ y desde q_3 se llega a $\varnothing$. Juntando los tres casos, queda $\{q_1\} \cup \{q_2\} \cup \varnothing = \{q_1, q_2\}$.

  Por tanto, no hay que añadir ningún estado; solo las transiciones desde $\{q_1, q_2, q_3\}$ que se han calculado.

  En la figura 7.3.7 de la página 37, se muestra lo obtenido en el paso 6.

- **Paso 7:**
A continuación, se calculan las transiciones correspondientes a $\{q_2\}$. En este caso, tenemos que considerar solo el estado q_2:

  - Con a, desde q_2 se llega a $\{q_2, q_3\}$.
  - Con b, desde q_2 se llega a $\{q_2\}$.
  - Con c, desde q_2 se llega a $\{q_2\}$.

  Por tanto, no hay que añadir ningún estado; solo las transiciones desde $\{q_2\}$ que se han calculado.

  En la figura 7.3.8 de la página 37, se muestra lo obtenido en el paso 7.

- **Paso 8:**
A continuación, se calcula Y'.

  En Y' se tienen aquellos estados nuevos de Q' que contengan algún estado de Y. Es decir, los que contengan q_1 o q_3: $\{q_1, q_2\}$, $\{q_2, q_3\}$ y $\{q_1, q_2, q_3\}$.

  En la figura 7.3.9 de la página 37, se muestra lo obtenido en el paso 8.

7.3.4.1.3 Ejemplo 1: AFDC obtenido

En este ejemplo, han hecho falta 8 pasos para construir el AFDC requerido.

En la figura 7.3.9 de la página 37, se muestra el AFDC calculado.

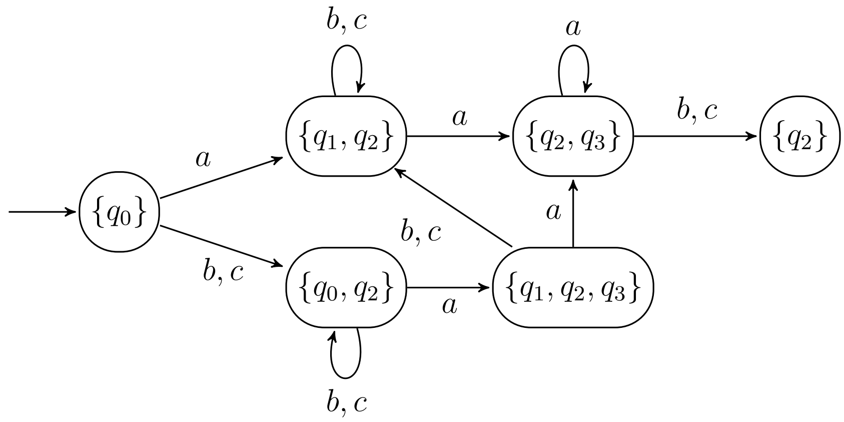


Figura 7.3.7. De AFND a AFD: ejemplo 1, paso 6.

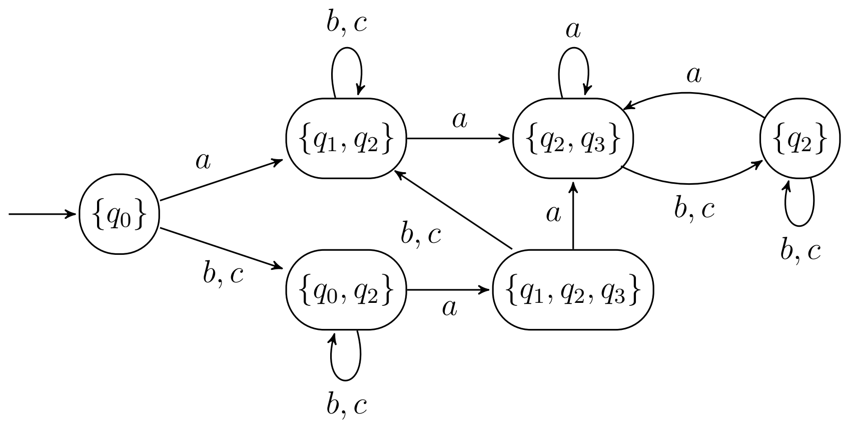


Figura 7.3.8. De AFND a AFD: ejemplo 1, paso 7.

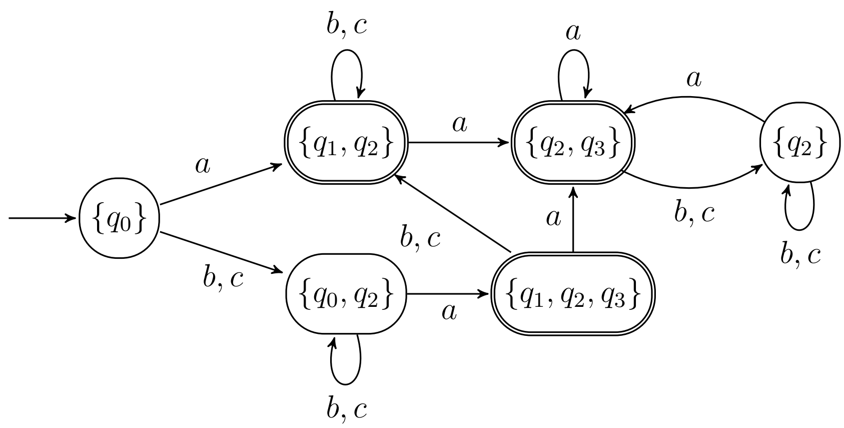


Figura 7.3.9. De AFND a AFD: ejemplo 1, paso 8. AFDC obtenido.

2^Q es el conjunto formado por todos los subconjuntos de $Q = \{q_0, q_1, q_2, q_3\}$:

$$\begin{array}{rl} 2^Q = & \{\varnothing, \{q_0\}, \{q_1\}, \{q_2\}, \{q_3\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_0, q_3\}, \{q_1, q_2\}, \{q_1, q_3\}, \\ & \{q_2, q_3\}, \{q_0, q_1, q_2\}, \{q_0, q_1, q_3\}, \{q_0, q_2, q_3\}, \{q_1, q_2, q_3\}, \{q_0, q_1, q_2, q_3\}\} \end{array}$$

Si Q tiene n elementos, 2^Q tendrá 2^n elementos. En este ejemplo, como Q tiene 4 elementos, 2^Q tiene 2^4 elementos, es decir, 16 elementos.

Los elementos de 2^Q que no aparecen en Q', representan estados que no son alcanzables desde el estado inicial $\{q_0\}$.

Los componentes del AFDC obtenido son los siguientes:

- $Q' = \{\{q_0\}, \{q_2\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_2, q_3\}, \{q_1, q_2, q_3\}\}$
- $\mathbb{A}' = \{a, b, c\}$
- $$\begin{array}{rl} \tau'_{\mathbb{A}'} = & \{(\{q_0\}, a, \{q_1, q_2\}), (\{q_0\}, b, \{q_0, q_2\}), (\{q_0\}, c, \{q_0, q_2\}), \\ & (\{q_0, q_2\}, a, \{q_1, q_2, q_3\}), (\{q_0, q_2\}, b, \{q_0, q_2\}), (\{q_0, q_2\}, c, \{q_0, q_2\}), \\ & (\{q_1, q_2\}, a, \{q_2, q_3\}), (\{q_1, q_2\}, b, \{q_1, q_2\}), (\{q_1, q_2\}, c, \{q_1, q_2\}), \\ & (\{q_2, q_3\}, a, \{q_2, q_3\}), (\{q_2, q_3\}, b, \{q_2\}), (\{q_2, q_3\}, c, \{q_2\}), \\ & (\{q_1, q_2, q_3\}, a, \{q_2, q_3\}), (\{q_1, q_2, q_3\}, b, \{q_1, q_2\}), (\{q_1, q_2, q_3\}, c, \{q_1, q_2\}), \\ & (\{q_2\}, a, \{q_2, q_3\}), (\{q_2\}, b, \{q_2\}), (\{q_2\}, c, \{q_2\})\} \end{array}$$
- $\tau'_\lambda = \varnothing$
- $\sigma' = \{q_0\}$
- $Y' = \{\{q_1, q_2\}, \{q_2, q_3\}, \{q_1, q_2, q_3\}\}$

7.3.4.2 Ejemplo 2 AFND – AFDC

7.3.4.2.1 Ejemplo 2: AFND original

En la figura 7.3.10 de la página 40, se muestra un AF $(R, \mathbb{A}, \tau_{\mathbb{A}}, \tau_\lambda, \sigma, Y)$ que no es determinista: en r_0 y r_2 para a hay más de una opción.

En ese AF, los componentes son los siguientes:

- $R = \{r_0, r_1, r_2, r_3\}$
- $\mathbb{A} = \{a, b, c\}$
- $$\begin{array}{rl} \tau_{\mathbb{A}} = & \{(r_0, a, r_0), (r_0, a, r_1), (r_1, b, r_2), (r_1, c, r_3), \\ & (r_2, a, r_0), (r_2, a, r_1), (r_2, a, r_2), (r_2, b, r_2), (r_3, c, r_3)\} \end{array}$$

- $\tau_\lambda = \varnothing$
- $\sigma = r_0$
- $Y = \{r_1, r_3\}$

7.3.4.2.2 Ejemplo 2: Pasos para construir un AFDC a partir del AFND original

El objetivo es construir un AF $(R', \mathbb{A}', \tau'_{\mathbb{A}'}, \tau'_\lambda, \sigma', Y')$ que defina el mismo lenguaje pero que sea determinista y completo y en el que el conjunto τ'_λ sea vacío.

Para empezar, sabemos que $\mathbb{A}' = \mathbb{A}$, $\tau'_\lambda = \varnothing$ y $\sigma' = \{\sigma\} = \{r_0\}$. Quedan por determinar R', $\tau'_{\mathbb{A}'}$ e Y'.

- **Paso 1:**
  Sabemos que el estado inicial σ' es $\{\sigma\}$ —es decir, $\{r_0\}$— y que $\{r_0\}$ pertenece a R'. En la figura 7.3.11 de la página 40, se muestra este punto de partida.

- **Paso 2:**
  A continuación, se calculan las transiciones correspondientes a $\{r_0\}$. En este caso, solo tenemos que considerar el estado r_0:

  - Con a, desde r_0 se llega a $\{r_0, r_1\}$
  - Con b, desde r_0 se llega a $\varnothing$
  - Con c, desde r_0 se llega a $\varnothing$

  Por tanto, se han de añadir esos dos estados nuevos —$\{r_0, r_1\}$ y $\varnothing$— y las transiciones desde $\{r_0\}$ a esos dos estados.

  En la figura 7.3.12 de la página 40, se muestra lo obtenido en el paso 2.

- **Paso 3:**
  A continuación, se calculan las transiciones correspondientes a $\{r_0, r_1\}$. En este caso, tenemos que considerar los estados r_0 y r_1:

  - Con a, desde r_0 se llega a $\{r_0, r_1\}$ y desde r_1 se llega a $\varnothing$. Juntando los dos casos, queda $\{r_0, r_1\} \cup \varnothing = \{r_0, r_1\}$.
  - Con b, desde r_0 se llega a $\varnothing$ y desde r_1 se llega a $\{r_2\}$. Juntando los dos casos, queda $\varnothing \cup \{r_2\} = \{r_2\}$.
  - Con c, desde r_0 se llega a $\varnothing$ y desde r_1 se llega a $\{r_3\}$. Juntando los dos casos, queda $\varnothing \cup \{r_3\} = \{r_3\}$.

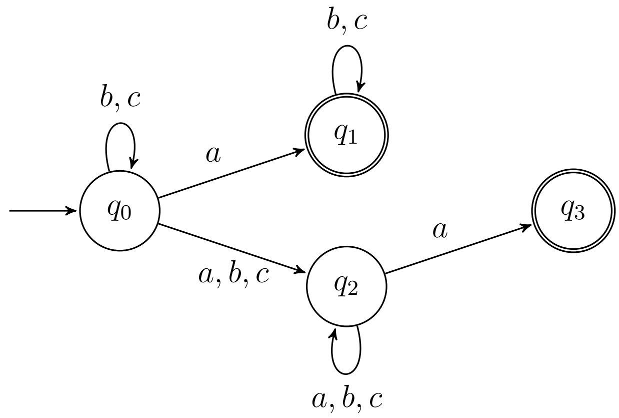


Figura 7.3.10. De AFND a AFDC: ejemplo 2. AFND de partida.

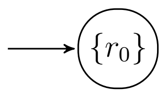


Figura 7.3.11. De AFND a AFD: ejemplo 2, paso 1.

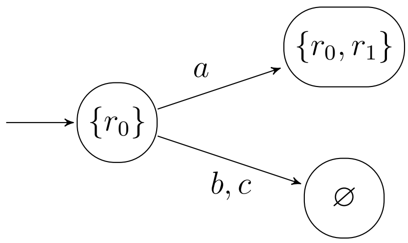


Figura 7.3.12. De AFND a AFD: ejemplo 2, paso 2.

Por tanto, se han de añadir los estados nuevos $\{r_2\}$ y $\{r_3\}$ y las transiciones desde $\{r_0, r_1\}$ que se han calculado.

En la figura 7.3.13 de la página 42, se muestra lo obtenido en el paso 3.

- **Paso 4:**
  A continuación, se calculan las transiciones correspondientes a $\{r_2\}$. En este caso, tenemos que considerar solo el estado r_2:

  - Con a, desde r_2 se llega a $\{r_0, r_1, r_2\}$.
  - Con b, desde r_2 se llega a $\{r_2\}$.
  - Con c, desde r_2 se llega a $\varnothing$.

  Por tanto, se ha de añadir el estado $\{r_0, r_1, r_2\}$ y las transiciones desde $\{r_2\}$ que se han calculado.

  En la figura 7.3.14 de la página 42, se muestra lo obtenido en el paso 4.

- **Paso 5:**
  A continuación, se calculan las transiciones correspondientes a $\{r_3\}$. En este caso, tenemos que considerar solo el estado r_3:

  - Con a, desde r_3 se llega a $\varnothing$.
  - Con b, desde r_3 se llega a $\varnothing$.
  - Con c, desde r_3 se llega a $\{r_3\}$.

  Por tanto, no hay que añadir ningún estado nuevo; solo las transiciones desde $\{r_3\}$ que se han calculado.

  En la figura 7.3.15 de la página 42, se muestra lo obtenido en el paso 5.

- **Paso 6:**
  A continuación, se calculan las transiciones correspondientes a $\{r_0, r_1, r_2\}$. En este caso, tenemos que considerar los estados r_0, r_1 y r_2:

  - Con a, desde r_0 se llega a $\{r_0, r_1\}$, desde r_1 se llega a $\varnothing$ y desde r_2 se llega a $\{r_0, r_1, r_2\}$. Juntando los tres casos, queda $\{r_0, r_1\} \cup \varnothing \cup \{r_0, r_1, r_2\} = \{r_0, r_1, r_2\}$.
  - Con b, desde r_0 se llega a $\varnothing$, desde r_1 se llega a $\{r_2\}$ y desde r_2 se llega a $\{r_2\}$. Juntando los tres casos, queda $\varnothing \cup \{r_2\} \cup \{r_2\} = \{r_2\}$.
  - Con c, desde r_0 se llega a $\varnothing$, desde r_1 se llega a $\{r_3\}$ y desde r_2 se llega a $\varnothing$. Juntando los dos casos, queda $\varnothing \cup \{r_3\} \cup \varnothing = \{r_3\}$.

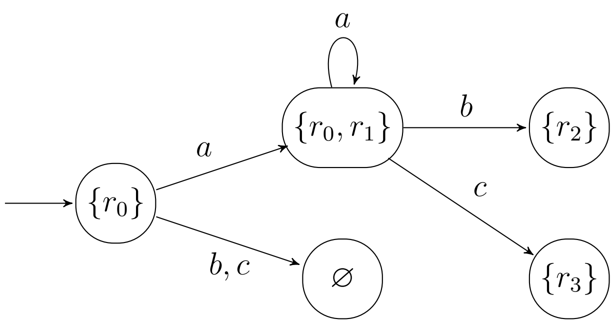


Figura 7.3.13. De AFND a AFD: ejemplo 2, paso 3.

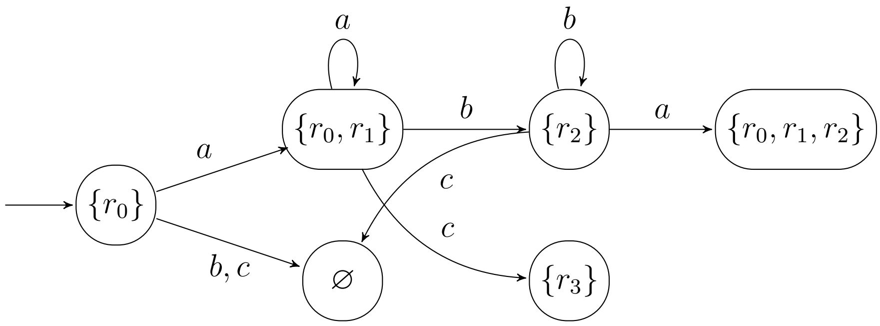


Figura 7.3.14. De AFND a AFD: ejemplo 2, paso 4.

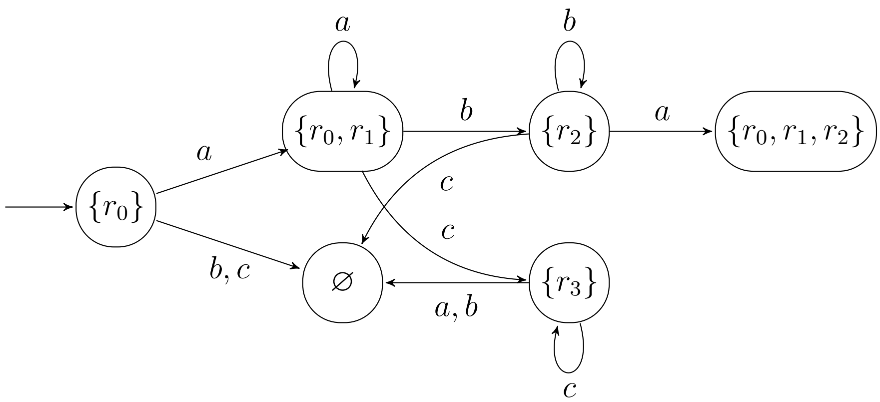


Figura 7.3.15. De AFND a AFD: ejemplo 2, paso 5.

Por tanto, no se ha de añadir ningún estado nuevo; solo las transiciones desde $\{r_0, r_1, r_2\}$ que se han calculado.

En la figura 7.3.16 de la página 44, se muestra lo obtenido en el paso 6.

- **Paso 7:**
  A continuación, se calculan las transiciones correspondientes a $\varnothing$. En este caso, no hay ningún estado a considerar. Por tanto, tanto con a, como con b como con c, desde $\varnothing$ se va a $\varnothing$. Consecuentemente, no se ha de añadir ningún estado nuevo; solo las transiciones desde $\varnothing$ que se han calculado.

  En la figura 7.3.17 de la página 44, se muestra lo obtenido en el paso 7.

- **Paso 8:**
  A continuación, se calcula Y'.

  En Y' se tienen aquellos estados nuevos de R' que contengan algún estado de Y. Es decir, los que contengan r_1 o r_3: $\{r_0, r_1\}$, $\{r_0, r_1, r_2\}$ y $\{r_3\}$.

  En la figura 7.3.18 de la página 44, se muestra lo obtenido en el paso 8.

7.3.4.2.3 Ejemplo 2: AFDC obtenido

En este ejemplo, han hecho falta 8 pasos para construir el AFDC requerido.

En la figura 7.3.18 de la página 44, se muestra el AFDC calculado.

$\mathbb{2}^R$ es el conjunto formado por todos los subconjuntos de $R = \{r_0, r_1, r_2, r_3\}$:

$$\begin{aligned} \mathbb{2}^R = \; & \{\varnothing, \{r_0\}, \{r_1\}, \{r_2\}, \{r_3\}, \{r_0, r_1\}, \{r_0, r_2\}, \{r_0, r_3\}, \{r_1, r_2\}, \{r_1, r_3\}, \\ & \{r_2, r_3\}, \{r_0, r_1, r_2\}, \{r_0, r_1, r_3\}, \{r_0, r_2, r_3\}, \{r_1, r_2, r_3\}, \{r_0, r_1, r_2, r_3\}\} \end{aligned}$$

Si R tiene n elementos, $\mathbb{2}^R$ tendrá 2^n elementos. En este ejemplo, como R tiene 4 elementos, $\mathbb{2}^R$ tiene 2^4 elementos, es decir, 16 elementos.

Los elementos de $\mathbb{2}^R$ que no aparecen en R', representan estados que no son alcanzables desde el estado inicial $\{r_0\}$.

Los componentes del AFDC obtenido son los siguientes:

- $R' = \{\varnothing, \{r_0\}, \{r_2\}, \{r_3\}, \{r_0, r_1\}, \{r_0, r_1, r_2\}\}$

- $\mathbb{A}' = \{a, b, c\}$

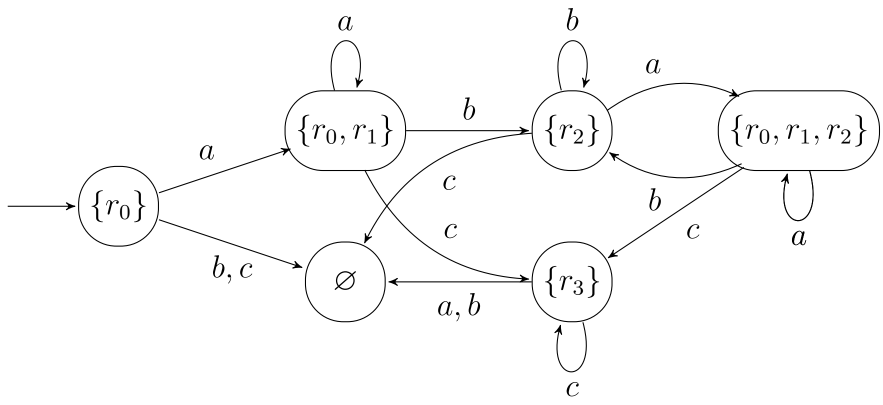


Figura 7.3.16. De AFND a AFD: ejemplo 2, paso 6.

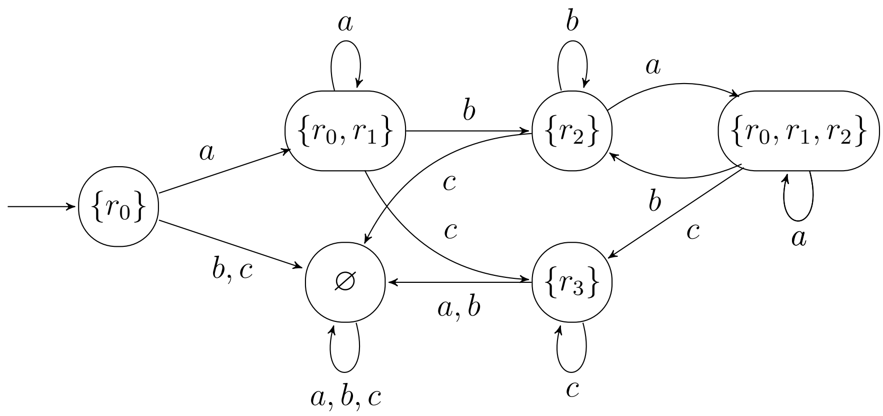


Figura 7.3.17. De AFND a AFD: ejemplo 2, paso 7.

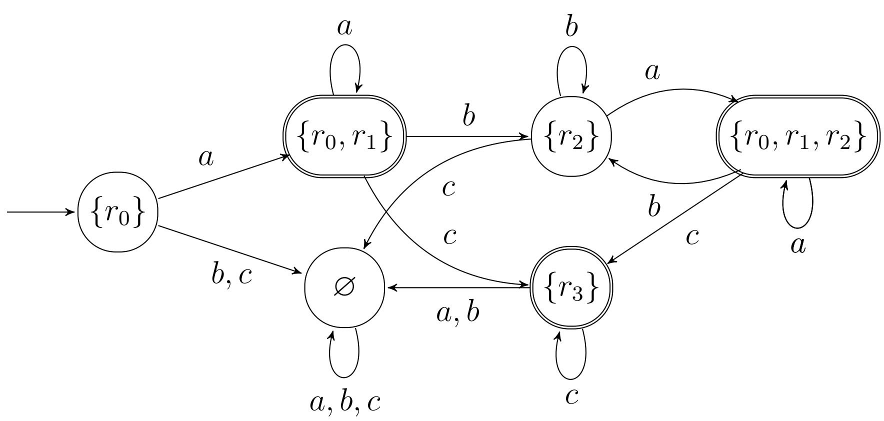


Figura 7.3.18. De AFND a AFD: ejemplo 2, paso 8. AFDC obtenido.

- $$\begin{aligned}\tau'_{\mathbb{A}'} = \; & \{(\{r_0\}, a, \{r_1, r_2\}), (\{r_0\}, b, \{r_0, r_2\}), (\{r_0\}, c, \{r_0, r_2\}), \\ & (\varnothing, a, \varnothing), (\varnothing, b, \varnothing), (\varnothing, c, \varnothing), \\ & (\{r_0, r_1\}, a, \{r_0, r_1\}), (\{r_0, r_1\}, b, \{r_2\}), (\{r_0, r_1\}, c, \{r_3\}), \\ & (\{r_2\}, a, \{r_0, r_1, r_2\}), (\{r_2\}, b, \{r_2\}), (\{r_2\}, c, \varnothing), \\ & (\{r_3\}, a, \varnothing), (\{r_3\}, b, \varnothing), (\{r_3\}, c, \{r_3\}), \\ & (\{r_0, r_1, r_2\}, a, \{r_0, r_1, r_2\}), (\{r_0, r_1, r_2\}, b, \{r_2\}), (\{r_0, r_1, r_2\}, c, \{r_3\})\}\end{aligned}$$
- $\tau'_\lambda = \varnothing$
- $\sigma' = \{r_0\}$
- $Y' = \{\{r_0, r_1\}, \{r_3\}, \{r_0, r_1, r_2\}\}$

7.4.

Método para transformar un AF en β-AF

7.4.1 β-AF

Se dice que un AF es un β-AF si sus funciones de transición cumplen las siguientes propiedades:

- No tener ninguna transición λ o tener exactamente dos transiciones λ por cada estado.
- Tener exactamente dos transiciones por cada estado y cada símbolo del alfabeto.

En la figura 7.4.1 se muestra el diagrama de transiciones de un AFDC correspondiente al lenguaje de las palabras que contienen una única a o terminan en a. Ese lenguaje se puede definir formalmente de la siguiente forma:

$$\{w \mid w \in \mathbb{A}^* \wedge (|w|_a = 1 \vee \exists u(u \in \mathbb{A}^* \wedge w = ua))\}$$

En el AFDC de la figura 7.4.1, el alfabeto $\mathbb{A}$ consta de los símbolos a, b y c y por cada estado y cada símbolo del alfabeto hay una única transición.

En la figura 7.4.2 se muestra el diagrama de transiciones de un AFNDNC correspondiente al lenguaje de las palabras que contienen una única a o terminan en a. En ese AFNDNC, el alfabeto $\mathbb{A}$ está formado por los símbolos a, b y c. En algunos estados no hay transiciones para algunos símbolos (por ejemplo, en el estado r_4 no hay transiciones para el símbolo a). En otros estados, para algunos símbolos hay más de una transición (por ejemplo, en el estado r_0 hay tres transiciones para el símbolo a).

En la figura 7.4.3 se muestra el diagrama de transiciones de un λ-AF correspondiente al lenguaje de las palabras que contienen una única a o terminan en a. El alfabeto de dicho λ-AF lo conforman los símbolos a, b y c. En el diagrama de transiciones de ese λ-AF hay dos transiciones vacías: la que va de s_0 a s_1 y la que va de s_0 a s_3. En un λ-AF puede ocurrir que para cada estado y cada símbolo del alfabeto haya una única transición (como en los AFDC). Pero también puede ocurrir que para al menos un estado y al menos un símbolo del alfabeto no haya transición o haya más de una transición (como en los AFNDNC). En el λ-AF de la figura

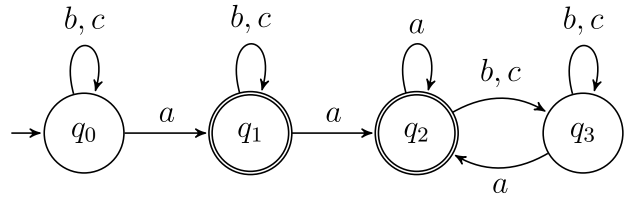


Figura 7.4.1. AFDC sin transiciones λ para las palabras que contienen cualquier número de b's y c's pero una única a o, si no, que terminan en a.

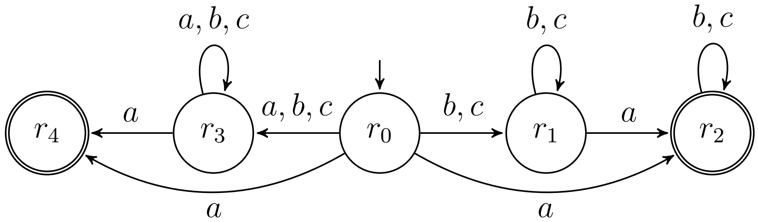


Figura 7.4.2. AFNDNC sin transiciones λ para las palabras que contienen cualquier número de b's y c's pero una única a o, si no, que terminan en a.

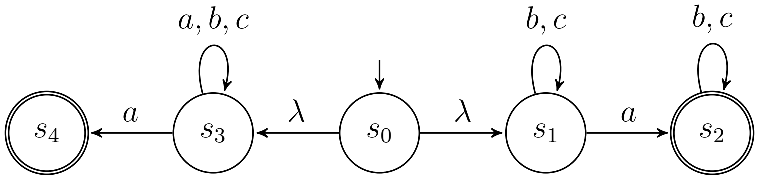


Figura 7.4.3. λ-AF para las palabras que contienen cualquier número de b's y c's pero una única a o, si no, que terminan en a.

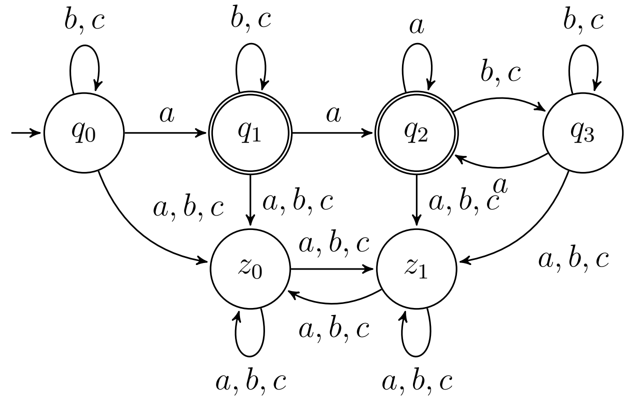


Figura 7.4.4. β-AF equivalente obtenido a partir del AFD de la figura 7.4.1.

7.4.3, por ejemplo, en el estado s_0 no hay ninguna transición para c y en el estado s_2 no hay ninguna transición para a. Por otra parte, en el estado s_3 hay dos transiciones para a.

En la figura 7.4.4 se muestra el diagrama de transiciones de un β-AF correspondiente al lenguaje de las palabras que contienen una única a o terminan en a. Ese β-AF se ha obtenido a partir del AFDC de la figura 7.4.1. Los símbolos del alfabeto son a, b y c y por cada estado y cada símbolo, hay exactamente dos transiciones. Por otra parte, no hay ninguna transición vacía (o transición λ). Para obtener un β-AF a partir del AFDC de la figura 7.4.1, se han creado los estados z_0 y z_1. Al construir el β-AF de la figura 7.4.4 se ha tenido en cuenta la idea presentada en el apartado 7.4.3.2 (figura 7.4.10 y figura 7.4.11).

En la figura 7.4.5 se muestra el diagrama de transiciones de un segundo β-AF correspondiente al lenguaje de las palabras que contienen una única a o terminan en a. Ese β-AF se ha obtenido a partir del AFNDNC de la figura 7.4.2. En este caso, además de las transiciones correspondientes a los tres símbolos a, b y c que conforman el alfabeto, es necesario incluir también transiciones vacías (o transiciones λ). Esa necesidad surge del hecho de que para el estado r_0 del AFNDNC de la figura 7.4.2 hay tres transiciones con el símbolo a. Para que desde r_0 haya solo dos transiciones con el símbolo a, se ha creado el estado x_0. Desde el estado x_0 se han puesto transiciones λ a los estados r_2 y r_4. Tras crear ese estado, desde r_0 hay solo dos transiciones con el símbolo a: una que va al estado r_3 y otra que va al estado x_0. Tras introducir esas dos transiciones λ, para construir un β-AF es necesario introducir también dos transiciones λ en el resto de estados. Recordemos que si en un β-AF aparece alguna transición λ, entonces es obligatorio que salgan exactamente dos transiciones λ desde todos los estados. Por tanto, en el β-AF de la figura 7.4.5, para cada estado y cada símbolo del alfabeto se tienen exactamente dos transiciones y, además, para cada estado se tienen exactamente dos transiciones vacías. Para obtener un β-AF a partir del AFNDNC de la figura 7.4.2, además del estado x_0 se han tenido que crear también los estados z_0 y z_1. Al construir el β-AF de la figura 7.4.5 se han tenido en cuenta las ideas presentadas en los apartados 7.4.3.1, 7.4.3.2 eta 7.4.3.3 (figuras 7.4.7, 7.4.8, 7.4.9, 7.4.10, 7.4.11, 7.4.12 y 7.4.13). Además, al construir el β-AF de la figura 7.4.5 de la página 50, se han puesto dos estados adicionales z'_0 y z'_1 que son copias de los estados z_0 eta z_1 y cuyo objetivo es facilitar la legibilidad del gráfico.

A modo de tercer ejemplo, en la figura 7.4.6 se muestra el diagrama de transiciones de otro β-AF correspondiente al lenguaje de las palabras que contienen una única a o terminan en a. Para obtener ese β-AF, se ha tomado como punto de partida el λ-AF de la figura 7.4.3 y se han realizado los cambios necesarios. Puesto que el AF de partida es un λ-AF, en el β-AF obtenido, aparte de tener exactamente dos transiciones para cada estado y cada símbolo del alfabeto, es obligatorio que desde cada estado salgan exactamente dos transiciones vacías. Por definición, si en un β-AF hay alguna transición λ, entonces desde cada estado han de salir exactamente dos transiciones λ. Para llegar a tener un β-AF, se han tenido que crear los estados nuevos z_0 y z_1. Al construir el β-AF de la figura 7.4.6 se han tenido en cuenta las ideas presentadas en los apartados 7.4.3.2 y 7.4.3.3 (figuras 7.4.10, 7.4.11, 7.4.12 y 7.4.13). Además, al construir el β-AF de la figura 7.4.6 de la página 50, se han puesto dos estados adicionales z'_0 y z'_1 que son

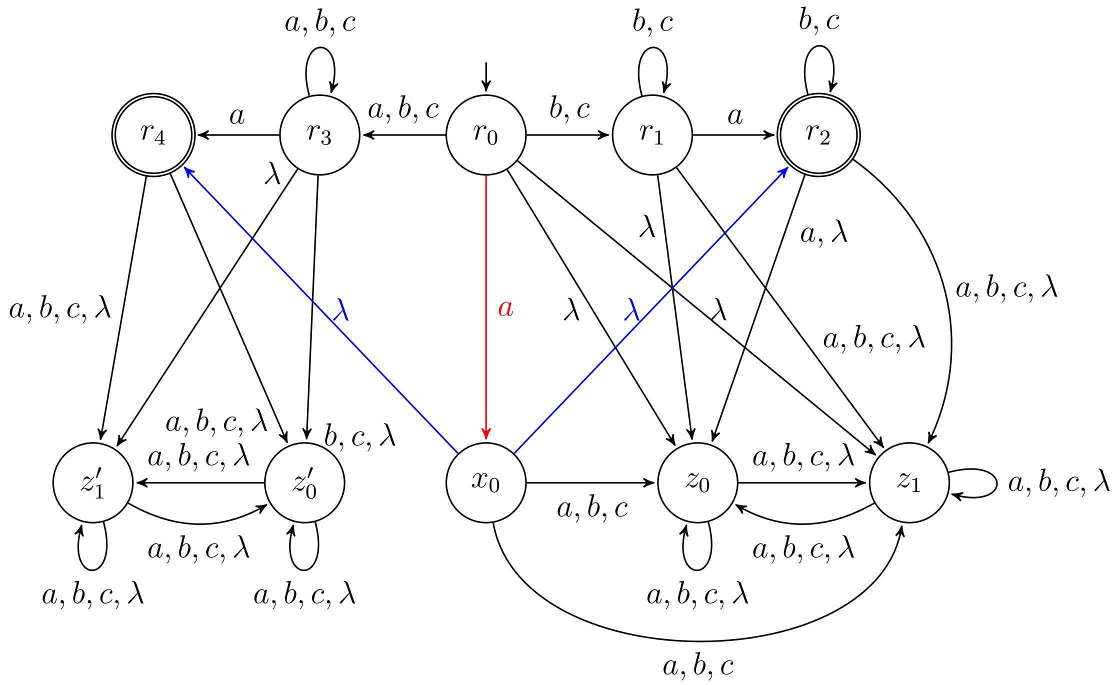


Figura 7.4.5. β-AF equivalente obtenido a partir del AFNDNC de la figura 7.4.2.

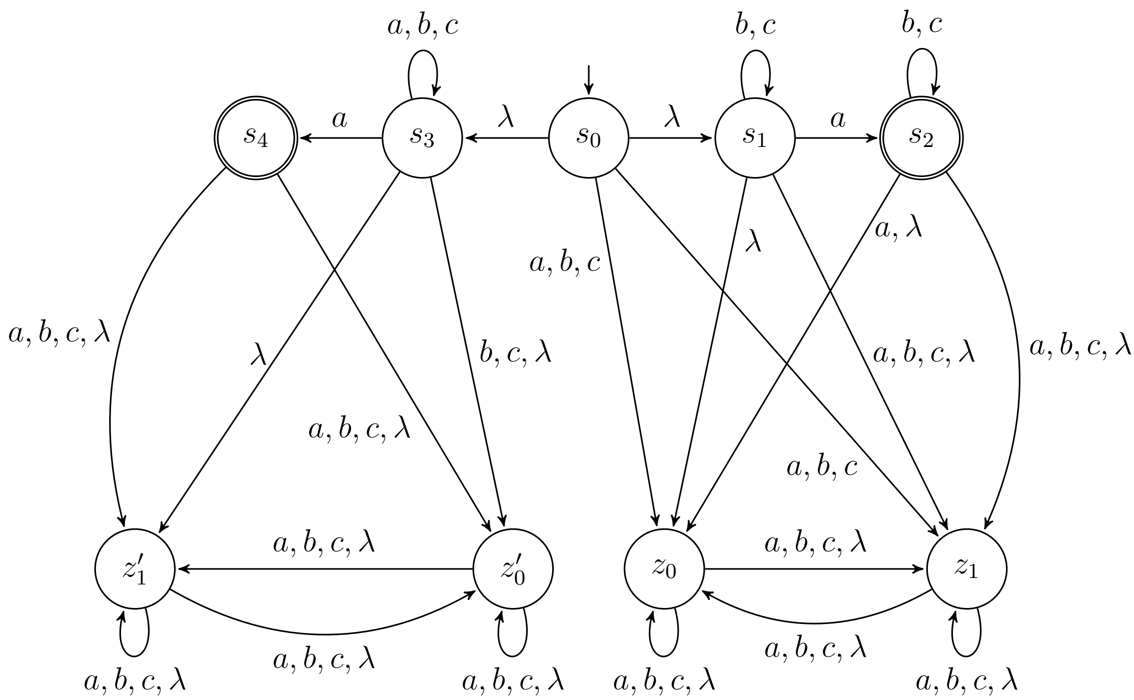


Figura 7.4.6. β-AF equivalente obtenido a partir del λ-AF de la figura 7.4.3.

copias de los estados z_0 eta z_1 y cuyo objetivo es facilitar la legibilidad del gráfico.

7.4.2 Interés de utilizar los β-AF

Dado un lenguaje, la tarea de diseñar un AFNDNC o un λ-AF correspondiente a ese lenguaje suele ser más fácil que el diseñar un AFDC. Desde ese punto de vista, los AFNDNC [1] y los λ-AF son más prácticos que los AFDC. Sin embargo, la función de transición de un AFDC contiene una única transición por cada estado y cada símbolo del alfabeto, mientras que la función de transición de un AFNDNC o de un λ-AF no tiene, en general, esa uniformidad (puesto que, en el caso de los AFNDNC, al menos para un estado y un símbolo del alfabeto no habrá ninguna transición o habrá más de una transición o también, en el caso de los λ-AF, además de poder ocurrir lo que se ha indicado para los AFNDNC, puede suceder que para un estado no haya ninguna transición vacía o haya más de una). Desde esa perspectiva, los AFDC son más adecuados. Dado un AFNDNC o un λ-AF, siempre se puede calcular u obtener un AFDC equivalente de manera sistemática (siguiendo un algoritmo). Por tanto, cuando se quiera construir un AF correspondiente a un lenguaje, podemos construir primero un AFNDNC o un λ-AF (porque es más fácil que diseñar directamente un AFDC) y, en una segunda fase, se puede obtener, de manera sistemática, un AFDC. Teniendo en cuenta lo expuesto hasta ahora, se puede pensar que para trabajar tanto los aspectos teóricos como prácticos de los AF, la mejor opción es utilizar los AFDC, debido a la uniformidad de la función de transición en los AFDC y porque a partir de un AFNDNC y un λ-AF siempre se puede obterner un AFDC equivalente de manera sistemática. Pero, en general, el coste computacional del cálculo de un AFDC a partir de un AFNDNC o un λ-AF es exponencial: si un AFNDNC o un λ-AF tiene n estados, el AFDC equivalente tendrá un número de estados del orden de 2^n. Consecuentemente, calcular un AFDC a partir de un AFNDNC o un λ-AF no es eficiente ni práctico. Ante ese problema, se justifica el interés por los β-AF. El coste computacional del cálculo de un β-AF equivalente a partir de un AFDC, un AFNDNC o un λ-AF no es exponencial: si en un estado para un símbolo hay k transiciones, para simular esas k transiciones mediante opciones binarias, se necesitan $log_2 k$ (con redondeo hacia arriba) estados nuevos. Esto significa que el proceso de construcción de un β-AF equivalente a un AF es muy eficiente. Además, puesto que en un β-AF se tienen exactamente dos transiciones por cada estado y cada símbolo, la función de transición es uniforme. Debido a todas estas consideraciones, los β-AF son muy importantes para formalizar algunas demostraciones teóricas sobre los AF y también para cuando se quieran diseñar, de manera eficiente, autómatas finitos con funciones de transiciones uniformes en cuanto al número de transiciones por cada estado y cada símbolo.

[1] Respecto a la pluralización de las siglas, la RAE (Real Academia Española) recomienda mantenerlas invariables e indicar el plural mediante los determinantes. Por tanto, se recomienda escribir *los AFDC, los AFNDNC, los β-AF, los λ-AF, unos AFDC, varios AFDC, muchos AFDC, diferentes AFDC, cuántos AFDC*, etc., sin añadir *ese* al final. Por eso, es recomendable utilizar siempre un determinante para introducir la sigla cuando esta ha de expresar pluralidad. En su lectura sí se añade una *ese*, de modo que *los AFDC* se dice *los aefedecés*. Ver https://www.rae.es/consultas/plural-de-las-siglas-las-ong-unos-dvd, https://www.fundeu.es/consulta/plural-de-abreviaturas-y-siglas-402/ y https://www.fundeu.es/recomendacion/siglas-y-acronimos-claves-de-redaccion/.

7.4.3 Esquemas para la construcción de los β-AF

7.4.3.1 Más de dos transiciones desde un estado con un mismo símbolo o λ

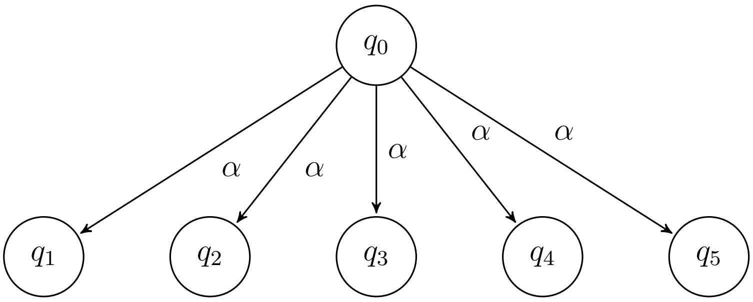
Cuando desde un estado hay más de dos transiciones con un mismo símbolo o λ, con el objeto de tener solo dos transiciones se han de generar nuevos estados y utilizar transiciones λ. Por ejemplo, si con respecto a un símbolo tenemos la estructura que se presenta en la figura 7.4.7, podemos sustituirla por la estructura de la figura 7.4.8 o por la estructura de la figura 7.4.9. Además, habrá que tener en cuenta también el resto de símbolos del alfabeto y habrá que poner transiciones λ donde corresponda, para lo cual se aplicarán los esquemas que se introducen en los apartados 7.4.3.2 y 7.4.3.3 cuando sea necesario. En las dos estructuras presentadas (figuras 7.4.8 y 7.4.9), se han creado los estados x_0, x_1 y x_2. Desde el estado q_0 hay dos transiciones con el símbolo $\alpha \in (\mathbb{A} \cup \{\lambda\})$ y todas las otras transiciones de las dos estructuras nuevas son vacías. Tal como se ha indicado en el apartado 7.4.2, cuando se tenga una estructura como la de la figura 7.4.7, si hay k transiciones con un símbolo concreto, entonces el número de estados nuevos necesarios será del orden de log_2k. Retomando nuestro ejemplo, al haber 5 transiciones con $\alpha \in (\mathbb{A} \cup \{\lambda\})$ (figura 7.4.7), se han necesitado 3 estados nuevos, es decir, log_25 con redondeo hacia arriba.

7.4.3.2 Una única transición desde un estado con un símbolo o λ

Si desde un estado con un símbolo del alfabeto o con λ solo hay una transición, entonces para pasar a tener dos transiciones podemos utilizar dos estados nuevos. Por ejemplo, ante el caso que se muestra en la figura 7.4.10, una opción es generar la estructura presentada en la figura 7.4.11. En esa última estructura tenemos dos estados nuevos, z_0 y z_1. Estos estados serán los mismos para cualquier aplicación del esquema descrito en la figura 7.4.11 que sea necesaria, y también serán los mismos para cualquier aplicación del esquema descrito en la figura 7.4.13. La estructura tendrá que ser completada con el resto de símbolos del alfabeto y con transiciones vacías, si ello es necesario, mediante sucesivas aplicaciones de los esquemas en las figuras 7.4.11 y 7.4.13.

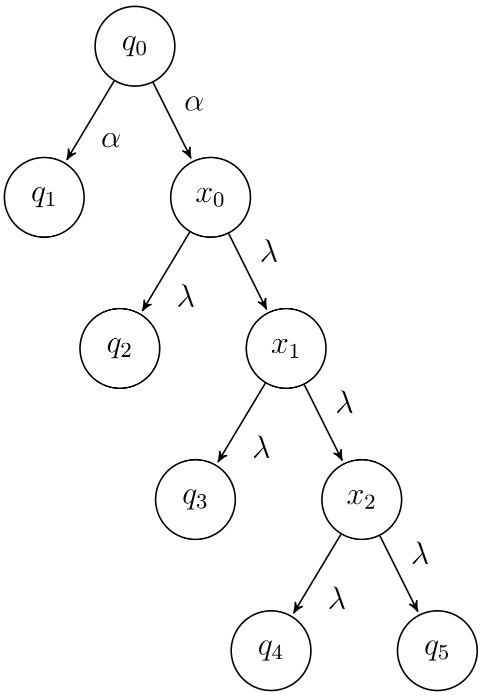
7.4.3.3 Ninguna transición desde un estado con un símbolo o λ

Cuando para un símbolo del alfabeto o para λ no hay ninguna transición desde un estado concreto, para que salgan dos transiciones desde ese estado con el símbolo considerado, podemos utilizar dos estados nuevos. Por ejemplo, ante una situación como la mostrada en la figura 7.4.12, donde para un símbolo α concreto, perteneciente a $\mathbb{A} \cup \{\lambda\}$, no hay ninguna transición, podemos generar la estructura de la figura 7.4.13. En esa segunda estructura se han añadido dos estados nuevos z_0 y z_1, que serán los mismos para cualquier aplicación de los esquemas descritos en las figuras 7.4.11 y 7.4.13 como hemos indicado anteriormente. Además, estos mismos esquemas serán utilizados también para completar la estructura teniendo en cuenta el resto de símbolos del alfabeto y las transiciones vacías, si es que estas últimas son necesarias.



siendo $\alpha \in (\mathbb{A} \cup \{\lambda\})$

Figura 7.4.7. Cálculo de β-AF cuando desde un estado hay más de dos transiciones con el mismo símbolo o λ (ver figuras 7.4.8 y 7.4.9).



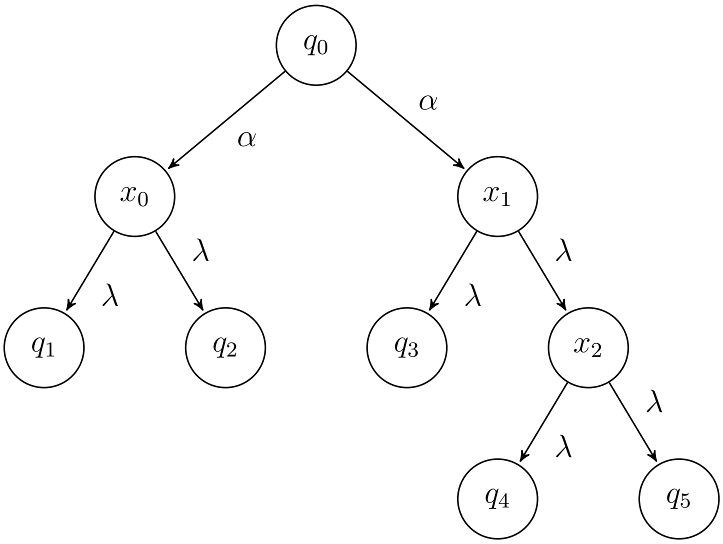
siendo $\alpha \in (\mathbb{A} \cup \{\lambda\})$

Figura 7.4.8. Una opción para transformar la estructura de la figura 7.4.7 con el objetivo de calcular un β-AF.

7.4.3.4 Algoritmo para la transformación a β-AFND

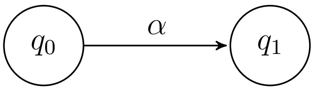
Dado un AF cualquiera, se puede obtener un β-AFND equivalente mediante la aplicación sucesiva de los esquemas descritos en los apartados anteriores.

Un posible algoritmo que permite la transformación de AF a β-AF es descrito en la figura 7.4.14.



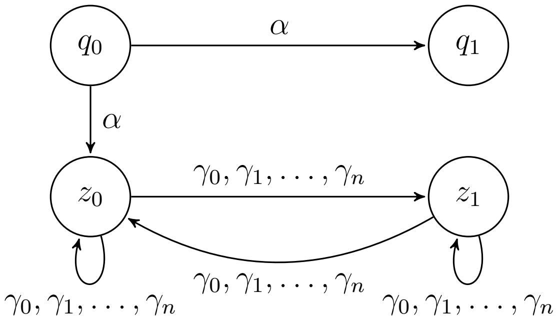
siendo $\alpha \in (\mathbb{A} \cup \{\lambda\})$

Figura 7.4.9. Otra opción para transformar la estructura de la figura 7.4.7 con el objetivo de calcular un β-AF.



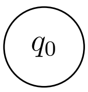
siendo $\alpha \in (\mathbb{A} \cup \{\lambda\})$

Figura 7.4.10. Cálculo de β-AF cuando desde un estado hay una única transición con un símbolo o con λ (ver figura 7.4.11).



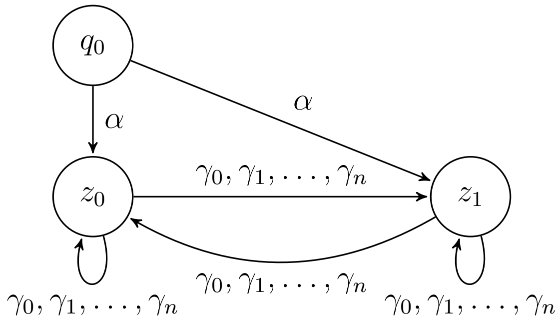
siendo $\alpha \in (\mathbb{A} \cup \{\lambda\})$ y
$\{\gamma_0, \gamma_1, \ldots, \gamma_n\} = \mathbb{A}$ o
$\{\gamma_0, \gamma_1, \ldots, \gamma_n\} = (\mathbb{A} \cup \{\lambda\})$
si λ ha de ser considerada.

Figura 7.4.11. Una opción para transformar la estructura de la figura 7.4.10 con el objetivo de calcular un β-AF.



para algún $\alpha \in (\mathbb{A} \cup \{\lambda\})$

Figura 7.4.12. Cálculo de β-AF cuando desde un estado no hay ninguna transición con un símbolo o con λ (ver figura 7.4.13).



siendo $\alpha \in (\mathbb{A} \cup \{\lambda\})$ y
$\{\gamma_0, \gamma_1, \ldots, \gamma_n\} = \mathbb{A}$ o
$\{\gamma_0, \gamma_1, \ldots, \gamma_n\} = (\mathbb{A} \cup \{\lambda\})$
si λ ha de ser considerada.

Figura 7.4.13. Una opción para transformar la estructura de la figura 7.4.12 con el objetivo de calcular un β-AF.

ALGORITMO

Paso 1: Para cada estado $q \in Q$ y cada $\alpha \in (\mathbb{A} \cup \{\lambda\})$ tal que haya más de 2 transiciones con origen en q para α, aplicar uno de los esquemas descritos en el apartado 7.4.3.1.
El conjunto de estados del autómata resultante se denota mediante Q'.

Paso 2: Si para cada estado $q \in Q'$ y cada $\alpha \in \mathbb{A}$ hay exactamente 2 transiciones con origen en q para α y bien no hay ninguna transición con λ para ningún $q \in Q'$ o bien para cada estado $q \in Q'$ hay exactamente 2 transiciones con λ con origen en q, entonces **FIN**.
Si no, ir al Paso 3.

Paso 3: Crear 2 nuevos estados z_0 y z_1 y añadir transiciones para todo $\alpha \in \mathbb{A}$ con origen en z_0 y destino a z_0 (bucle) y destino a z_1, y hacer lo mismo considerando z_1 como origen y z_0 y z_1 como destinos. Además, si existe alguna transición con λ en el autómata, añadir una transición con λ de z_0 a z_0 (bucle) y otra de z_0 a z_1 y también de z_1 a z_0 y a z_1.

Paso 4: Para cada estado $q \in Q'$ y cada $\alpha \in \mathbb{A}$ tal que solo hay una transición con origen en q para α, añadir una transición con origen en q y destino a z_0 para α.
Además, si hay alguna transición con λ en el autómata, para cada estado $q \in Q'$ tal que solo hay una transición con origen en q para λ, añadir una transición con origen en q y destino a z_0 para λ.

Paso 5: Para cada estado $q \in Q'$ y cada $\alpha \in \mathbb{A}$ tal que no hay ninguna transición con origen en q para α, añadir una transición con origen en q y destino a z_0 y otra transición con origen en q y destino a z_1 para α.
Además, si hay alguna transición con λ en el autómata, para cada estado $q \in Q'$ tal que no hay ninguna transición con origen en q para λ, añadir una transición con origen en q y destino a z_0 y otra transición con origen en q y destino a z_1 para λ.

Figura 7.4.14. Algoritmo para obtener un β-AF a partir de un AF.

7.5.

Método para decidir si dos AF definen el mismo lenguaje

Se dice que dos AFDC son equivalentes si tienen como lenguaje asociado el mismo lenguaje. Para que dos AFDC puedan ser equivalentes es imprescindible que tengan el mismo alfabeto. El resto de componentes de los AFDC pueden ser distintos.

En los siguientes apartados, primero se presenta un algoritmo que decide si dos AFDC son equivalentes y, a continuación, se muestran tres ejemplos de aplicación del algoritmo. En el primer ejemplo, el algoritmo se aplica a dos AFDC equivalentes mientras que en el segundo ejemplo y el tercer ejemplo el algoritmo es aplicado a dos AFDC no equivalentes.

7.5.1 Algoritmo para decidir si dos AFDC son equivalentes

Como se acaba de indicar, decidir si dos AFDC son equivalentes supone averiguar si el lenguaje asociado a uno de ellos coincide con el lenguaje asociado al otro. Puesto que los lenguajes asociados pueden ser infinitos, la opción de comparar directamente los lenguajes no es adecuada. El algoritmo que se va a presentar consiste en la construcción de un árbol de comparación de autómatas. Este árbol permite convertir el problema de la comparación de los lenguajes asociados a los AFDC en un problema de comparación de los estados de los AFDC.

Dados dos AFDC $D = (Q, \mathbb{A}, \tau_{\mathbb{A}}, \tau_{\lambda}, \sigma, Y)$ y $D' = (Q', \mathbb{A}, \tau'_{\mathbb{A}}, \tau'_{\lambda}, \sigma', Y')$, se dice que un estado q perteneciente a Q y un estado q' perteneciente a Q' son **compatibles** si y solo si q pertenece a Y y q' pertenece a Y' o q pertenece a $Q \setminus Y$ y q' pertenece a $Q' \setminus Y'$. Dicho de otra forma, si se cumple que $q \in Y$ y $q' \in Y'$ o se cumple que $q \notin Y$ y $q' \notin Y'$, entonces q y q' son compatibles. Si no se cumple esa condición, los dos estados en cuestión son **incompatibles**.

La idea del algoritmo de comparación de dos AFDC D y D' consiste en averiguar si existe alguna secuencia de símbolos, es decir, alguna palabra w tal que siguiéndola simultáneamente en D y D' se llegue a estados incompatibles. Si no existe ninguna secuencia de símbolos que, siguiéndola simultáneamente en D y D', lleve a estados incompatibles, entonces D y D' son equivalentes.

Puesto que el número de posibles palabras puede ser infinito, el **algoritmo** se basa en examinar todas las posibles combinaciones de estados de D y D', es decir, todos los pares de estados de la forma (q, q') donde $q \in Q$ y $q' \in Q'$. El número de los pares generables con ese criterio es siempre finito porque Q y Q' son siempre finitos. En definitiva, se explora el conjunto $Q \times Q'$. Dicha exploración se realiza mediante la construcción de un árbol siguiendo los siguientes pasos:

$\#1$. Inicialmente, crear un árbol formado únicamente por el par (σ, σ'), que será la raíz. A continuación ir al paso $\#2$.

$\#2$. Si alguna hoja del árbol está formado por un par (q, q') de estados incompatibles, terminar la construcción del árbol. La conclusión es que los AFDC no son equivalentes. En caso contrario, es decir, si todas las hojas están formadas por pares de estados que son compatibles, seguir en el paso $\#3$.

$\#3$. Puesto que todas las hojas están formadas por pares de estados que son compatibles, si en cada rama del árbol la hoja está formada por un par (q, q') que aparece previamente en la misma rama, es decir, el par (q, q') está repetido en la rama, terminar la construcción del árbol. La conclusión es que los AFDC son equivalentes. En caso contrario, es decir, si todas las hojas están formadas por pares de estados que son compatibles y, además, alguna de esas hojas está formada por un par de estados que no aparece repetido en su rama, entonces continuar en el paso $\#4$.

$\#4$. Puesto que todas las hojas están formadas por pares de estados que son compatibles y hay al menos una hoja que no está repetida en su rama, seleccionar una hoja formada por un par (q, q') tal que el par (q, q') no esté repetido en esa rama y desarrollarlo. Para desarrollar la hoja (q, q'), habrá que tener en cuenta las triplas (q, α, q'') y (q', α, q''') para cada $\alpha \in \mathbb{A}$ y añadir como hijo de la hoja (q, q') el par (q'', q''') obtenido para cada $\alpha \in \mathbb{A}$. Por tanto, se ha de añadir un hijo por cada símbolo del alfabeto $\mathbb{A}$. Después de añadir los nuevos nodos indicados, continuar en el paso $\#2$.

En resumen, el algoritmo parte del árbol cuya raíz es (σ, σ') y va expandiendo el árbol hasta que todas las hojas aparezcan repetidas en sus respectivas ramas o hasta que surja una hoja formada por estados incompatibles. Las hojas que están repetidas en sus respectivas ramas no admiten expansión. En cada paso de expansión se añade un nodo por cada símbolo del alfabeto.

7.5.2 Ejemplos de aplicación del algoritmo para decidir si dos AFDC son equivalentes

A continuación se ilustra con tres ejemplos la aplicación del algoritmo introducido en el apartado anterior y que sirve para averiguar si dos AFDC que tienen el mismo alfabeto son equivalentes.

7.5.2.1 Ejemplo de dos AFDC equivalentes: D_1 y D_2

En la figura 7.5.1 se muestra el diagrama de transiciones del AFDC D_1 para el lenguaje de las palabras que no contienen el símbolo b. Por otro parte, en la figura 7.5.2 se muestra el diagrama de transiciones del AFDC D_2 cuyo lenguaje asociado es también el lenguaje de las palabras que no contienen el símbolo b.

En la figura 7.5.3 se muestra el árbol de comparación de autómatas que construiría el algoritmo presentado. Puesto que todas las ramas terminan por repetición de nodo, la conclusión final es que D_1 y D_2 son equivalentes.

Para construir ese árbol, el algoritmo seguiría los siguientes pasos:

#1. Crear un árbol cuya raíz y único nodo es el par (σ, σ'), es decir, el nodo (r_0, s_0). Ir al punto #2.

#2. Como los estados r_0 y s_0 que conforman la única hoja del árbol son compatibles, ir al punto #3.

#3. Como la única hoja (r_0, s_0) del árbol no está repetida en su rama, ir al punto #4.

#4. (r_0, s_0) es la única hoja que se tiene y la única que no está repetida en su rama. Por tanto, se seleccionará la hoja (r_0, s_0) y se añadirán los nodos (r_1, s_0), (r_2, s_1) y (r_1, s_0). Ahora se ha de volver al punto #2.

#2. Se tienen tres hojas: (r_1, s_0), (r_2, s_1) y (r_1, s_0). Ninguna de ellas está formada por estados incompatibles. Consecuentemente, se ha de ir al punto #3.

#3. Ninguna de las tres hojas está repetida en su rama. Se ha de ir al punto #4.

#4. Hay tres hojas seleccionables: (r_1, s_0), (r_2, s_1) y (r_1, s_0). Hay que seleccionar una de ellas y expandir el árbol. El algoritmo no fija ningún orden de selección, lo cual significa que se puede seleccionar cualquiera de las tres hojas. Si seleccionamos la primera hoja, es decir, el par (r_1, s_0), se generarán los pares (r_0, s_0), (r_2, s_1) y (r_0, s_0). A continuación se ha de ir al punto #2.

#2. Ahora se tienen cinco hojas: (r_0, s_0), (r_2, s_1) y (r_0, s_0) generadas en el punto anterior y (r_2, s_1) y (r_1, s_0) generadas la primera vez que se ha pasado por el punto #4. Como en las cinco hojas los estados que las conforman son compatibles, se ha de ir al punto #3.

#3. Hay al menos una hoja que no aparece repetida en su rama. En concreto, las hojas (r_2, s_1), (r_2, s_1) y (r_1, s_0) no aparecen repetidas en sus respectivas ramas. Por tanto, se ha de ir al punto #4.

#4. Hay tres hojas seleccionables: (r_2, s_1), (r_2, s_1) y (r_1, s_0). Se podría seleccionar cualquiera de las tres para realizar la siguiente expansión. Si se selecciona la primera, es decir, (r_2, s_1), se generarán los nodos (r_2, s_1), (r_2, s_1) y (r_2, s_1). A continuación se ha de ir al punto #2.

#2. Se tienen siete hojas: (r_0, s_0), (r_2, s_1), (r_2, s_1), (r_2, s_1), (r_0, s_0), (r_2, s_1) y (r_1, s_0). Ninguna de ellas está formada por estados incompatibles. Consecuentemente, se ha de ir al punto #3.

#3. Hay al menos una hoja que no aparece repetida en su rama. En concreto, las hojas (r_2, s_1) y (r_1, s_0) generadas la primera vez que se ha pasado por el punto #4 no aparecen repetidas en sus respectivas ramas. Por tanto, se ha de ir al punto #4.

#4. Hay dos hojas seleccionables: (r_2, s_1) y (r_1, s_0). Se podría seleccionar cualquiera de las dos. Supongamos que se selecciona el nodo (r_2, s_1). Se generarán los nodos (r_2, s_1), (r_2, s_1) y (r_2, s_1). A continuación se ha de ir al punto #2.

#2. Se tienen nueve hojas: (r_0, s_0), (r_2, s_1), (r_2, s_1), (r_2, s_1), (r_0, s_0), (r_2, s_1), (r_2, s_1), (r_2, s_1) y (r_1, s_0). Ninguna de ellas está formada por estados incompatibles. Consecuentemente, se ha de ir al punto #3.

#3. Hay al menos una hoja que no aparece repetida en su rama. En concreto, la hoja (r_1, s_0). Por tanto, se ha de ir al punto #4.

#4. Solo hay una hoja seleccionable: (r_1, s_0). Consecuentemente, se ha de seleccionar la hoja (r_1, s_0). Se generarán los nodos (r_0, s_0), (r_2, s_1) y (r_0, s_0). A continuación se ha de ir al punto #2.

#2. Se tienen once hojas: (r_0, s_0), (r_2, s_1), (r_2, s_1), (r_2, s_1), (r_0, s_0), (r_2, s_1), (r_2, s_1), (r_2, s_1) (r_0, s_0), (r_2, s_1) y (r_0, s_0). Ninguna de ellas está formada por estados incompatibles. Consecuentemente, se ha de ir al punto #3.

#3. Hay al menos una hoja que no aparece repetida en su rama. En concreto, la hoja (r_2, s_1) generada la anterior vez que se ha pasado por el punto #4. Por tanto, se ha de ir otra vez al punto #4.

#4. Solo hay una hoja seleccionable: (r_2, s_1). Por consiguiente, se ha de seleccionar la hoja (r_2, s_1). Se generarán los nodos (r_2, s_1), (r_2, s_1) y (r_2, s_1). A continuación se ha de ir al punto #2.

#2. Se tienen trece hojas. Ninguna de ellas está formada por estados incompatibles, lo cual significa que se ha de ir al punto #3.

#3. Todas las hojas aparecen repetidas en sus respectivas ramas. Se ha de terminar. La conclusión es que los AFDC D_1 y D_2 son equivalentes.

Al desarrollar el ejemplo, y en particular al pasar por el punto #4, se ha podido constatar que a veces es necesario seleccionar un nodo entre varios candidatos. Es importante recalcar que en este ejemplo, independientemente del nodo seleccionado cada vez que se pase por el punto #4, al final se construirá el mismo árbol aunque durante el proceso de construcción se tendrán árboles provisionales que serán distintos en función del nodo seleccionado para llevar a cabo cada paso de expansión.

7.5.2.2 Primer ejemplo de dos AFDC no equivalentes: D_3 y D_4

En la figura 7.5.4 se muestra el diagrama de transiciones del AFDC D_3 para el lenguaje de las palabras que terminan con la cadena bb. Por otro parte, en la figura 7.5.5 se muestra el diagrama de transiciones del AFD D_4 cuyo lenguaje asociado es el lenguaje de las palabras que contienen la cadena bb.

En la figura 7.5.6 se muestra el árbol de comparación de autómatas para D_3 y D_4 que construiría el algoritmo presentado en el apartado 7.5.1. Puesto que aparecen nodos formados por estados incompatibles, la conclusión final es que D_3 y D_4 no son equivalentes.

Para construir ese árbol, el algoritmo seguiría los siguientes pasos:

#1. Crear un árbol cuya raíz y único nodo es el par (t_0, m_0). Ir al punto #2.

#2. Puesto que los estados t_0 y m_0 que conforman la única hoja del árbol son compatibles, ir al punto #3.

#3. Como la única hoja (t_0, m_0) del árbol no está repetida en su rama, ir al punto #4.

#4. (t_0, m_0) es la única hoja que se tiene y la única que no está repetida en su rama. Por tanto, se seleccionará la hoja (t_0, m_0) y se añadirán los nodos (t_0, m_0), (t_1, m_1) y (t_0, m_0). Ahora se ha de volver al punto #2.

#2. Se tienen tres hojas: (t_0, m_0), (t_1, m_1) y (t_0, m_0). Ninguna de ellas está formada por estados incompatibles. Consecuentemente, se ha de ir al punto #3.

#3. Hay una hoja que no está repetida en su rama: (t_1, m_1). Las otras dos hojas están repetidas en sus respectivas ramas. Se ha de ir al punto #4.

#4. Solo hay una hoja seleccionable: (t_1, m_1). Por tanto, se selecciona esa hoja y se expande el árbol. Se generarán los pares (t_0, m_0), (t_2, m_2) y (t_0, m_0). A continuación se ha de ir al punto #2.

#2. Ahora se tienen cinco hojas: (t_0, m_0), (t_0, m_0), (t_2, m_2), (t_0, m_0) y (t_0, m_0). Como en las cinco hojas los estados que las conforman son compatibles, se ha de ir al punto #3.

#3. Hay al menos una hoja que no aparece repetida en su rama. En concreto, la única hoja que no está repetida en su correspondiente rama es (t_2, m_2). Por tanto, se ha de ir al punto #4.

#4. Solo hay una hoja seleccionable: (t_2, m_2). A partir de ese nodo se generarán los nodos (t_0, m_2), (t_2, m_2) y (t_0, m_2). A continuación se ha de ir al punto #2.

#2. Se tienen siete hojas: (t_0, m_0), (t_0, m_0), (t_0, m_2), (t_2, m_2), (t_0, m_2), (t_0, m_0) y (t_0, m_0). Dos de ellas están formadas por estados incompatibles: (t_0, m_2) y (t_0, m_2). Consecuentemente, se ha de terminar concluyendo que los AFD's D_3 y D_4 no son equivalentes.

En este ejemplo, cada vez que se ha pasado por el punto $\#4$, solo había una hoja seleccionable. Por tanto, el árbol mostrado en la figura 7.5.6 es el único árbol construible y, además, cada vez que se ha pasado por el punto $\#4$, el árbol (provisional) obtenido es único, es decir, no hay lugar para árboles alternativos.

7.5.2.3 Segundo ejemplo de dos AFDC no equivalentes: D_4 y D_5

En la figura 7.5.5 se muestra el diagrama de transiciones del AFDC D_4 para el lenguaje de las palabras que contienen la cadena bb. Por otra parte, en la figura 7.5.7 se muestra el diagrama de transiciones del AFDC D_5 cuyo lenguaje asociado es el lenguaje de las palabras que contienen la cadena cc.

Tal como se ha indicado en los ejemplos desarrollados en los apartados 7.5.2.1 y 7.5.2.2, cuando se tienen varios nodos que admiten expansión, el algoritmo para la construcción del árbol de comparación de autómatas no establece ningún orden. Por tanto, diferentes implementaciones del algoritmo pueden dar lugar a árboles distintos al comparar los mismos dos autómatas. En los ejemplos presentados en los apartados 7.5.2.1 y 7.5.2.2, los árboles finales serían, en cualquier caso, los presentados en la figura 7.5.3 y la figura 7.5.6, respectivamente. De todas formas, durante la construcción del árbol de la figura 7.5.3 se pueden obtener distintos árboles parciales dependiendo del nodo seleccionado en cada paso de expansión. Al aplicar el algoritmo de comparación a los AFDC D_4 y D_5, ocurre que dependiendo de las selecciones realizadas durante el proceso, el árbol final cambia. A continuación se ilustrará este hecho.

En la figura 7.5.8 y en la figura 7.5.9 se muestran dos árboles de comparación de autómatas para D_4 y D_5 construidos siguiendo el algoritmo presentado en el apartado 7.5.1. Las diferencias provienen de las selecciones realizadas durante el proceso de construcción. Puesto que en ambos aparece un nodo formado por estados incompatibles, la conclusión final es, en cualquier caso, que D_4 y D_5 no son equivalentes. Tan pronto como aparece un nodo formado por estados incompatibles, el algoritmo termina el proceso de construcción del árbol. Por ello, los árboles de las figuras 7.5.8 y 7.5.9 contienen hojas que no están repetidas en su rama y que están formadas por estados compatibles pero que se han quedado sin ser desarrolladas.

Para construir el árbol de la figura 7.5.8, el algoritmo seguiría los siguientes pasos:

- $\#1.$ Crear un árbol cuya raíz y único nodo es el par (m_0, p_0). Ir al punto $\#2$.
- $\#2.$ Puesto que los estados m_0 y p_0 que conforman la única hoja del árbol son compatibles, ir al punto $\#3$.
- $\#3.$ Como la única hoja (m_0, p_0) del árbol no está repetida en su rama, ir al punto $\#4$.
- $\#4.$ (m_0, p_0) es la única hoja que se tiene y la única que no está repetida en su rama. Por tanto, se seleccionará la hoja (m_0, p_0) y se añadirán los nodos (m_0, p_0), (m_1, p_0) y (m_0, p_1). Ahora se ha de volver al punto $\#2$.

$\#2.$ Se tienen tres hojas: (m_0, p_0), (m_1, p_0) y (m_0, p_1). Ninguna de ellas está formada por estados incompatibles. Consecuentemente, se ha de ir al punto $\#3$.

$\#3.$ Hay dos hojas que no están repetidas en su rama: (m_1, p_0) y (m_0, p_1). La otra hoja está repetida en su rama. Se ha de ir al punto $\#4$.

$\#4.$ Hay dos hojas seleccionables: (m_1, p_0) y (m_0, p_1). Puesto que el algoritmo no establece un criterio para seleccionar entre distintas hojas seleccionables, supongamos que se selecciona la hoja (m_1, p_0). En ese caso se generarán los nodos (m_0, p_0), (m_2, p_0) y (m_0, p_1). A continuación se ha de ir al punto $\#2$.

$\#2.$ Ahora se tienen cinco hojas: (m_0, p_0), (m_0, p_0), (m_2, p_0), (m_0, p_1) y (m_0, p_1). Puesto que en la hoja (m_2, p_0) los estados que la conforman son incompatibles, se ha de terminar concluyendo que los AFDC D_4 y D_5 no son equivalentes.

Tal como se puede observar, las dos últimas hojas, es decir, (m_0, p_1) y (m_0, p_1), ni están formadas por estados incompatibles ni están repetidas en sus respectivas ramas, pero se han quedado sin ser desarrolladas porque se ha terminado el proceso de construcción del árbol.

La construcción del árbol de la figura 7.5.9 es similar. De hecho, todo el proceso es igual hasta llegar al punto $\#4$ por segunda vez. Una vez llegado al punto $\#4$ por segunda vez, cambia el nodo que se ha seleccionado:

$\#1.$ Crear un árbol cuya raíz y único nodo es el par (m_0, p_0). Ir al punto $\#2$.

$\#2.$ Puesto que los estados m_0 y p_0 que conforman la única hoja del árbol son compatibles, ir al punto $\#3$.

$\#3.$ Como la única hoja (m_0, p_0) del árbol no está repetida en su rama, ir al punto $\#4$.

$\#4.$ (m_0, p_0) es la única hoja que se tiene y la única que no está repetida en su rama. Por tanto, se seleccionará la hoja (m_0, p_0) y se añadirán los nodos (m_0, p_0), (m_1, p_0) y (m_0, p_1). Ahora se ha de volver al punto $\#2$.

$\#2.$ Se tienen tres hojas: (m_0, p_0), (m_1, p_0) y (m_0, p_1). Ninguna de ellas está formada por estados incompatibles. Consecuentemente, se ha de ir al punto $\#3$.

$\#3.$ Hay dos hojas que no están repetidas en su rama: (m_1, p_0) y (m_0, p_1). La otra hoja está repetida en su rama. Se ha de ir al punto $\#4$.

$\#4.$ Hay dos hojas seleccionables: (m_1, p_0) y (m_0, p_1). Debido a que el algoritmo no establece un criterio para seleccionar entre distintas hojas seleccionables, supongamos que se selecciona la hoja (m_0, p_1). En ese caso se generarán los nodos (m_0, p_0), (m_1, p_0) y (m_0, p_2). A continuación se ha de ir al punto $\#2$.

$\#2.$ Ahora se tienen cinco hojas: (m_0, p_0), (m_1, p_0), (m_0, p_0), (m_1, p_0) y (m_0, p_2). Puesto que en la hoja (m_0, p_2) los estados que la conforman son incompatibles, se ha de terminar concluyendo que los AFDC D_4 y D_5 no son equivalentes.

En este caso, son las hojas (m_1, p_0) y (m_1, p_0) las que ni están formadas por estados incompatibles ni están repetidas en sus respectivas ramas, pero se han quedado sin ser desarrolladas porque se ha terminado el proceso de construcción del árbol.

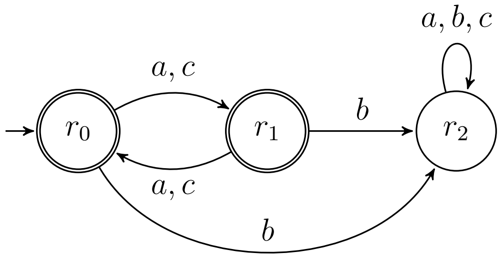


Figura 7.5.1. AFDC D_1 para el lenguaje de las palabras que no contienen el símbolo b.

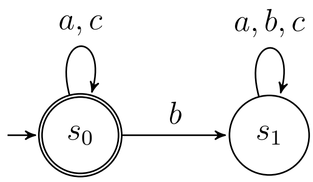


Figura 7.5.2. AFDC D_2 para el lenguaje de las palabras que no contienen el símbolo b.

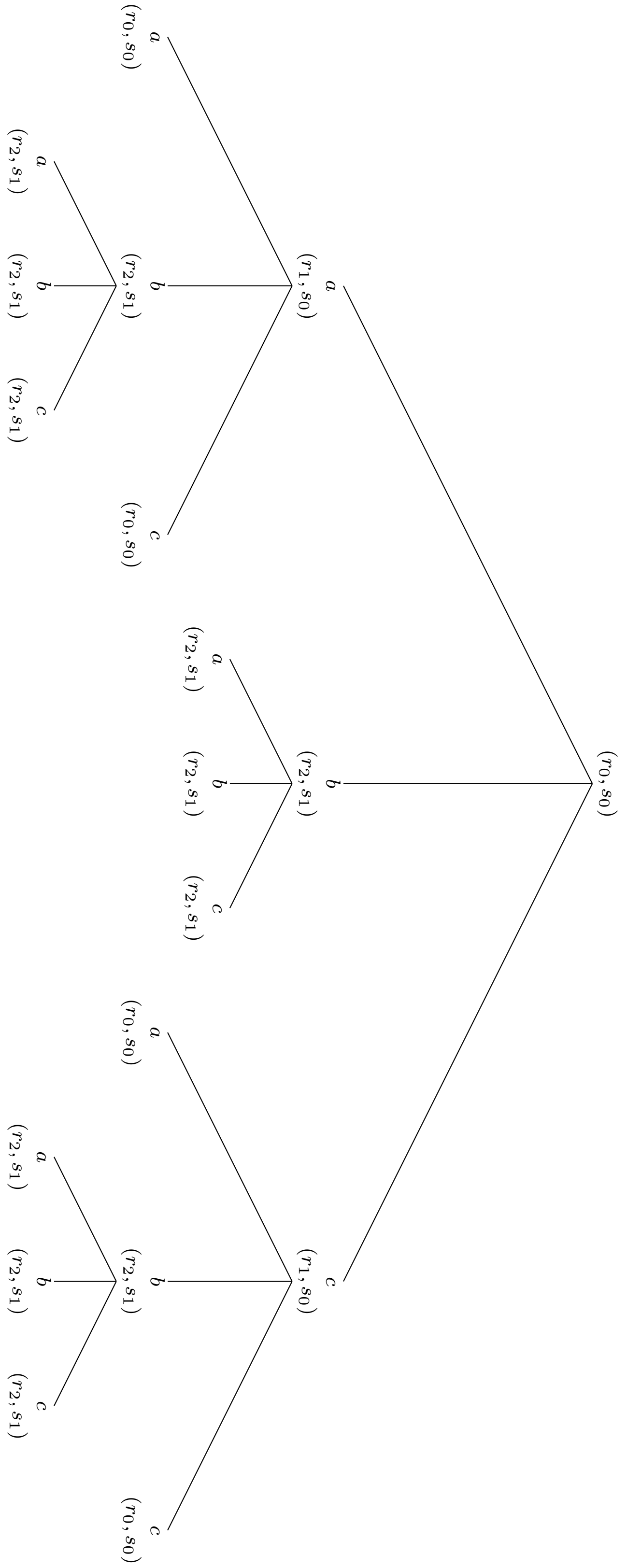


Figura 7.5.3. Árbol de comparación de autómatas para los AFDC D_1 y D_2. Todas las hojas están repetidas en su respectiva rama.

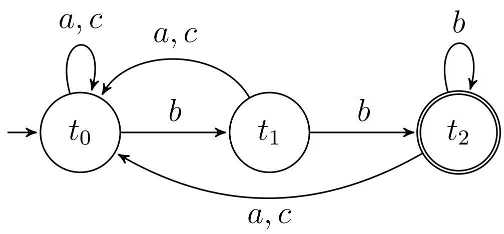


Figura 7.5.4. AFDC D_3 para el lenguaje de las palabras que terminan con la cadena bb.

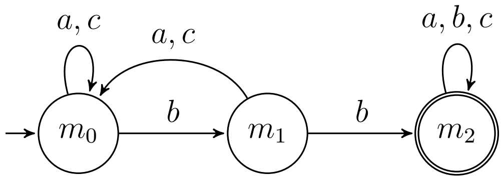


Figura 7.5.5. AFDC D_4 para el lenguaje de las palabras que contienen la cadena bb.

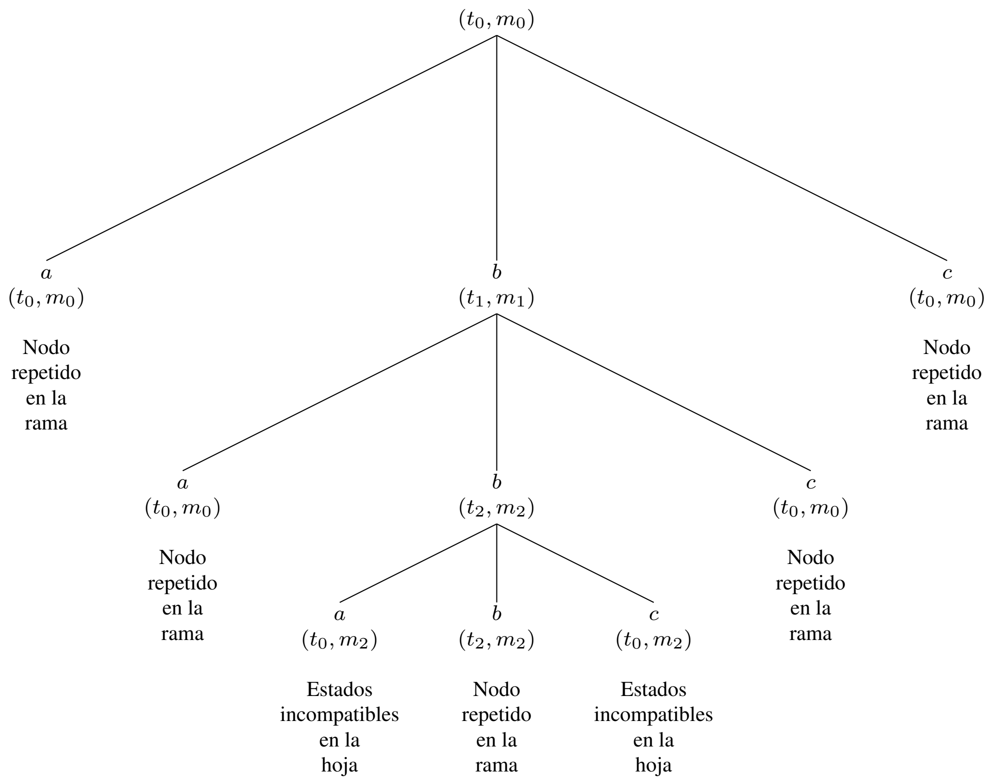


Figura 7.5.6. Árbol de comparación de autómatas para los AFDC D_3 y D_4.

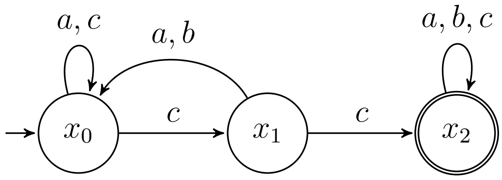


Figura 7.5.7. AFD D_5 para el lenguaje de las palabras que contienen la cadena cc.

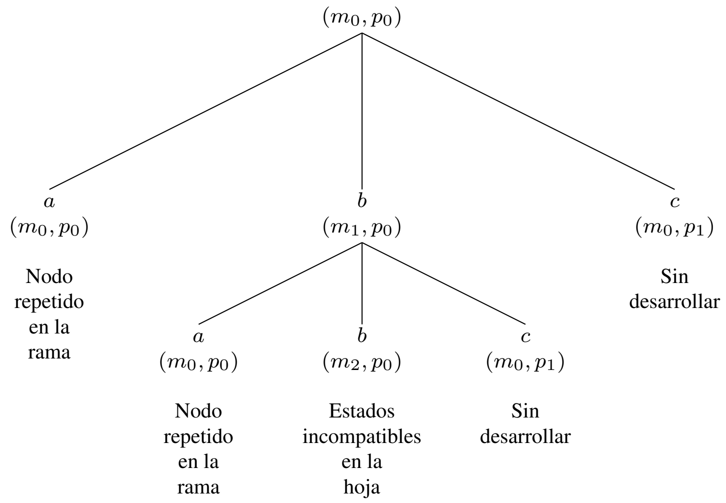


Figura 7.5.8. Un árbol de comparación de autómatas para los AFDC D_4 y D_5.

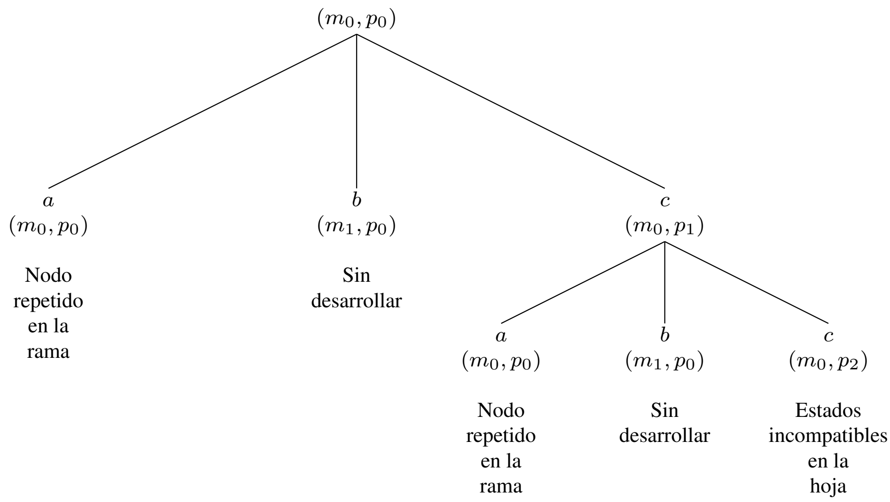


Figura 7.5.9. Otro árbol de comparación de autómatas para los AFDC D_4 y D_5.

7.6.

Minimización de los AFDC sin transiciones λ

En general, para cada lenguaje es posible diseñar distintos AFDC que varían en el número de estados que contienen. En este apartado, se presenta un método para minimizar un AF determinista completo (AFDC) sin transiciones λ. Se obtendrá otro AF determinista completo (AFDC) sin transiciones λ. El nuevo AFDC tendrá el menor número de estados posible. Es decir, no existirá ningún otro AFDC que defina el mismo lenguaje y tenga menos estados. Este método detecta y elimina los estados innecesarios.

En el resto de este apartado, todo AF será sin transiciones λ.

7.6.1 Descripción del AFDC minimal correspondiente a un AFDC

Dado un AFND $\mathcal{F}$ de la forma $(Q, \mathbb{A}, \tau_{\mathbb{A}}, \tau_\lambda, \sigma, Y)$ tal que $\tau_\lambda = \varnothing$, se obtendrá un AFDC $\mathcal{F}'$ de la forma $(Q', \mathbb{A}', \tau'_{\mathbb{A}'}, \tau'_\lambda, \sigma', Y')$ donde:

- $Q' \subseteq 2^Q$ y, además, Q' representa una partición disjunta de Q. [1]
- $\mathbb{A}' = \mathbb{A}$
- $\tau'_\lambda = \varnothing$
- $\sigma' \in Q'$ y, además, $\sigma \in \sigma'$.
- $Y' \subseteq Q'$ y, además, Y' representa una partición disjunta de Y.

[1] Dado un conjunto C, una partición disjunta de C lo forman una serie de conjuntos no vacíos $C_1, C_2, \ldots, C_n$ donde $n \geq 1$, $C = (C_1 \cup C_2 \cup \cdots \cup C_n)$ y $C_i \cap C_h = \varnothing$ para cualquier i y h tal que $i \neq h$, $i \in \{1, \ldots, n\}$ y $h \in \{1, \ldots, n\}$.

Por tanto, en el nuevo AF $\mathcal{F}'$ no habrá transiciones vacías, el alfabeto será el mismo y el conjunto de estados Q' estará formado por elementos que constituyen una partición disjunta de Q. Por consiguiente, cada estado de Q' es un conjunto formado por estados de Q. Por otra parte, el estado inicial será el elemento de Q' que contenga σ y, finalmente, el conjunto Y' estará formado por elementos que constituyen una partición disjunta de Y. Consecuentemente, cada estado de Y' es un conjunto formado por estados de Y.

En cuanto a $\tau'_{\mathbb{A}'}$, lo primero es tener en cuenta que por cada $\varrho \in Q'$ y cada $\alpha \in \mathbb{A}$, tiene que existir una transición de la forma $(\varrho, \alpha, \varrho')$ donde ϱ' es un elemento de Q'. Además, solo puede haber una única transición con ϱ y α como primeros dos elementos. Recordemos que se quiere diseñar un AFDC.

El elemento ϱ' correspondiente a un estado ϱ y un símbolo α es el siguiente:

$$\varrho' = \{q_h \mid (q_h \in Q) \wedge \exists q_i(q_i \in \varrho \wedge (q_i, \alpha, q_h) \in \tau_{\mathbb{A}})\}$$

Es decir, ϱ' estará formado por todos aquellos estados que son accesibles en un solo paso con α desde algún estado de ϱ.

7.6.2 Construcción del AFDC minimal correspondiente a un AFDC

El objetivo es identificar los estados que son equivalentes entre ellos y mantener solo un estado de cada grupo de estados equivalentes. Para identificar los estados que son equivalentes entre ellos, se lleva a cabo un proceso iterativo en el que, en cada momento se tiene una propuesta de estados que podrían ser equivalentes. Esa propuesta consiste en una partición de los estados de Q. Por tanto, cada propuesta agrupa los estados de Q en varios grupos disjuntos (los grupos no pueden compartir elementos). Ante una propuesta, se analiza el comportamiento de los componentes de cada grupo y se decide si hay que partir el grupo en subgrupos. Los elementos que pertenezcan a un grupo pero muestren un comportamiento distinto, han de pertenecer a grupos distintos y eso se consigue partiendo el grupo en subgrupos formados por elementos que muestran el mismo comportamiento.

Puesto que $\mathbb{A}'$ es $\mathbb{A}$, el conjunto τ'_λ es vacío y σ' es el conjunto unitario $\{\sigma\}$, la construcción del nuevo AFDC se centra en el cálculo de $\tau'_{\mathbb{A}'}$, Q' eY'.

7.6.2.1 Cálculo de Q'

Se calculará Q' de manera progresiva.

- **Inicialización de Q':**
  Para empezar, a modo de primera propuesta, Q' contendrá dos estados: Uno de ellos estará formado por todos los estados pertenecientes a Y; el otro estará formado por todos

los estados no pertenecientes a Y, es decir, $Q \setminus Y$. Como cada estado de Q' es un conjunto de estados de Q, por cada estado de Q' se nombrará un representante. Por ejemplo, si un estado ρ de Q' es el conjunto $\{q_3, q_7, q_8\}$, podríamos elegir a q_3 como representante y expresar el conjunto ρ como $[q_3]$.

- **Actualización de Q':**
  Para cada estado ρ de Q' se han de analizar las transiciones de los elementos que conforman ρ para averiguar si todos coinciden teniendo en cuenta los grupos de Q'. Si en ρ hay elementos que muestran distintos comportamientos entre ellos, habrá que generar un subgrupo por cada uno de esos comportamiento distintos. Esto supondrá partir ρ en subgrupos disjuntos. Los subgrupos generados son los nuevos estados con los que se actualiza Q'.

  El proceso de actualización se ha de repetir hasta que se cumpla la condición para finalizar el proceso.

- **Finalización del cálculo de Q':**
  Si para todo estado ρ de Q' no surge ninguna partición, se termina el proceso de cálculo de Q' y de $\tau'_{\mathbb{A}'}$.

El conjunto Q' será una partición disjunta de Q y para cada estado ρ de Q', todos los elementos de ρ tendrán el mismo comportamiento teniendo en cuenta los grupos de Q'.

7.6.2.2 Cálculo de $\tau'_{\mathbb{A}'}$

Se calculará $\tau'_{\mathbb{A}'}$ una vez que se haya calculado Q'.

Por cada estado $\rho \in Q'$ y cada símbolo $\alpha \in \mathbb{A}$, se tiene que $(\rho, \alpha, \rho') \in \tau'_{\mathbb{A}'}$ si y solamente si, para el representante q_i de ρ se cumple que $(q_i, \alpha, q_h) \in \tau_{\mathbb{A}}$ y q_h es un estado de ρ'.

7.6.2.3 Cálculo de Y'

Una vez calculado el conjunto Q' final, el conjunto Y' estará formado por los elementos de Q' que contengan solo elementos de Y.

7.6.3 Ejemplos de cálculo del AFDC minimal correspondiente a un AFDC

7.6.3.1 Ejemplo 1 minimización de AFDC

7.6.3.1.1 Ejemplo 1: AFDC original

En la figura 7.6.1 de la página 76, se muestra un AFDC $(Q, \mathbb{A}, \tau_{\mathbb{A}}, \tau_{\lambda}, \sigma, Y)$.

En ese AFDC, los componentes son los siguientes:

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$
- $\mathbb{A} = \{a, b\}$
- $\tau_{\mathbb{A}} = \begin{array}{l}\{(q_0, a, q_1), (q_0, b, q_2), (q_1, a, q_3), (q_1, b, q_4), (q_2, a, q_5), (q_2, b, q_6), (q_3, a, q_3), \\ (q_3, b, q_4), (q_4, a, q_5), (q_4, b, q_6), (q_5, a, q_3), (q_5, b, q_4), (q_6, a, q_5), (q_6, b, q_6)\}\end{array}$
- $\tau_{\lambda} = \varnothing$
- $\sigma = q_0$
- $Y = \{q_5\}$

7.6.3.1.2 Ejemplo 1: Pasos para construir un AFDC minimal

El objetivo es construir un AFDC $(Q', \mathbb{A}', \tau'_{\mathbb{A}'}, \tau'_{\lambda}, \sigma', Y')$ que defina el mismo lenguaje pero que sea minimal.

Para empezar, sabemos que $\mathbb{A}' = \mathbb{A}$ y que $\tau'_{\lambda} = \varnothing$.

A continuación se mostrarán los pasos a seguir para minimizar el AFDC de este ejemplo. El proceso consiste en agrupar los estados que tienen siempre el mismo comportamiento. Si un conjunto de estados se comportan siempre igual, es posible dejar solo uno de ellos y eliminar el resto. De esta manera, aseguramos que los estados que quedan son necesarios porque todos tienen un comportamiento distinto, es decir, no hay estados repetidos.

Hay que tener en cuenta que cada estado de un AFDC desempeña el papel de memoria. Cada estado recuerda una condición o propiedad. Por tanto, es innecesario tener más de un estado que recuerde una misma propiedad. El método de minimización mantiene solo un estado de entre aquellos que recuerdan la misma propiedad.

Primera partición La primera partición consiste en generar un conjunto con los estados que pertenecen a Y (los estados de aceptación) y otro con los que no pertenecen a Y. A cada conjunto lo llamaremos $[q_j]$ donde j es el menor de los índices que aparecen en el conjunto en cuestión.

$$[q_0] = \{q_0, q_1, q_2, q_3, q_4, q_6\}$$
$$[q_5] = \{q_5\}$$

Segunda partición Recordemos que la tabla correspondiente a la función de transición $\tau_{\mathbb{A}}$ es la siguiente:

$\tau_{\mathbb{A}}$	a	b
q_0	q_1	q_2
q_1	q_3	q_4
q_2	q_5	q_6
q_3	q_3	q_4
q_4	q_5	q_6
q_5	q_3	q_4
q_6	q_5	q_6

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo un a se pasa al estado q_3, en vez de q_3 vamos a poner su grupo, es decir, $[q_0]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_0]$	$[q_0]$
q_1	$[q_0]$	$[q_0]$
q_2	$[q_5]$	$[q_0]$
q_3	$[q_0]$	$[q_0]$
q_4	$[q_5]$	$[q_0]$
q_5	$[q_0]$	$[q_0]$
q_6	$[q_5]$	$[q_0]$

Ahora vamos a ver si hay que partir los conjuntos $[q_0]$ y $[q_5]$. Obviamente, el conjunto $[q_5]$ no se puede partir porque contiene un único elemento, así que nos centramos en el conjunto $[q_0]$.

En el conjunto $[q_0]$ vemos que los estados q_0, q_1 y q_3 tienen el mismo comportamiento: Cuando se lee un a van a un estado que pertenece al conjunto $[q_0]$ y cuando se lee un b van a un estado que pertenece al conjunto $[q_0]$. En cambio, los estados q_2, q_4 y q_6 tienen otro comportamiento: Cuando se lee un a van a un estado que pertenece al conjunto $[q_5]$ y cuando se lee un b van a un estado que pertenece al conjunto $[q_0]$. Por tanto vamos a partir el conjunto $[q_0]$ teniendo en cuenta esos dos comportamientos y nos quedan tres conjuntos:

$$[q_0] = \{q_0, q_1, q_3\}$$
$$[q_2] = \{q_2, q_4, q_6\}$$
$$[q_5] = \{q_5\}$$

IMPORTANTE: Es importantante darse cuenta de que aunque q_5 tenga el mismo comportamiento que los estados de $[q_0]$, no hay que meterlos en el mismo conjunto. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar.

No hay tercera partición Ahora vamos a recalcular la tabla de transiciones a conjuntos, teniendo en cuenta la segunda partición:

	a	b
q_0	$[q_0]$	$[q_2]$
q_1	$[q_0]$	$[q_2]$
q_2	$[q_5]$	$[q_2]$
q_3	$[q_0]$	$[q_2]$
q_4	$[q_5]$	$[q_2]$
q_5	$[q_0]$	$[q_2]$
q_6	$[q_5]$	$[q_2]$

Ahora vamos a ver si hay que partir los conjuntos $[q_0]$ y $[q_2]$. Como se ha dicho antes, el conjunto $[q_5]$ no se puede partir porque contiene un único elemento, así que nos centramos en los conjuntos $[q_0]$ y $[q_2]$.

En el conjunto $[q_0]$ vemos que los estados q_0, q_1 y q_3 tienen el mismo comportamiento. Cuando se lee un a van a un estado que pertenece al conjunto $[q_0]$ y cuando se lee un b van a un estado que pertenece al conjunto $[q_2]$. Por tanto, el conjunto $[q_0]$ no se ha de partir. En el conjunto $[q_2]$ ocurre algo parecido. Los estados q_2, q_4 y q_6 tienen el mismo comportamiento. Cuando se lee un a van a un estado que pertenece al conjunto $[q_5]$ y cuando se lee un b van a un estado que pertenece al conjunto $[q_2]$. Por tanto, el conjunto $[q_2]$ no se ha de partir.

IMPORTANTE: Es importantante darse cuenta de que aunque q_5 tenga el mismo comportamiento que los estados de $[q_0]$, no hay que meterlos en el mismo conjunto. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar.

Por tanto, los conjuntos son estos tres:

$$
\begin{aligned}
[q_0] &= \{q_0, q_1, q_3\} \\
[q_2] &= \{q_2, q_4, q_6\} \\
[q_5] &= \{q_5\}
\end{aligned}
$$

Cada conjunto representa un estado. Teniendo en cuenta la última tabla, se calculan las transiciones:

$\tau'_{\mathbb{A}'}$	a	b
q_0	$[q_0]$	$[q_2]$
q_2	$[q_5]$	$[q_2]$
q_5	$[q_0]$	$[q_2]$

Cálculo de σ' $[q_0]$ será el estado inicial σ' porque contiene a q_0, que es el estado inicial del AFDC original.

Cálculo de Y' En este caso solo el conjunto $[q_5]$ está formado por estados de Y. Consecuentemente, $Y' = \{[q_5]\}$.

7.6.3.1.3 Ejemplo 1: AFDC minimizado

En la figura 7.6.2 de la página 76, se muestra el AFDC minimal obtenido.

7.6.3.2 Ejemplo 2 minimización de AFDC

7.6.3.2.1 Ejemplo 2: AFDC original

En la figura 7.6.3 de la página 79, se muestra un AFDC $(Q, \mathbb{A}, \tau_{\mathbb{A}}, \tau_{\lambda}, \sigma, Y)$.

En ese AFDC, los componentes son los siguientes:

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\mathbb{A} = \{a, b, c\}$
- $\begin{aligned} \tau_{\mathbb{A}} = \ & \{(q_0, a, q_1), (q_0, b, q_2), (q_0, c, q_3), (q_1, a, q_1), (q_1, b, q_2), (q_1, c, q_3), \\ & (q_2, a, q_2), (q_2, b, q_2), (q_2, c, q_3), (q_3, a, q_3), (q_3, b, q_2), (q_3, c, q_3)\} \end{aligned}$
- $\tau_{\lambda} = \varnothing$
- $\sigma = q_0$
- $Y = \{q_0, q_1\}$

7.6.3.2.2 Ejemplo 2: Pasos para construir un AFDC minimal

El objetivo es construir un AFDC $(Q', \mathbb{A}', \tau'_{\mathbb{A}'}, \tau'_{\lambda}, \sigma', Y')$ que defina el mismo lenguaje pero que sea minimal.

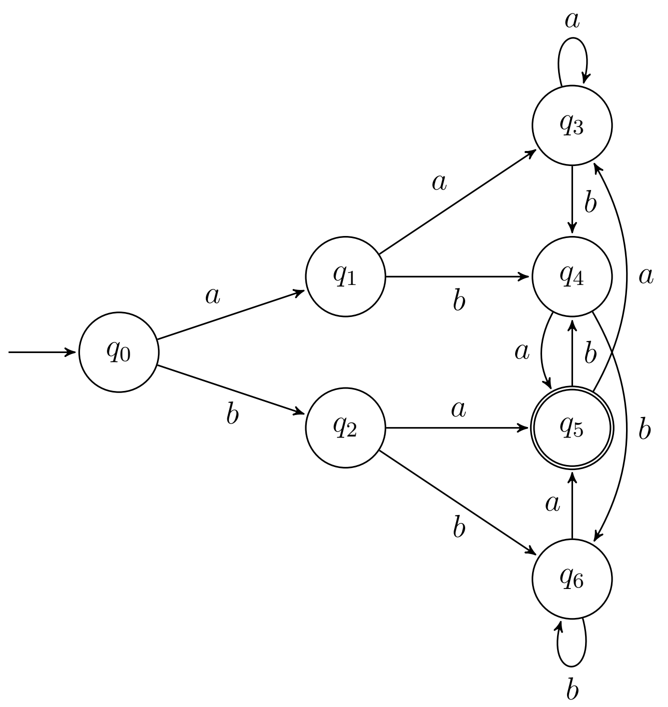


Figura 7.6.1. Minimización: ejemplo 1. AFDC de partida.

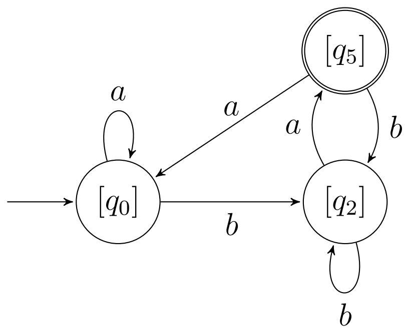


Figura 7.6.2. Minimización: ejemplo 1. AFDC minimal correspondiente al AFDC de la figura 7.6.1 (página 76).

Para empezar, sabemos que $\mathbb{A}' = \mathbb{A}$ y que $\tau'_\lambda = \varnothing$.

A continuación se mostrarán los pasos a seguir para minimizar el AFDC de este ejemplo. El proceso consiste en agrupar los estados que tienen siempre el mismo comportamiento. Si un conjunto de estados se comportan siempre igual, es posible dejar solo uno de ellos y eliminar el resto. De esta manera, aseguramos que los estados que quedan son necesarios porque todos tienen un comportamiento distinto, es decir, no hay estados repetidos.

Hay que tener en cuenta que cada estado de un AFDC desempeña el papel de memoria. Cada estado recuerda una condición o propiedad. Por tanto, es innecesario tener más de un estado que recuerde una misma propiedad. El método de minimización mantiene solo un estado de entre aquellos que recuerdan la misma propiedad.

Primera partición La primera partición consiste en generar un conjunto con los estados que pertenecen a Y (los estados de aceptación) y otro con los que no pertenecen a Y. A cada conjunto lo llamaremos $[q_j]$ donde j es el menor de los índices que aparecen en el conjunto en cuestión.

$$[q_0] = \{q_0, q_1\}$$
$$[q_2] = \{q_2, q_3\}$$

No hay segunda partición Recordemos que la tabla correspondiente a la función de transición $\tau_{\mathbb{A}}$ es la siguiente:

$\tau_{\mathbb{A}}$	a	b	c
q_0	q_1	q_2	q_3
q_1	q_1	q_2	q_3
q_2	q_2	q_2	q_3
q_3	q_3	q_2	q_3

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo una a se pasa al estado q_1, en vez de q_1 vamos a poner su grupo, es decir, $[q_0]$. Procederemos de la misma forma en todos los casos:

	a	b	c
q_0	$[q_0]$	$[q_2]$	$[q_2]$
q_1	$[q_0]$	$[q_2]$	$[q_2]$
q_2	$[q_2]$	$[q_2]$	$[q_2]$
q_3	$[q_2]$	$[q_2]$	$[q_2]$

Ahora vamos a ver si hay que partir los conjuntos $[q_0]$ y $[q_2]$.

En el conjunto $[q_0]$ vemos que los estados q_0 y q_1 tienen el mismo comportamiento: Cuando se lee una a van a un estado que pertenece al conjunto $[q_0]$, cuando se lee una b van a un estado que pertenece al conjunto $[q_2]$ y cuando se lee una c van a un estado que pertenece al conjunto $[q_2]$. Por tanto, no hay que partir el conjunto $[q_0]$.

En el conjunto $[q_2]$ vemos que los estados q_2 y q_3 tienen el mismo comportamiento: Cuando se lee una a van a un estado que pertenece al conjunto $[q_2]$, cuando se lee una b van a un estado que pertenece al conjunto $[q_2]$ y cuando se lee una c van a un estado que pertenece al conjunto $[q_2]$. Por tanto, no hay que partir el conjunto $[q_2]$.

Por tanto, los conjuntos son estos dos:

$$[q_0] = \{q_0, q_1\}$$
$$[q_2] = \{q_2, q_3\}$$

Cada conjunto representa un estado. Teniendo en cuenta la última tabla, se calculan las transiciones:

$\tau'_{\mathbb{A}'}$	a	b	c
q_0	$[q_0]$	$[q_2]$	$[q_2]$
q_2	$[q_2]$	$[q_2]$	$[q_2]$

Cálculo de σ' $[q_0]$ será el estado inicial σ' porque contiene a q_0, que es el estado inicial del AFDC original.

Cálculo de Y' En este caso solo el conjunto $[q_0]$ está formado por estados de Y. Consecuentemente, $Y' = \{[q_0]\}$.

7.6.3.2.3 Ejemplo 2: AFDC minimizado

En la figura 7.6.4 de la página 79, se muestra el AFDC minimal obtenido.

7.6.3.3 Ejemplo 3 minimización de AFDC

7.6.3.3.1 Ejemplo 3: AFDC original

En la figura 7.6.5 de la página 83, se muestra un AFDC $(Q, \mathbb{A}, \tau_{\mathbb{A}}, \tau_\lambda, \sigma, Y)$.

En ese AFDC, los componentes son los siguientes:

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
- $\mathbb{A} = \{a, b\}$
- $\tau_{\mathbb{A}} = \{(q_0, a, q_1), (q_0, b, q_2), (q_1, a, q_1), (q_1, b, q_3), (q_2, a, q_4), (q_2, b, q_2), (q_3, a, q_5), (q_3, b, q_3), (q_4, a, q_4), (q_4, b, q_4), (q_5, a, q_5), (q_5, b, q_5)\}$

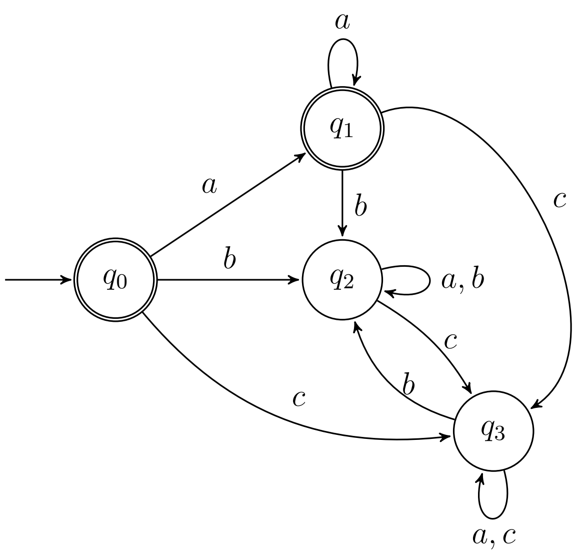


Figura 7.6.3. Minimización: ejemplo 2. AFDC de partida.

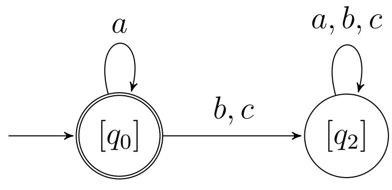


Figura 7.6.4. Minimización: ejemplo 2. AFDC minimal correspondiente al AFDC de la figura 7.6.3 (página 79).

- $\tau_\lambda = \varnothing$
- $\sigma = q_0$
- $Y = \{q_0, q_1, q_2, q_3\}$

7.6.3.3.2 Ejemplo 3: Pasos para construir un AFDC minimal

El objetivo es construir un AFDC $(Q', \mathbb{A}', \tau'_{\mathbb{A}'}, \tau'_\lambda, \sigma', Y')$ que defina el mismo lenguaje pero que sea minimal.

Para empezar, sabemos que $\mathbb{A}' = \mathbb{A}$ y que $\tau'_\lambda = \varnothing$.

A continuación se mostrarán los pasos a seguir para minimizar el AFDC de este ejemplo. El proceso consiste en agrupar los estados que tienen siempre el mismo comportamiento. Si un conjunto de estados se comportan siempre igual, es posible dejar solo uno de ellos y eliminar el resto. De esta manera, aseguramos que los estados que quedan son necesarios porque todos tienen un comportamiento distinto, es decir, no hay estados repetidos.

Hay que tener en cuenta que cada estado de un AFDC desempeña el papel de memoria. Cada estado recuerda una condición o propiedad. Por tanto, es innecesario tener más de un estado que recuerde una misma propiedad. El método de minimización mantiene solo un estado de entre aquellos que recuerdan la misma propiedad.

Primera partición La primera partición consiste en generar un conjunto con los estados que pertenecen a Y (los que llevan doble círculo) y otro con los que no pertenecen a Y. A cada conjunto lo llamaremos $[q_j]$ donde j es el menor de los índices que aparecen en el conjunto en cuestión.

$$[q_0] = \{q_0, q_1, q_2, q_3\}$$
$$[q_4] = \{q_4, q_5\}$$

Segunda partición Recordemos que la tabla correspondiente a la función de transición $\tau_\mathbb{A}$ es la siguiente:

$\tau_\mathbb{A}$	a	b
q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_4	q_2
q_3	q_5	q_3
q_4	q_4	q_4
q_5	q_5	q_5

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo una b se pasa al estado q_3, en vez de q_3 vamos a poner su grupo, es decir, $[q_0]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_0]$	$[q_0]$
q_1	$[q_0]$	$[q_0]$
q_2	$[q_4]$	$[q_0]$
q_3	$[q_4]$	$[q_0]$
q_4	$[q_4]$	$[q_4]$
q_5	$[q_4]$	$[q_4]$

Ahora vamos a ver si hay que partir los conjuntos $[q_0]$ y $[q_4]$.

En el conjunto $[q_0]$ vemos que los estados q_0 y q_1 tienen el mismo comportamiento: Cuando se lee una a van a un estado que pertenece al conjunto $[q_0]$ y cuando se lee una b van a un estado que pertenece al conjunto $[q_0]$. En cambio, los estados q_2 y q_3 tienen otro comportamiento: Cuando se lee una a van a un estado que pertenece al conjunto $[q_4]$ y cuando se lee una b van a un estado que pertenece al conjunto $[q_0]$. Por tanto, vamos a partir el conjunto $[q_0]$ teniendo en cuenta esos dos comportamientos:

$$[q_0] = \{q_0, q_1\}$$
$$[q_2] = \{q_2, q_3\}$$

Por otra parte, en el conjunto $[q_4]$ vemos que los estados q_4 y q_5 tienen el mismo comportamiento: Cuando se lee una a van a un estado que pertenece al conjunto $[q_4]$ y cuando se lee una b van a un estado que pertenece al conjunto $[q_4]$. Por tanto, no hay que partir el conjunto $[q_4]$.

En total nos quedan tres conjuntos tras la segunda partición:

$$[q_0] = \{q_0, q_1\}$$
$$[q_2] = \{q_2, q_3\}$$
$$[q_4] = \{q_4, q_5\}$$

No hay tercera partición Ahora vamos a recalcular la tabla de transiciones a conjuntos, teniendo en cuenta la segunda partición:

	a	b
q_0	$[q_0]$	$[q_2]$
q_1	$[q_0]$	$[q_2]$
q_2	$[q_4]$	$[q_2]$
q_3	$[q_4]$	$[q_2]$
q_4	$[q_4]$	$[q_4]$
q_5	$[q_4]$	$[q_4]$

Ahora vamos a ver si hay que partir los conjuntos $[q_0]$, $[q_2]$ y $[q_4]$.

En el conjunto $[q_0]$ vemos que los estados q_0 y q_1 tienen el mismo comportamiento: Cuando se lee una a van a un estado que pertenece al conjunto $[q_0]$ y cuando se lee una b van a un estado que pertenece al conjunto $[q_0]$. Por tanto, el conjunto $[q_0]$ no se ha de partir. En el conjunto $[q_2]$ ocurre algo parecido. Los estados q_2 y q_3 tienen el mismo comportamiento: Cuando se lee una a van a un estado que pertenece al conjunto $[q_4]$ y cuando se lee una b van a un estado que pertenece al conjunto $[q_2]$. Por tanto, el conjunto $[q_2]$ no se ha de partir. Tampoco se ha partir el conjunto $[q_4]$ porque los estados q_4 y q_5 tienen el mismo comportamiento: Cuando se lee una a van a un estado que pertenece al conjunto $[q_4]$ y cuando se lee una b van a un estado que pertenece al conjunto $[q_4]$.

Por tanto, los conjuntos son estos tres:

$$[q_0] = \{q_0, q_1\}$$
$$[q_2] = \{q_2, q_3\}$$
$$[q_5] = \{q_4, q_5\}$$

Cada conjunto representa un estado. Teniendo en cuenta la última tabla, se calculan las transiciones:

$\tau'_{\mathbb{A}'}$	a	b
q_0	$[q_0]$	$[q_2]$
q_2	$[q_4]$	$[q_2]$
q_4	$[q_4]$	$[q_4]$

Cálculo de σ' $[q_0]$ será el estado inicial σ' porque contiene a q_0, que es el estado inicial del AFDC original.

Cálculo de Y' En este caso, los conjuntos $[q_0]$ y $[q_2]$ están formados por estados de Y. Consecuentemente, $Y' = \{[q_0], [q_2]\}$.

7.6.3.3.3 Ejemplo 3: AFDC minimizado

En la figura 7.6.6 de la página 83, se muestra el AFDC minimal obtenido.

7.6.3.4 Ejemplo 4 minimización de AFDC

7.6.3.4.1 Ejemplo 4: AFDC original

En la figura 7.6.7 de la página 87, se muestra un AFDC $(Q, \mathbb{A}, \tau_{\mathbb{A}}, \tau_{\lambda}, \sigma, Y)$.

En ese AFDC, los componentes son los siguientes:

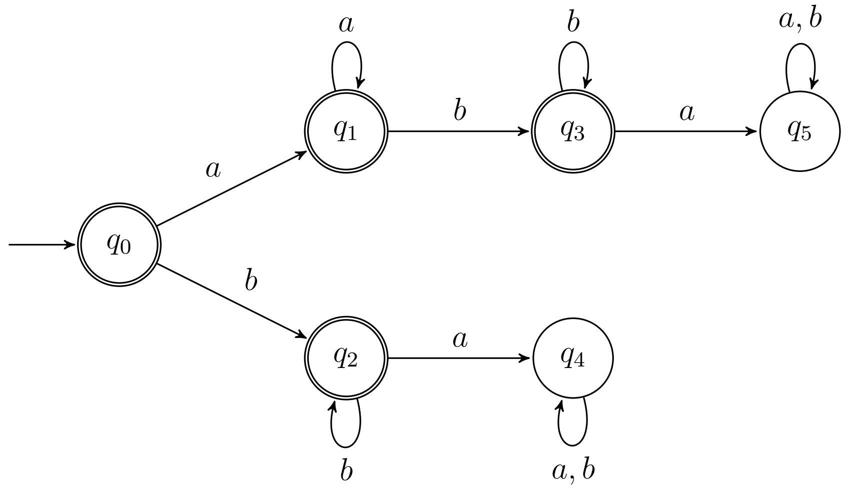


Figura 7.6.5. Minimización: ejemplo 3. AFDC de partida.

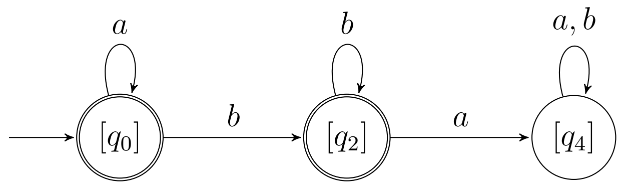


Figura 7.6.6. Minimización: ejemplo 3. AFDC minimal correspondiente al AFDC de la figura 7.6.5 (página 83).

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\mathbb{A} = \{a, b, c\}$
- $\tau_{\mathbb{A}} = \{(q_0, a, q_1), (q_0, b, q_2), (q_0, c, q_0), (q_1, a, q_0), (q_1, b, q_3), (q_1, c, q_1), (q_2, a, q_3), (q_2, b, q_0), (q_2, c, q_2), (q_3, a, q_2), (q_3, b, q_1), (q_3, c, q_3)\}$
- $\tau_\lambda = \varnothing$
- $\sigma = q_0$
- $Y = \{q_0\}$

7.6.3.4.2 Ejemplo 4: Pasos para construir un AFDC minimal

El objetivo es construir un AFDC $(Q', \mathbb{A}', \tau'_{\mathbb{A}'}, \tau'_\lambda, \sigma', Y')$ que defina el mismo lenguaje pero que sea minimal.

Para empezar, sabemos que $\mathbb{A}' = \mathbb{A}$ y que $\tau'_\lambda = \varnothing$.

A continuación se mostrarán los pasos a seguir para minimizar el AFDC de este ejemplo. El proceso consiste en agrupar los estados que tienen siempre el mismo comportamiento. Si un conjunto de estados se comportan siempre igual, es posible dejar solo uno de ellos y eliminar el resto. De esta manera, aseguramos que los estados que quedan son necesarios porque todos tienen un comportamiento distinto, es decir, no hay estados repetidos.

Hay que tener en cuenta que cada estado de un AFDC desempeña el papel de memoria. Cada estado recuerda una condición o propiedad. Por tanto, es innecesario tener más de un estado que recuerde una misma propiedad. El método de minimización mantiene solo un estado de entre aquellos que recuerdan la misma propiedad.

Primera partición La primera partición consiste en generar un conjunto con los estados que pertenecen a Y (los estados de aceptación) y otro con los que no pertenecen a Y. A cada conjunto lo llamaremos $[q_j]$ donde j es el menor de los índices que aparecen en el conjunto en cuestión.

$$[q_0] = \{q_0\}$$
$$[q_1] = \{q_1, q_2, q_3\}$$

Segunda partición Recordemos que la tabla correspondiente a la función de transición $\tau_{\mathbb{A}}$ es la siguiente:

$\tau_{\mathbb{A}}$	a	b	c
q_0	q_1	q_2	q_0
q_1	q_0	q_3	q_1
q_2	q_3	q_0	q_2
q_3	q_2	q_1	q_3

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo una b se pasa al estado q_3, en vez de q_3 vamos a poner su grupo, es decir, $[q_1]$. Procederemos de la misma forma en todos los casos:

	a	b	c
q_0	$[q_1]$	$[q_1]$	$[q_0]$
q_1	$[q_0]$	$[q_1]$	$[q_1]$
q_2	$[q_1]$	$[q_0]$	$[q_1]$
q_3	$[q_1]$	$[q_1]$	$[q_1]$

Ahora vamos a ver si hay que partir los conjuntos $[q_0]$ y $[q_1]$.

Como el conjunto $[q_0]$ solo contiene un elemento, ya no se puede partir más. En el conjunto $[q_1]$ vemos que los estados q_1, q_2 y q_3 tienen distinto comportamiento. En q_1, cuando se lee una a o una b se va al conjunto $[q_1]$ y con c al conjunto $[q_0]$. En q_2, cuando se lee una a o una c se va al conjunto $[q_1]$ y con b al conjunto $[q_0]$. En q_3, tanto con a como con b como con c se va al conjunto $[q_1]$. Por tanto, vamos a partir el conjunto $[q_1]$ teniendo en cuenta esos tres comportamientos:

$$
\begin{aligned}
[q_0] &= \{q_0\}\\
[q_1] &= \{q_1\}\\
[q_2] &= \{q_2\}\\
[q_3] &= \{q_3\}
\end{aligned}
$$

No hay tercera partición Como cada partición contiene un único estado, no es posible partir más los conjuntos. Por tanto, la partición definitiva es:

$$
\begin{aligned}
[q_0] &= \{q_0\}\\
[q_1] &= \{q_1\}\\
[q_2] &= \{q_2\}\\
[q_3] &= \{q_3\}
\end{aligned}
$$

Esto significa que no se pueden eliminar estados del AFDC de la figura 7.6.8 de la página 87. Ese AFDC es minimal.

Por tanto, los conjuntos son estos cuatro:

$$
\begin{aligned}
[q_0] &= \{q_0\}\\
[q_1] &= \{q_1\}\\
[q_2] &= \{q_2\}\\
[q_3] &= \{q_3\}
\end{aligned}
$$

Cada conjunto representa un estado. En este caso, $\tau_{\mathbb{A}}$ y $\tau'_{\mathbb{A}'}$ son iguales:

$\tau'_{\mathbb{A}'}$	a	b	c
q_0	$[q_1]$	$[q_2]$	$[q_0]$
q_1	$[q_0]$	$[q_3]$	$[q_1]$
q_2	$[q_3]$	$[q_0]$	$[q_2]$
q_3	$[q_2]$	$[q_1]$	$[q_3]$

Cálculo de σ' $[q_0]$ será el estado inicial σ' porque contiene a q_0, que es el estado inicial del AFDC original.

Cálculo de Y' En este caso, el conjunto $[q_0]$ es el único que está formado por estados de Y. Consecuentemente, $Y' = \{[q_0]\}$.

7.6.3.4.3 Ejemplo 4: AFDC minimizado

En la figura 7.6.8 de la página 87, se muestra el AFDC minimal obtenido, que viene a ser el AFDC original.

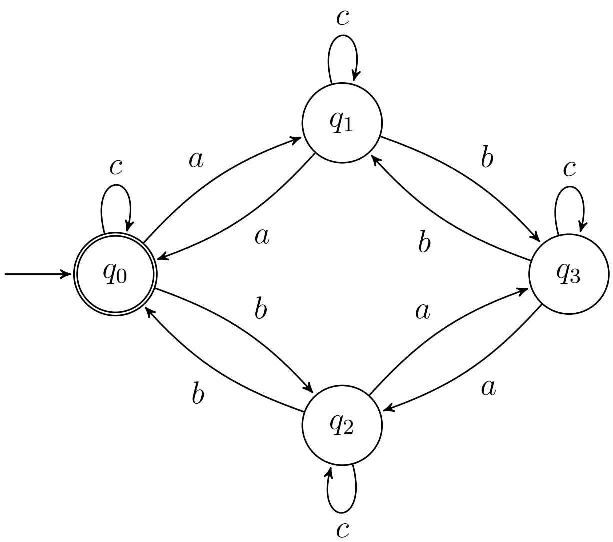


Figura 7.6.7. Minimización: ejemplo 4. AFDC de partida.

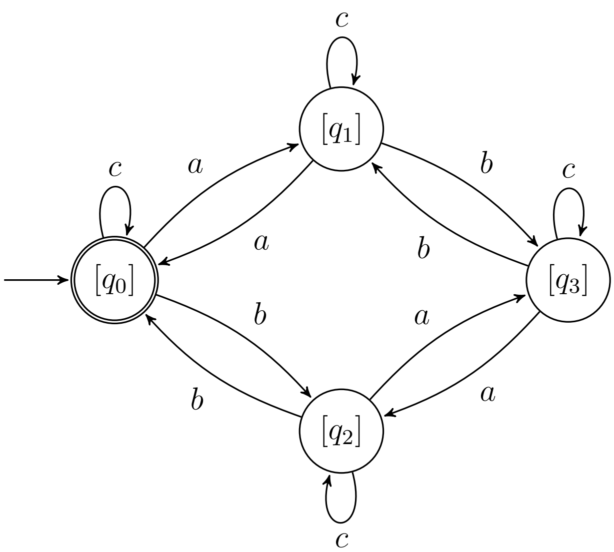


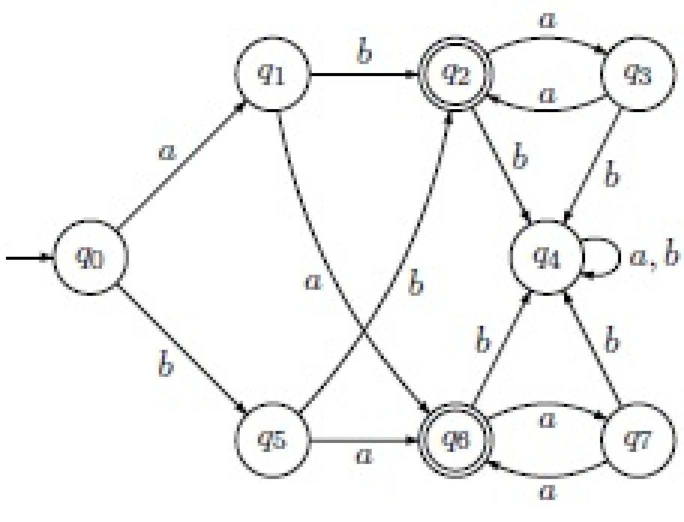
Figura 7.6.8. Minimización: ejemplo 4. AFDC minimal correspondiente al AFDC de la figura 7.6.7 (página 87).

7.7.

Minimización de los AFDC sin transiciones λ: Ejercicios resueltos

7.7.1 Minimización: Ejercicio 1

Minimizar el siguiente AFDC:



La tabla correspondiente a la función de transición $\tau_{\mathbb{A}}$ es la siguiente:

$\tau_{\mathbb{A}}$	a	b
q_0	q_1	q_5
q_1	q_6	q_2
q_2	q_3	q_4
q_3	q_2	q_4
q_4	q_4	q_4
q_5	q_6	q_2
q_6	q_7	q_4
q_7	q_6	q_4

7.7.1.1 Primera partición

La primera partición consiste en generar un conjunto con los estados que pertenecen a Y (los estados que responden "Sí") y otro con los que no pertenecen a Y (los estados que responden "No"). A cada conjunto lo llamaremos $[q_j]$ donde j es el menor de los índices que aparecen en el conjunto en cuestión.

$$[q_0] = \{q_0, q_1, q_3, q_4, q_5, q_7\}$$
$$[q_2] = \{q_2, q_6\}$$

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo una a se pasa al estado q_6, en vez de q_6 vamos a poner su grupo, es decir, $[q_2]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_0]$	$[q_0]$
q_1	$[q_2]$	$[q_2]$
q_2	$[q_0]$	$[q_0]$
q_3	$[q_2]$	$[q_0]$
q_4	$[q_0]$	$[q_0]$
q_5	$[q_2]$	$[q_2]$
q_6	$[q_0]$	$[q_0]$
q_7	$[q_2]$	$[q_0]$

7.7.1.2 Segunda partición

Vamos a comprobar si hay que partir los conjuntos $[q_0]$ y $[q_2]$.

En el conjunto $[q_0]$ vemos que los estados q_0 y q_4 tienen el mismo comportamiento: tanto cuando se lee una a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_0]$. En cambio, los estados q_1 y q_5 tienen otro comportamiento: tanto cuando se lee una a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_2]$. Por su parte, los estados q_3 y q_7 muestran otro comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_2]$ y cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_0]$. Por tanto, vamos a partir el conjunto $[q_0]$ teniendo en cuenta esos tres comportamientos. En cuanto al conjunto $[q_2]$, sus dos componentes muestran el mismo comportamiento: tanto cuando se lee una a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_0]$. Por tanto, nos quedan cuatro conjuntos:

$$[q_0] = \{q_0, q_4\}$$
$$[q_1] = \{q_1, q_5\}$$
$$[q_2] = \{q_2, q_6\}$$
$$[q_3] = \{q_3, q_7\}$$

IMPORTANTE: Es importante darse cuenta de que aunque q_0 y q_4 tengan el mismo comportamiento que los estados q_2 y q_6, no hay que ponerlos en el mismo conjunto. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar.

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo una a se pasa al estado q_6, en vez de q_6 vamos a poner su grupo, es decir, $[q_2]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_1]$	$[q_1]$
q_1	$[q_2]$	$[q_2]$
q_2	$[q_3]$	$[q_0]$
q_3	$[q_2]$	$[q_0]$
q_4	$[q_0]$	$[q_0]$
q_5	$[q_2]$	$[q_2]$
q_6	$[q_3]$	$[q_0]$
q_7	$[q_2]$	$[q_0]$

7.7.1.3 Tercera partición

Vamos a comprobar si hay que partir los conjuntos $[q_0]$, $[q_1]$, $[q_2]$ y $[q_3]$.

En el conjunto $[q_0]$ vemos que los estados q_0 y q_4 tienen distinto comportamiento: tanto cuando se lee una a como cuando se lee una b, del estado q_0 se pasa a un estado que pertenece al conjunto $[q_1]$, mientras que tanto cuando se lee una a como cuando se lee una b, del estado q_4 se pasa a un estado que pertenece al conjunto $[q_0]$. Esto quiere decir que hay que partir el conjunto $[q_0]$ en dos. En cambio, en el conjunto $[q_1]$ los estados q_1 y q_5 tienen el mismo comportamiento: tanto cuando se lee una a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_2]$. Por su parte, en el conjunto $[q_3]$ los estados q_3 y q_7 muestran también el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_2]$ y cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_0]$. En cuanto al conjunto $[q_2]$, sus dos componentes (q_2 y q_6) muestran el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_3]$ y cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_0]$. Por tanto, solo vamos a partir el conjunto $[q_0]$. Consecuentemente, nos quedan cinco conjuntos:

$$\begin{array}{l} [q_0] = \{q_0\} \\ [q_1] = \{q_1, q_5\} \\ [q_2] = \{q_2, q_6\} \\ [q_3] = \{q_3, q_7\} \\ [q_4] = \{q_4\} \end{array}$$

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo una a se pasa al estado q_6, en vez de q_6 vamos a poner su grupo, es decir, $[q_2]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_1]$	$[q_1]$
q_1	$[q_2]$	$[q_2]$
q_2	$[q_3]$	$[q_4]$
q_3	$[q_2]$	$[q_4]$
q_4	$[q_4]$	$[q_4]$
q_5	$[q_2]$	$[q_2]$
q_6	$[q_3]$	$[q_4]$
q_7	$[q_2]$	$[q_4]$

7.7.1.4 No hay cuarta partición

Ahora vamos a ver si hay que partir los conjuntos $[q_0]$, $[q_1]$, $[q_2]$, $[q_3]$ y $[q_4]$. Los conjuntos $[q_0]$ y $[q_4]$ no se pueden partir porque contienen un único elemento, así que nos centramos en los conjuntos $[q_1]$, $[q_2]$ y $[q_3]$.

En el conjunto $[q_1]$ vemos que los estados q_1 y q_5 tienen el mismo comportamiento: tanto cuando se lee una a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_2]$. En el conjunto $[q_2]$ vemos que los estados q_2 y q_6 tienen el mismo comportamiento: cuando se lee una a pasan a un estado que pertenece al conjunto $[q_3]$ y cuando se lee una b pasan a un estado que pertenece al conjunto $[q_4]$. En el conjunto $[q_3]$ ocurre algo parecido, es decir, los estados q_3 y q_7 tienen el mismo comportamiento: cuando se lee una a pasan a un estado que pertenece al conjunto $[q_2]$ y cuando se lee una b pasan a un estado que pertenece al conjunto $[q_4]$. Por tanto, no surge ninguna partición nueva.

7.7.1.5 AFDC minimal

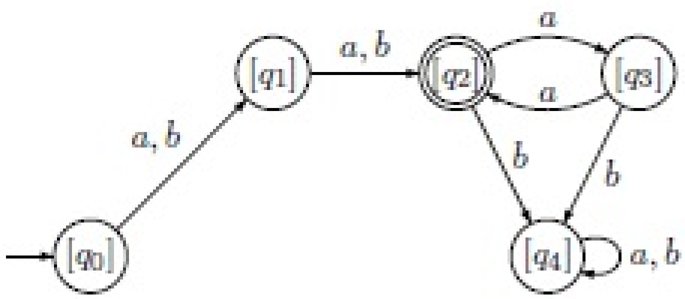
Por tanto, los conjuntos son estos cinco:

$$
\begin{aligned}
[q_0] &= \{q_0\} \\
[q_1] &= \{q_1, q_5\} \\
[q_2] &= \{q_2, q_6\} \\
[q_3] &= \{q_3, q_7\} \\
[q_4] &= \{q_4\}
\end{aligned}
$$

Cada conjunto representa un estado. Teniendo en cuenta la última tabla, se calculan las transiciones:

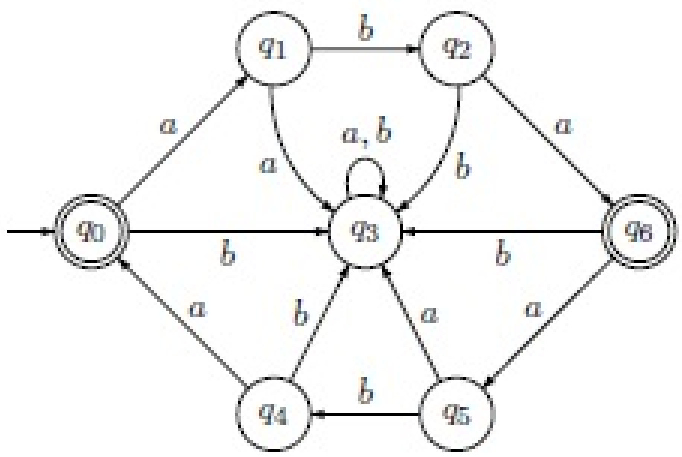
	a	b
q_0	$[q_1]$	$[q_1]$
q_1	$[q_2]$	$[q_2]$
q_2	$[q_3]$	$[q_4]$
q_3	$[q_2]$	$[q_4]$
q_4	$[q_4]$	$[q_4]$

$[q_0]$ será el estado inicial porque contiene al estado inicial q_0 del AFDC original. Por otra parte $[q_2]$ será el único estado con doble círculo porque es el único conjunto formado por estados que pertenecen al conjunto Y del AFDC original.



7.7.2 Minimización: Ejercicio 2

Minimizar el siguiente AFDC:



La tabla correspondiente a la función de transición $\tau_{\mathbb{A}}$ es la siguiente:

$\tau_{\mathbb{A}}$	a	b
q_0	q_1	q_3
q_1	q_3	q_2
q_2	q_6	q_3
q_3	q_3	q_3
q_4	q_0	q_3
q_5	q_3	q_4
q_6	q_5	q_3

7.7.2.1 Primera partición

La primera partición consiste en generar un conjunto con los estados que pertenecen a Y (los estados que responden "Sí") y otro con los que no pertenecen a Y (los estados que responden "No"). A cada conjunto lo llamaremos $[q_j]$ donde j es el menor de los índices que aparecen en el conjunto en cuestión.

$$[q_0] = \{q_0, q_6\}$$
$$[q_1] = \{q_1, q_2, q_3, q_4, q_5\}$$

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo una a se pasa al estado q_3, en vez de q_3 vamos a poner su grupo, es decir, $[q_1]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_1]$	$[q_1]$
q_1	$[q_1]$	$[q_1]$
q_2	$[q_0]$	$[q_1]$
q_3	$[q_1]$	$[q_1]$
q_4	$[q_0]$	$[q_1]$
q_5	$[q_1]$	$[q_1]$
q_6	$[q_1]$	$[q_1]$

7.7.2.2 Segunda partición

Vamos a comprobar si hay que partir los conjuntos $[q_0]$ y $[q_1]$.

En el conjunto $[q_0]$ vemos que los estados q_0 y q_6 tienen el mismo comportamiento: tanto cuando se lee una a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_1]$. En el conjunto $[q_1]$, los estados q_1, q_3 y q_5 tienen el mismo comportamiento: tanto cuando se lee una a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_1]$. Por su parte, los estados q_2 y q_4 muestran otro comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_0]$ y cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_1]$. Por tanto, vamos a partir el conjunto $[q_1]$ teniendo en cuenta esos dos casos de comportamientos dentro del conjunto $[q_1]$. Consecuentemente, nos quedan tres conjuntos:

$$\begin{aligned}[] [q_0] &= \{q_0, q_6\} \\ [q_1] &= \{q_1, q_3, q_5\} \\ [q_2] &= \{q_2, q_4\} \end{aligned}$$

IMPORTANTE: Es importante darse cuenta de que aunque q_0 y q_6 tengan el mismo comportamiento que los estados q_1, q_3 y q_5, no hay que ponerlos en el mismo conjunto. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar.

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo un a se pasa al estado q_3, en vez de q_3 vamos a poner su grupo, es decir, $[q_1]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_1]$	$[q_1]$
q_1	$[q_1]$	$[q_2]$
q_2	$[q_0]$	$[q_1]$
q_3	$[q_1]$	$[q_1]$
q_4	$[q_0]$	$[q_1]$
q_5	$[q_1]$	$[q_2]$
q_6	$[q_1]$	$[q_1]$

7.7.2.3 Tercera partición

Vamos a comprobar si hay que partir los conjuntos $[q_0]$, $[q_1]$ y $[q_2]$.

En el conjunto $[q_0]$ vemos que los estados q_0 y q_6 tienen el mismo comportamiento: tanto cuando se lee una a como cuando se lee una b, se pasa a un estado que pertenece al conjunto $[q_1]$. En el conjunto $[q_1]$ vemos que los estados q_1 y q_5 tienen el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_1]$ y cuando se lee una b, se pasa a un estado que pertenece al conjunto $[q_2]$. En cambio, del estado q_3 tanto cuando se lee una a como cuando se lee una b, se pasa a un estado que pertenece al conjunto $[q_1]$. Por tanto, hay que partir el conjunto $[q_1]$: por un lado q_1 y q_5 y, por otro lado, q_3. En cuanto al conjunto $[q_2]$, sus dos componentes muestran el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_0]$ y cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_1]$. Por tanto, solo vamos a partir el conjunto $[q_1]$. Consecuentemente, nos quedan cuatro conjuntos:

$$\begin{array}{l} [q_0] = \{q_0, q_6\} \\ [q_1] = \{q_1, q_5\} \\ [q_2] = \{q_2, q_4\} \\ [q_3] = \{q_3\} \end{array}$$

IMPORTANTE: Es importantante darse cuenta de que aunque q_0 y q_6 tengan el mismo comportamiento que el estado q_3, no hay que ponerlos en el mismo conjunto. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar.

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_5 leyendo una b se pasa al estado q_4, en vez de q_4 vamos a poner su grupo, es decir, $[q_2]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_1]$	$[q_3]$
q_1	$[q_3]$	$[q_2]$
q_2	$[q_0]$	$[q_3]$
q_3	$[q_3]$	$[q_3]$
q_4	$[q_0]$	$[q_3]$
q_5	$[q_3]$	$[q_2]$
q_6	$[q_1]$	$[q_3]$

7.7.2.4 No hay cuarta partición

Ahora vamos a ver si hay que partir los conjuntos $[q_0]$, $[q_1]$, $[q_2]$ y $[q_3]$. El conjunto $[q_3]$ no se puede partir porque contiene un único elemento, así que nos centramos en los conjuntos $[q_0]$, $[q_1]$ y $[q_2]$.

En el conjunto $[q_0]$ vemos que los estados q_0 y q_6 tienen el mismo comportamiento: cuando se lee una a pasan a un estado que pertenece al conjunto $[q_1]$ y cuando se lee una b pasan a un estado que pertenece al conjunto $[q_3]$. En el conjunto $[q_1]$ vemos que los estados q_1 y q_5 tienen el mismo comportamiento: cuando se lee una a pasan a un estado que pertenece al conjunto $[q_3]$ y cuando se lee una b pasan a un estado que pertenece al conjunto $[q_2]$. En el conjunto $[q_2]$ vemos que los estados q_2 y q_4 tienen el mismo comportamiento: cuando se lee una a pasan a un estado que pertenece al conjunto $[q_0]$ y cuando se lee una b pasan a un estado que pertenece al conjunto $[q_3]$. Por tanto, no surge ninguna partición nueva.

7.7.2.5 AFDC minimal

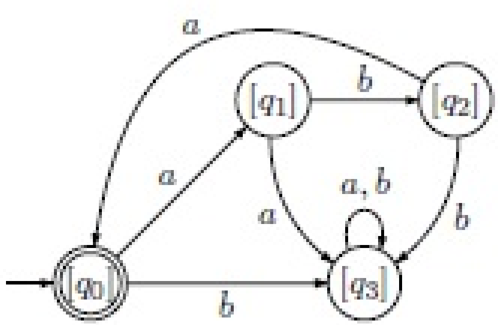
Los conjuntos definitivos son estos cuatro:

$$\begin{aligned}
[q_0] &= \{q_0, q_6\} \\
[q_1] &= \{q_1, q_5\} \\
[q_2] &= \{q_2, q_4\} \\
[q_3] &= \{q_3\}
\end{aligned}$$

Cada conjunto representa un estado. Teniendo en cuenta la última tabla, se calculan las transiciones:

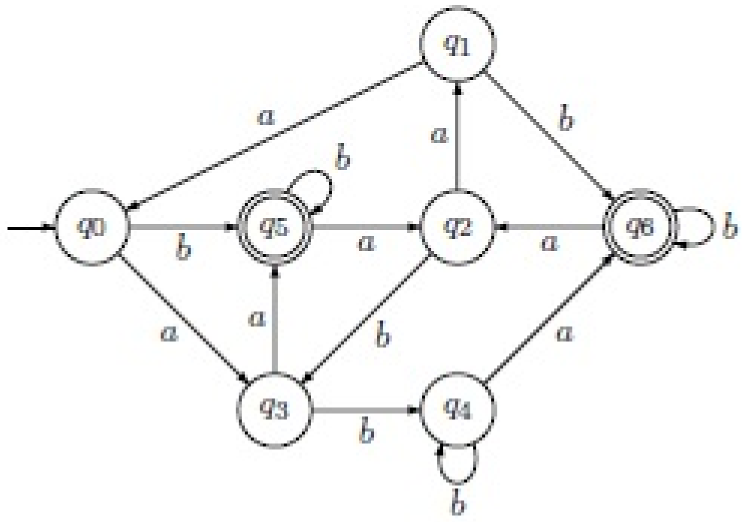
	a	b
q_0	$[q_1]$	$[q_3]$
q_1	$[q_3]$	$[q_2]$
q_2	$[q_0]$	$[q_3]$
q_3	$[q_3]$	$[q_3]$

$[q_0]$ será el estado inicial porque contiene al estado inicial q_0 del AFD original. Por otra parte $[q_0]$ será también el único estado con doble círculo porque es el único conjunto formado por estados que pertenecen al conjunto Y del AFD original.



7.7.3 Minimización: Ejercicio 3

Minimizar el siguiente AFDC:



La tabla correspondiente a la función de transición $\tau_{\mathbb{A}}$ es la siguiente:

$\tau_{\mathbb{A}}$	a	b
q_0	q_3	q_5
q_1	q_0	q_6
q_2	q_1	q_3
q_3	q_5	q_4
q_4	q_6	q_4
q_5	q_2	q_5
q_6	q_2	q_6

7.7.3.1 Primera partición

La primera partición consiste en generar un conjunto con los estados que pertenecen a Y (los estados que responden "Sí") y otro con los que no pertenecen a Y (los estados que responden "No"). A cada conjunto lo llamaremos $[q_j]$ donde j es el menor de los índices que aparecen en el conjunto en cuestión.

$$[q_0] = \{q_0, q_1, q_2, q_3, q_4\}$$
$$[q_5] = \{q_5, q_6\}$$

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo una b se pasa al estado q_6, en vez de q_6 vamos a poner su grupo, es decir, $[q_5]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_0]$	$[q_5]$
q_1	$[q_0]$	$[q_5]$
q_2	$[q_0]$	$[q_0]$
q_3	$[q_5]$	$[q_0]$
q_4	$[q_5]$	$[q_0]$
q_5	$[q_0]$	$[q_5]$
q_6	$[q_0]$	$[q_5]$

7.7.3.2 Segunda partición

Vamos a comprobar si hay que partir los conjuntos $[q_0]$ y $[q_5]$.

En el conjunto $[q_0]$ vemos que los estados q_0 y q_1 tienen el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_0]$ y cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_5]$. Por su parte, el estado q_2 muestra un comportamiento que no coincide con el de ningún otro estado de $[q_0]$: tanto cuando se lee una a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_0]$. Por otro lado, los estados q_3 y q_4 tienen el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_5]$ y cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_0]$. Todo esto significa que hay que partir el conjunto $[q_0]$ en tres conjuntos: $[q_0] = \{q_0, q_1\}$, $[q_2] = \{q_2\}$ y $[q_3] = \{q_3, q_4\}$. En cuanto al conjunto $[q_5]$, los estados q_5 y q_6 tienen el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_0]$ y cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_5]$. Por tanto, no hay que partir el conjunto $[q_5]$. Consecuentemente, nos quedan cuatro conjuntos:

$$
\begin{aligned}
&[q_0] = \{q_0, q_1\} \\
&[q_2] = \{q_2\} \\
&[q_3] = \{q_3, q_4\} \\
&[q_5] = \{q_5, q_6\}
\end{aligned}
$$

IMPORTANTE: Es importante darse cuenta de que aunque q_0 y q_1 tengan el mismo comportamiento que los estados q_5 y q_6, no hay que ponerlos en el mismo conjunto. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar.

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo un b se pasa al estado q_6, en vez de q_6 vamos a poner su grupo, es decir, $[q_5]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_3]$	$[q_5]$
q_1	$[q_0]$	$[q_5]$
q_2	$[q_0]$	$[q_3]$
q_3	$[q_5]$	$[q_3]$
q_4	$[q_5]$	$[q_3]$
q_5	$[q_2]$	$[q_5]$
q_6	$[q_2]$	$[q_5]$

7.7.3.3 Tercera partición

Vamos a comprobar si hay que partir los conjuntos $[q_0]$, $[q_3]$ y $[q_5]$. Como el conjunto $[q_2]$ solo tiene un elemento, se quedará tal cual.

En el conjunto $[q_0]$ vemos que los estados q_0 y q_1 tienen distinto comportamiento: desde q_0 leyendo una a se pasa a un estado que pertenece al conjunto $[q_3]$ y leyendo una b, se pasa a un estado que pertenece al conjunto $[q_5]$; en cambio, desde q_1 leyendo una a se pasa a un estado que pertenece al conjunto $[q_0]$ y leyendo una b se pasa a un estado que pertenece al conjunto $[q_5]$. Esto significa que hay que partir el conjunto $[q_0]$ en dos: $[q_0] = \{q_0\}$ y $[q_1] = \{q_1\}$. En el conjunto $[q_3]$ vemos que los estados q_3 y q_4 tienen el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_5]$ y cuando se lee una b, se pasa a un estado que pertenece al conjunto $[q_3]$. Por tanto, no hay que partir el conjunto $[q_3]$. Ocurre lo mismo en el conjunto $[q_5]$ puesto que los estados q_5 y q_6 tienen el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_2]$ y cuando se lee una b, se pasa a un estado que pertenece al conjunto $[q_5]$. Esto quiere decir que no hay que partir el conjunto $[q_5]$. Consecuentemente, nos quedan cinco conjuntos:

$$
\begin{aligned}
[q_0] &= \{q_0\} \\
[q_1] &= \{q_1\} \\
[q_2] &= \{q_2\} \\
[q_3] &= \{q_3, q_4\} \\
[q_5] &= \{q_5, q_6\}
\end{aligned}
$$

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_4 leyendo una b se pasa al estado q_4, en vez de q_4 vamos a poner su grupo, es decir, $[q_3]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_3]$	$[q_5]$
q_1	$[q_0]$	$[q_5]$
q_2	$[q_1]$	$[q_3]$
q_3	$[q_5]$	$[q_3]$
q_4	$[q_5]$	$[q_3]$
q_5	$[q_2]$	$[q_5]$
q_6	$[q_2]$	$[q_5]$

7.7.3.4 No hay cuarta partición

Ahora vamos a ver si hay que partir los conjuntos $[q_0]$, $[q_1]$, $[q_2]$, $[q_3]$ y $[q_5]$. Los conjuntos $[q_0]$, $[q_1]$ y $[q_2]$ no se pueden partir porque cada uno de ellos contiene un único elemento, así que nos centramos en los conjuntos $[q_3]$ y $[q_5]$.

En el conjunto $[q_3]$ vemos que los estados q_3 y q_4 tienen el mismo comportamiento: cuando se lee una a pasan a un estado que pertenece al conjunto $[q_5]$ y cuando se lee una b pasan a un estado que pertenece al conjunto $[q_3]$. Esto significa que no hay que partir el conjunto $[q_3]$. En el conjunto $[q_5]$ ocurre lo mismo, puesto que los estados q_5 y q_6 tienen el mismo comportamiento: cuando se lee una a pasan a un estado que pertenece al conjunto $[q_2]$ y cuando se lee una b pasan a un estado que pertenece al conjunto $[q_5]$. Esto significa que tampoco hay que partir el conjunto $[q_5]$. Por tanto, no surge ninguna partición nueva.

7.7.3.5 AFDC minimal

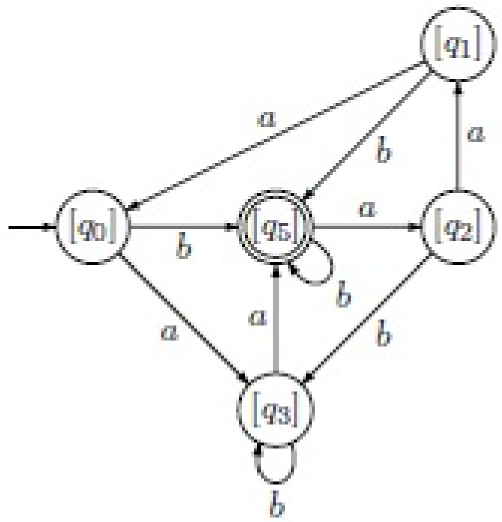
Los conjuntos definitivos son estos cinco:

$$
\begin{array}{l}
[q_0] = \{q_0\} \\
[q_1] = \{q_1\} \\
[q_2] = \{q_2\} \\
[q_3] = \{q_3, q_4\} \\
[q_5] = \{q_5, q_6\}
\end{array}
$$

Cada conjunto representa un estado. Teniendo en cuenta la última tabla, se calculan las transiciones:

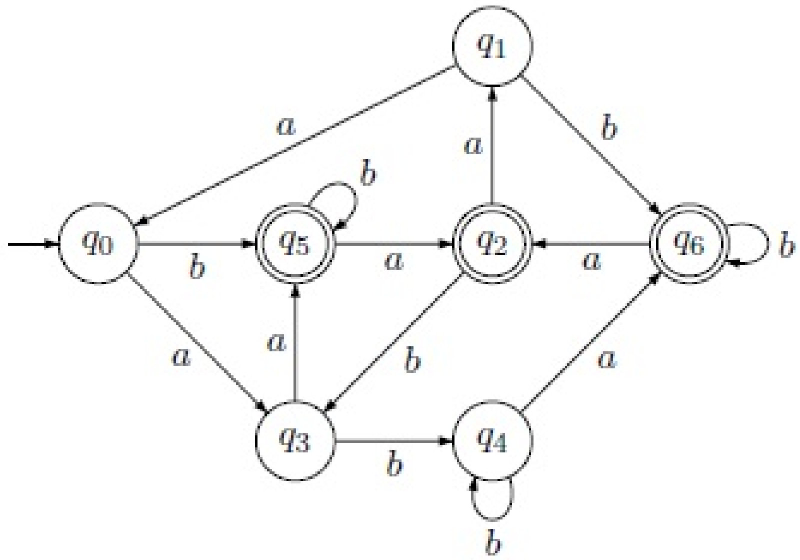
	a	b
q_0	$[q_3]$	$[q_5]$
q_1	$[q_0]$	$[q_5]$
q_2	$[q_1]$	$[q_3]$
q_3	$[q_5]$	$[q_3]$
q_5	$[q_2]$	$[q_5]$

$[q_0]$ será el estado inicial porque contiene al estado inicial q_0 del AFD original. Por otra parte, $[q_5]$ será el único estado con doble círculo porque es el único conjunto formado por estados que pertenecen al conjunto Y del AFD original.



7.7.4 Minimización: Ejercicio 4

Minimizar el siguiente AFDC:



La tabla correspondiente a la función de transición $\tau_{\mathbb{A}}$ es la siguiente:

$\tau_{\mathbb{A}}$	a	b
q_0	q_3	q_5
q_1	q_0	q_6
q_2	q_1	q_3
q_3	q_5	q_4
q_4	q_6	q_4
q_5	q_2	q_5
q_6	q_2	q_6

7.7.4.1 Primera partición

La primera partición consiste en generar un conjunto con los estados que pertenecen a Y (los estados que responden "Sí") y otro con los que no pertenecen a Y (los estados que responden "No"). A cada conjunto lo llamaremos $[q_j]$ donde j es el menor de los índices que aparecen en el conjunto en cuestión.

$$[q_0] = \{q_0, q_1, q_3, q_4\}$$
$$[q_2] = \{q_2, q_5, q_6\}$$

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo una b se pasa al estado q_6, en vez de q_6 vamos a poner su grupo, es decir, $[q_2]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_0]$	$[q_2]$
q_1	$[q_0]$	$[q_2]$
q_2	$[q_0]$	$[q_0]$
q_3	$[q_2]$	$[q_0]$
q_4	$[q_2]$	$[q_0]$
q_5	$[q_2]$	$[q_2]$
q_6	$[q_2]$	$[q_2]$

7.7.4.2 Segunda partición

Vamos a comprobar si hay que partir los conjuntos $[q_0]$ y $[q_2]$.

En el conjunto $[q_0]$ vemos que los estados q_0 y q_1 tienen el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_0]$ y cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_2]$. Por otro lado, los estados q_3 y q_4 tienen el mismo comportamiento pero distinto con respecto al comportamiento de q_0 y q_1: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_2]$ y cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_0]$. Esto significa que hay que partir el conjunto $[q_0]$ en dos conjuntos: $[q_0] = \{q_0, q_1\}$ y $[q_3] = \{q_3, q_4\}$. En cuanto al conjunto $[q_2]$, el estado q_2 muestra un comportamiento que no coincide con el de ningún otro estado de $[q_2]$: tanto cuando se lee una

a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_0]$. Por otra parte, los estados q_5 y q_6 tienen el mismo comportamiento: tanto cuando se lee una a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_2]$. Esto significa que hay que partir el conjunto $[q_2]$ en dos conjuntos: $[q_2] = \{q_2\}$ y $[q_5] = \{q_5, q_6\}$. Consecuentemente, nos quedan cuatro conjuntos:

$$\begin{aligned}
[q_0] &= \{q_0, q_1\}\\
[q_2] &= \{q_2\}\\
[q_3] &= \{q_3, q_4\}\\
[q_5] &= \{q_5, q_6\}
\end{aligned}$$

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo una b se pasa al estado q_6, en vez de q_6 vamos a poner su grupo, es decir, $[q_5]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_3]$	$[q_5]$
q_1	$[q_0]$	$[q_5]$
q_2	$[q_0]$	$[q_3]$
q_3	$[q_5]$	$[q_3]$
q_4	$[q_5]$	$[q_3]$
q_5	$[q_2]$	$[q_5]$
q_6	$[q_2]$	$[q_5]$

7.7.4.3 Tercera partición

Como el conjunto $[q_2]$ solo tiene un elemento, no se puede partir. Vamos a centrarnos en averiguar si hay que partir los conjuntos $[q_0]$, $[q_3]$ y $[q_5]$.

En el conjunto $[q_0]$ vemos que los estados q_0 y q_1 tienen distinto comportamiento: desde q_0 leyendo una a se pasa a un estado que pertenece al conjunto $[q_3]$ y leyendo una b, se pasa a un estado que pertenece al conjunto $[q_5]$; en cambio, desde q_1 leyendo una a se pasa a un estado que pertenece al conjunto $[q_0]$ y leyendo una b, se pasa a un estado que pertenece al conjunto $[q_5]$. Esto significa que hay que partir el conjunto $[q_0]$ en dos: $[q_0] = \{q_0\}$ y $[q_1] = \{q_1\}$. En el conjunto $[q_3]$ vemos que los estados q_3 y q_4 tienen el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_5]$ y cuando se lee una b, se pasa a un estado que pertenece al conjunto $[q_3]$. Por tanto, no hay que partir el conjunto $[q_3]$. Ocurre lo mismo en el conjunto $[q_5]$ puesto que los estados q_5 y q_6 tienen el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_2]$ y cuando se lee una b, se pasa a un estado que pertenece al conjunto $[q_5]$. Esto quiere decir que no hay que partir el conjunto $[q_5]$. Consecuentemente, nos quedan cinco conjuntos:

$$\begin{aligned}
[q_0] &= \{q_0\}\\
[q_1] &= \{q_1\}\\
[q_2] &= \{q_2\}\\
[q_3] &= \{q_3, q_4\}\\
[q_5] &= \{q_5, q_6\}
\end{aligned}$$

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_4 leyendo una b se pasa al estado q_4, en vez de q_4 vamos a poner su grupo, es decir, $[q_3]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_3]$	$[q_5]$
q_1	$[q_0]$	$[q_5]$
q_2	$[q_1]$	$[q_3]$
q_3	$[q_5]$	$[q_3]$
q_4	$[q_5]$	$[q_3]$
q_5	$[q_2]$	$[q_5]$
q_6	$[q_2]$	$[q_5]$

7.7.4.4 No hay cuarta partición

Los conjuntos $[q_0]$, $[q_1]$ y $[q_2]$ no se pueden partir porque son unitarios, es decir, cada uno de ellos contiene un único elemento, por lo cual, solo nos queda averiguar si hay que partir los conjuntos $[q_3]$ y $[q_5]$.

En el conjunto $[q_3]$ vemos que los estados q_3 y q_4 tienen el mismo comportamiento: cuando se lee una a pasan a un estado que pertenece al conjunto $[q_5]$ y cuando se lee una b pasan a un estado que pertenece al conjunto $[q_3]$. Esto significa que no hay que partir el conjunto $[q_3]$. En el conjunto $[q_5]$ ocurre lo mismo, puesto que los estados q_5 y q_6 tienen el mismo comportamiento: cuando se lee una a pasan a un estado que pertenece al conjunto $[q_2]$ y cuando se lee una b pasan a un estado que pertenece al conjunto $[q_5]$. Esto significa que tampoco hay que partir el conjunto $[q_5]$. Por tanto, no surge ninguna partición nueva.

7.7.4.5 AFDC minimal

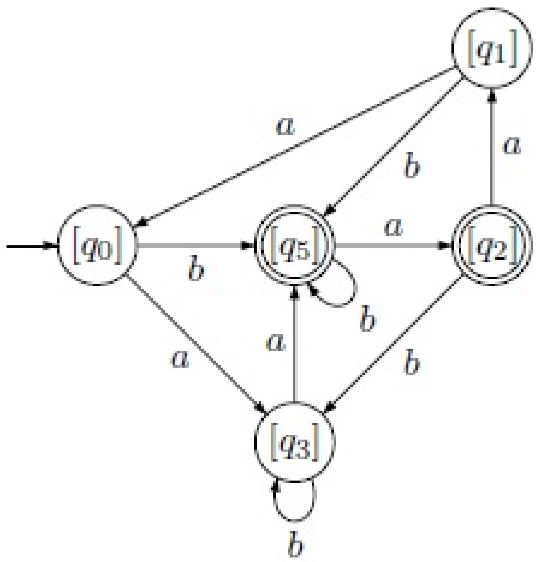
Los conjuntos definitivos son estos cinco:

$$\begin{aligned}
[q_0] &= \{q_0\}\\
[q_1] &= \{q_1\}\\
[q_2] &= \{q_2\}\\
[q_3] &= \{q_3, q_4\}\\
[q_5] &= \{q_5, q_6\}
\end{aligned}$$

Cada conjunto representa un estado. Teniendo en cuenta la última tabla, se calculan las transiciones:

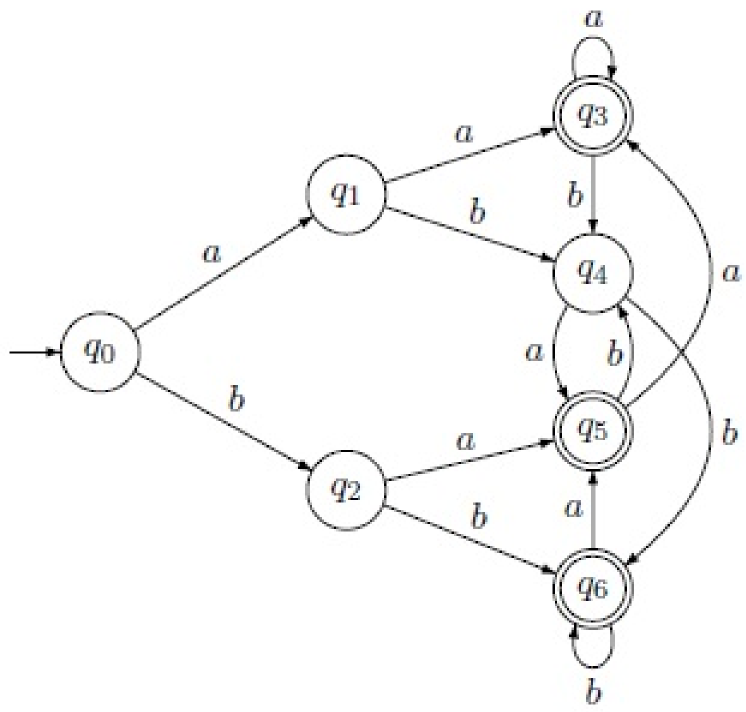
	a	b
q_0	$[q_3]$	$[q_5]$
q_1	$[q_0]$	$[q_5]$
q_2	$[q_1]$	$[q_3]$
q_3	$[q_5]$	$[q_3]$
q_5	$[q_2]$	$[q_5]$

$[q_0]$ será el estado inicial porque contiene al estado inicial q_0 del AFDC original. Por otra parte, $[q_2]$ y $[q_5]$ serán los estados con doble círculo porque están formados por estados que pertenecen al conjunto Y del AFDC original.



7.7.5 Minimización: Ejercicio 5

Minimizar el siguiente AFDC:



La tabla correspondiente a la función de transición $\tau_{\mathbb{A}}$ es la siguiente:

$\tau_{\mathbb{A}}$	a	b
q_0	q_1	q_2
q_1	q_3	q_4
q_2	q_5	q_6
q_3	q_3	q_4
q_4	q_5	q_6
q_5	q_3	q_4
q_6	q_5	q_6

7.7.5.1 Primera partición

La primera partición consiste en generar un conjunto con los estados que pertenecen a Y (los estados que responden "Sí") y otro con los que no pertenecen a Y (los estados que responden "No"). A cada conjunto lo llamaremos $[q_j]$ donde j es el menor de los índices que aparecen en el conjunto en cuestión.

$$[q_0] = \{q_0, q_1, q_2, q_4\}$$
$$[q_3] = \{q_3, q_5, q_6\}$$

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo una b se pasa al estado q_6, en vez de q_6 vamos a poner su grupo, es decir, $[q_2]$. Procederemos de la misma

	a	b
q_0	$[q_0]$	$[q_0]$
q_1	$[q_3]$	$[q_0]$
q_2	$[q_3]$	$[q_3]$
q_3	$[q_3]$	$[q_0]$
q_4	$[q_3]$	$[q_3]$
q_5	$[q_3]$	$[q_0]$
q_6	$[q_3]$	$[q_3]$

7.7.5.2 Segunda partición

Vamos a comprobar si hay que partir los conjuntos $[q_0]$ y $[q_3]$.

En el conjunto $[q_0]$ vemos que el estado q_0 muestra un comportamiento que no coincide con el de ningún otro estado de $[q_0]$: tanto cuando se lee una a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_0]$; el estado q_1 muestra también un comportamiento que no coincide con el de ningún otro estado de $[q_0]$: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_3]$ y cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_0]$; por su parte, los estados q_2 y q_4 tienen el mismo comportamiento: tanto cuando se lee una a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_3]$. Esto todo significa que hay que partir el conjunto $[q_0]$ en tres conjuntos: $[q_0] = \{q_0\}$, $[q_1] = \{q_1\}$ y $[q_2] = \{q_2, q_4\}$. En cuanto al conjunto $[q_3]$, el estado q_6 muestra un comportamiento que no coincide con el de ningún otro estado de $[q_3]$: tanto cuando se lee una a como cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_3]$. Por otra parte, los estados q_3 y q_5 tienen el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_3]$ y cuando se lee una b se pasa a un estado que pertenece al conjunto $[q_0]$. Esto significa que hay que partir el conjunto $[q_3]$ en dos conjuntos: $[q_3] = \{q_3, q_5\}$ y $[q_6] = \{q_6\}$. Consecuentemente, nos quedan cinco conjuntos:

$$
\begin{aligned}
&[q_0] = \{q_0\} \\
&[q_1] = \{q_1\} \\
&[q_2] = \{q_2, q_4\} \\
&[q_3] = \{q_3, q_5\} \\
&[q_6] = \{q_6\}
\end{aligned}
$$

IMPORTANTE: Es importante darse cuenta de que aunque q_2 y q_4 tengan el mismo comportamiento que q_6, no hay que ponerlos en el mismo conjunto. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar. Ocurre lo mismo con los estados q_3 y q_5 con respecto al estado q_1.

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo una b se pasa al estado q_4, en vez de q_4 vamos a poner su grupo, es decir, $[q_2]$. Procederemos de la misma forma en todos los casos:

	a	b
q_0	$[q_1]$	$[q_2]$
q_1	$[q_3]$	$[q_2]$
q_2	$[q_3]$	$[q_6]$
q_3	$[q_3]$	$[q_2]$
q_4	$[q_3]$	$[q_6]$
q_5	$[q_3]$	$[q_2]$
q_6	$[q_3]$	$[q_6]$

7.7.5.3 No hay tercera partición

Como los conjuntos $[q_0]$, $[q_1]$ y $[q_6]$ son unitarios, es decir, contienen un único elemento, no se pueden partir. Por ello, nos queda averiguar si hay que partir los conjuntos $[q_2]$ y $[q_3]$.

En el conjunto $[q_2]$ vemos que los estados q_2 y q_4 tienen el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_3]$ y cuando se lee una b, se pasa a un estado que pertenece al conjunto $[q_6]$. Por tanto, no hay que partir el conjunto $[q_2]$. Ocurre lo mismo en el conjunto $[q_3]$ puesto que los estados q_3 y q_5 tienen el mismo comportamiento: cuando se lee una a se pasa a un estado que pertenece al conjunto $[q_3]$ y cuando se lee una b, se pasa a un estado que pertenece al conjunto $[q_2]$. Esto quiere decir que no hay que partir el conjunto $[q_3]$. Consecuentemente, nos quedan los mismos cinco conjuntos que han surgido tras la segunda partición:

$$\begin{array}{l}
[q_0] = \{q_0\} \\
[q_1] = \{q_1\} \\
[q_2] = \{q_2, q_4\} \\
[q_3] = \{q_3, q_5\} \\
[q_6] = \{q_6\}
\end{array}$$

IMPORTANTE: Tal como se ha dicho tras efectuar la segunda partición, aunque q_2 y q_4 tengan el mismo comportamiento que q_6, no hay que ponerlos en el mismo conjunto. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar. Ocurre lo mismo con los estados q_3 y q_5 con respecto al estado q_1.

7.7.5.4 AFDC minimal

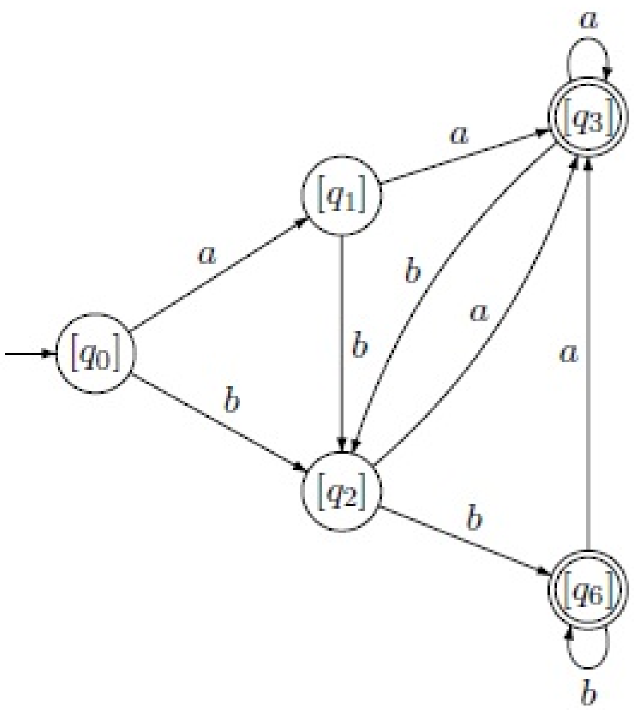
Los conjuntos definitivos son estos cinco:

$$\begin{array}{l}
[q_0] = \{q_0\} \\
[q_1] = \{q_1\} \\
[q_2] = \{q_2, q_4\} \\
[q_3] = \{q_3, q_5\} \\
[q_6] = \{q_6\}
\end{array}$$

Cada conjunto representa un estado. Teniendo en cuenta la última tabla, se calculan las transiciones:

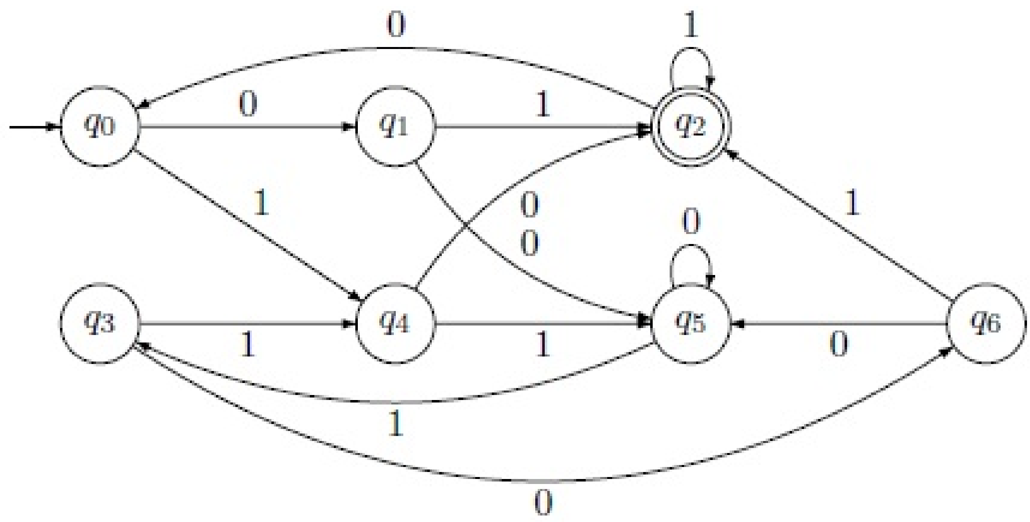
	a	b
q_0	$[q_1]$	$[q_2]$
q_1	$[q_3]$	$[q_2]$
q_2	$[q_3]$	$[q_6]$
q_3	$[q_3]$	$[q_2]$
q_6	$[q_3]$	$[q_6]$

$[q_0]$ será el estado inicial porque contiene al estado inicial q_0 del AFDC original. Por otra parte, $[q_3]$ y $[q_6]$ serán los estados con doble círculo porque están formados por estados que pertenecen al conjunto Y del AFDC original.



7.7.6 Minimización: Ejercicio 6

Minimizar el siguiente AFDC:



La tabla correspondiente a la función de transición $\tau_{\mathbb{A}}$ es la siguiente:

$\tau_{\mathbb{A}}$	0	1
q_0	q_1	q_4
q_1	q_5	q_2
q_2	q_0	q_2
q_3	q_6	q_4
q_4	q_2	q_5
q_5	q_5	q_3
q_6	q_5	q_2

7.7.6.1 Primera partición

Por una parte, ponemos en un conjunto los estados que pertenecen a Y (los estados que responden "Sí") y, por otra parte, ponemos en otro conjunto los que no pertenecen a Y (los estados que responden "No"). A cada conjunto lo llamaremos $[q_j]$ donde j es el menor de los índices que aparecen en el conjunto en cuestión.

$$
\begin{aligned}
&[q_0] = \{q_0, q_1, q_3, q_4, q_5, q_6\} \\
&[q_2] = \{q_2\}
\end{aligned}
$$

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_3 leyendo el símbolo 0 se pasa al estado q_6, en vez de q_6 vamos a poner su grupo, es decir, $[q_0]$. Procederemos de la misma forma en todos los casos:

	0	1
q_0	$[q_0]$	$[q_0]$
q_1	$[q_0]$	$[q_2]$
q_2	$[q_0]$	$[q_2]$
q_3	$[q_0]$	$[q_0]$
q_4	$[q_2]$	$[q_0]$
q_5	$[q_0]$	$[q_0]$
q_6	$[q_0]$	$[q_2]$

7.7.6.2 Segunda partición

Vamos a comprobar si hay que partir los conjuntos $[q_0]$ y $[q_2]$.

Puesto que $[q_2]$ contiene un único estado, se queda como está. En el conjunto $[q_0]$ vemos tres comportamientos diferentes: en primer lugar, los estados q_0, q_3 y q_5 tienen el mismo comportamiento: tanto cuando se lee el símbolo 0 como cuando se lee el símbolo 1 se pasa a un estado que pertenece al conjunto $[q_0]$; en segundo lugar, los estados q_1 y q_6 tienen el mismo comportamiento pero distinto con respecto al comportamiento de q_0, q_3 y q_5: cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_0]$ y cuando se lee el símbolo 1 se pasa a un estado que pertenece al conjunto $[q_2]$; en tercer lugar, el estado q_4 muestra un tercer comportamiento distinto con respecto al comportamiento del resto de estados de $[q_0]$: cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_2]$ y cuando se lee el símbolo 1 se pasa a un estado que pertenece al conjunto $[q_0]$. Esto significa que hay que partir el conjunto $[q_0]$ en tres conjuntos: $[q_0] = \{q_0, q_3, q_5\}$, $[q_1] = \{q_1, q_6\}$ y $[q_4] = \{q_4\}$. Consecuentemente, nos quedan cuatro conjuntos:

$$
\begin{aligned}
&[q_0] = \{q_0, q_3, q_5\} \\
&[q_1] = \{q_1, q_6\} \\
&[q_2] = \{q_2\} \\
&[q_4] = \{q_4\}
\end{aligned}
$$

IMPORTANTE: Es importante darse cuenta de que aunque q_1 y q_6 tengan el mismo comportamiento que el estado q_2, no hay que ponerlos en el mismo conjunto. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar.

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_1 leyendo el símbolo 0 se pasa al estado q_5, en vez de q_5 vamos a poner su grupo, es decir, $[q_0]$. Procederemos de la misma forma en todos los casos:

	0	1
q_0	$[q_1]$	$[q_4]$
q_1	$[q_0]$	$[q_2]$
q_2	$[q_0]$	$[q_2]$
q_3	$[q_1]$	$[q_4]$
q_4	$[q_2]$	$[q_0]$
q_5	$[q_0]$	$[q_0]$
q_6	$[q_0]$	$[q_2]$

7.7.6.3 Tercera partición

Como los conjuntos $[q_2]$ y $[q_4]$ solo tienen un elemento, no se pueden partir. Vamos a centrarnos en averiguar si hay que partir los conjuntos $[q_0]$ y $[q_1]$.

En el conjunto $[q_0]$ vemos dos comportamientos distintos. Por una parte, q_0 y q_3 tienen el mismo comportamiento: desde q_0 leyendo el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_1]$ y leyendo el símbolo 1, se pasa a un estado que pertenece al conjunto $[q_4]$; en cambio, desde q_5 tanto con el símbolo 0 como con el símbolo 1 se pasa a un estado que pertenece al conjunto $[q_0]$. Esto significa que hay que partir el conjunto $[q_0]$ en dos: $[q_0] = \{q_0, q_3\}$ y $[q_5] = \{q_5\}$. En el conjunto $[q_1]$ vemos que los estados q_1 y q_6 tienen el mismo comportamiento: cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_0]$ y cuando se lee el símbolo 1, se pasa a un estado que pertenece al conjunto $[q_2]$. Por tanto, no hay que partir el conjunto $[q_1]$. Consecuentemente, nos quedan cinco conjuntos:

$$
\begin{aligned}
[q_0] &= \{q_0, q_3\} \\
[q_1] &= \{q_1, q_6\} \\
[q_2] &= \{q_2\} \\
[q_4] &= \{q_4\} \\
[q_5] &= \{q_5\}
\end{aligned}
$$

IMPORTANTE: Es importante darse cuenta de que aunque q_1 y q_6 tengan el mismo comportamiento que el estado q_2, no hay que ponerlos en el mismo conjunto. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar.

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_4 leyendo el símbolo 0 se pasa al estado q_3, en vez de q_3 vamos a poner su grupo, es decir, $[q_1]$. Procederemos de la misma forma en todos los casos:

	0	1
q_0	$[q_1]$	$[q_4]$
q_1	$[q_5]$	$[q_2]$
q_2	$[q_0]$	$[q_2]$
q_3	$[q_1]$	$[q_4]$
q_4	$[q_2]$	$[q_5]$
q_5	$[q_5]$	$[q_0]$
q_6	$[q_5]$	$[q_2]$

7.7.6.4 No hay cuarta partición

Los conjuntos $[q_2]$, $[q_4]$ y $[q_5]$ no se pueden partir porque son unitarios, es decir, cada uno de ellos contiene un único elemento, por lo cual, solo nos queda averiguar si hay que partir los conjuntos $[q_0]$ y $[q_1]$.

En el conjunto $[q_0]$ tenemos que q_0 y q_3 tienen el mismo comportamiento: desde q_0 leyendo el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_1]$ y leyendo el símbolo 1, se pasa a un estado que pertenece al conjunto $[q_4]$. Esto significa que no hay que partir el conjunto $[q_0]$. En el conjunto $[q_1]$ tenemos que los estados q_1 y q_6 tienen el mismo comportamiento: cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_5]$ y cuando se lee el símbolo 1, se pasa a un estado que pertenece al conjunto $[q_2]$. Por tanto, no hay que partir el conjunto $[q_1]$. Consecuentemente, no surge ninguna partición nueva.

7.7.6.5 AFDC minimal

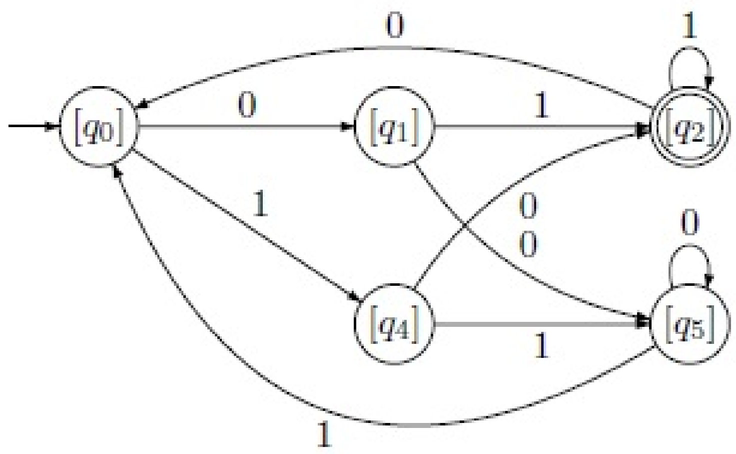
Los conjuntos definitivos son estos cinco:

$$\begin{aligned}
&[q_0] = \{q_0, q_3\}\\
&[q_1] = \{q_1, q_6\}\\
&[q_2] = \{q_2\}\\
&[q_3] = \{q_4\}\\
&[q_5] = \{q_5\}
\end{aligned}$$

Cada conjunto representa un estado. Teniendo en cuenta la última tabla, se calculan las transiciones:

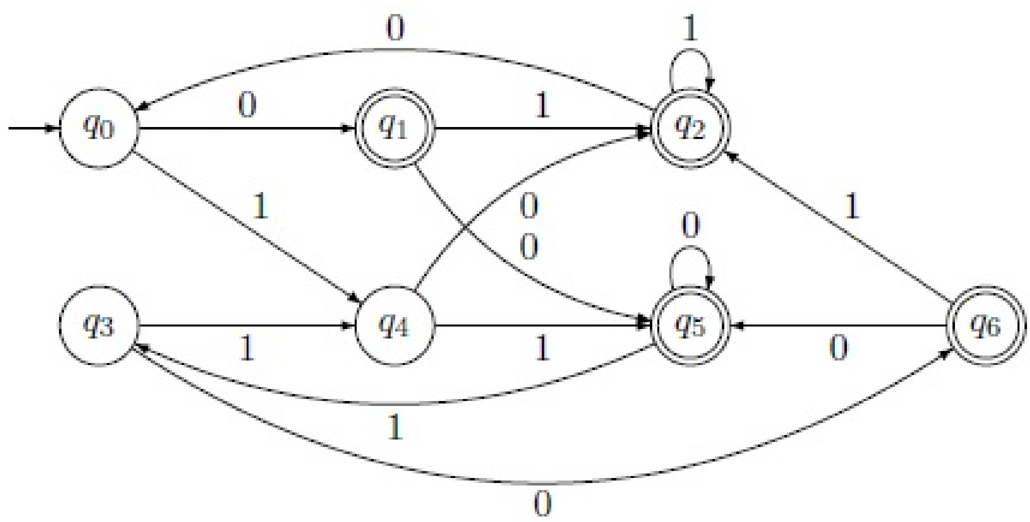
	0	1
q_0	$[q_1]$	$[q_4]$
q_1	$[q_5]$	$[q_2]$
q_2	$[q_0]$	$[q_2]$
q_4	$[q_2]$	$[q_5]$
q_5	$[q_5]$	$[q_0]$

$[q_0]$ será el estado inicial porque contiene al estado inicial q_0 del AFDC original. Por otra parte, $[q_2]$ será el estado con doble círculo porque está formado por el único estado que pertenece al conjunto Y del AFDC original.



7.7.7 Minimización: Ejercicio 7

Minimizar el siguiente AFDC:



La tabla correspondiente a la función de transición $\tau_{\mathbb{A}}$ es la siguiente:

$\tau_{\mathbb{A}}$	0	1
q_0	q_1	q_4
q_1	q_5	q_2
q_2	q_0	q_2
q_3	q_6	q_4
q_4	q_2	q_5
q_5	q_5	q_3
q_6	q_5	q_2

7.7.7.1 Primera partición

La primera partición coloca en un conjunto los estados que pertenecen a Y (los estados que responden “Sí”) y en otro conjunto los que no pertenecen a Y (los estados que responden “No”). A cada conjunto lo llamaremos $[q_j]$ donde j es el menor de los índices que aparecen en el conjunto en cuestión.

$$[q_0] = \{q_0, q_3, q_4\}$$
$$[q_1] = \{q_1, q_2, q_5, q_6\}$$

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_3 leyendo el símbolo 0 se pasa al estado q_6, en vez de q_6 vamos a poner su grupo, es decir, $[q_1]$. Procederemos de la misma forma en todos los casos:

	0	1
q_0	$[q_1]$	$[q_0]$
q_1	$[q_1]$	$[q_1]$
q_2	$[q_0]$	$[q_1]$
q_3	$[q_1]$	$[q_0]$
q_4	$[q_1]$	$[q_1]$
q_5	$[q_1]$	$[q_0]$
q_6	$[q_1]$	$[q_1]$

7.7.7.2 Segunda partición

Vamos a comprobar si hay que partir los conjuntos $[q_0]$ y $[q_1]$.

En el conjunto $[q_0]$ vemos dos comportamientos diferentes. En primer lugar, los estados q_0 y q_3 tienen el mismo comportamiento: cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_1]$ y cuando se lee el símbolo 1 se pasa a un estado que pertenece al conjunto $[q_0]$; en segundo lugar, el estado q_4 muestra un comportamiento distinto con respecto al comportamiento de los estados q_0 y q_3: tanto cuando se lee el símbolo 0 como cuando se lee el símbolo 1 se pasa a un estado que pertenece al conjunto $[q_1]$. Esto significa que hay que partir el conjunto $[q_0]$ en dos conjuntos: $[q_0] = \{q_0, q_3\}$ y $[q_4] = \{q_4\}$. Por su parte, en el conjunto $[q_1]$ vemos tres comportamientos diferentes. En primer lugar, tanto cuando se lee el símbolo 0 como cuando se lee el símbolo 1 se pasa a un estado que pertenece al conjunto $[q_1]$; en segundo lugar, el estado q_2 muestra un comportamiento distinto con respecto al comportamiento del resto de los estados de $[q_1]$: cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_0]$ y cuando se lee el símbolo 1 se pasa a un estado que pertenece al conjunto $[q_1]$; en tercer lugar, el estado q_5 muestra un tercer comportamiento distinto: cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_1]$ y cuando se lee el símbolo 1 se pasa a un estado que pertenece al conjunto $[q_0]$. Esto significa que hay que partir el conjunto $[q_1]$ en tres conjuntos: $[q_1] = \{q_1, q_6\}$, $[q_2] = \{q_2\}$ y $[q_5] = \{q_5\}$. Consecuentemente, nos quedan cinco conjuntos:

$$\begin{array}{l}
[q_0] = \{q_0, q_3\} \\
[q_1] = \{q_1, q_6\} \\
[q_2] = \{q_2\} \\
[q_4] = \{q_4\} \\
[q_5] = \{q_5\}
\end{array}$$

IMPORTANTE: Es importante darse cuenta de que aunque q_1 y q_6 tengan el mismo comportamiento que el estado q_4, no hay que ponerlos en el mismo conjunto. Ocurre lo mismo con q_0 y q_3 por un lado y q_5 por otro lado, es decir, no hay que ponerlos en el mismo conjunto porque q_5 pertenece a otro conjunto desde la primera partición. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar.

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_3 leyendo el símbolo 0 se pasa al estado q_6, en vez de q_6 vamos a poner su grupo, es decir, $[q_1]$. Procederemos de la misma forma en todos los casos:

	0	1
q_0	$[q_1]$	$[q_4]$
q_1	$[q_5]$	$[q_2]$
q_2	$[q_0]$	$[q_2]$
q_3	$[q_1]$	$[q_4]$
q_4	$[q_2]$	$[q_5]$
q_5	$[q_5]$	$[q_0]$
q_6	$[q_5]$	$[q_2]$

7.7.7.3 No hay tercera partición

Como los conjuntos $[q_2]$, $[q_4]$ y $[q_5]$ solo tienen un elemento, no se pueden partir. Vamos a centrarnos en averiguar si hay que partir los conjuntos $[q_0]$ y $[q_1]$.

En el conjunto $[q_0]$ vemos que q_0 y q_3 tienen el mismo comportamiento: desde q_0 leyendo el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_1]$ y leyendo el símbolo 1, se pasa a un estado que pertenece al conjunto $[q_4]$. Esto significa que no hay que partir el conjunto $[q_0]$. En el conjunto $[q_1]$ vemos que los estados q_1 y q_6 tienen el mismo comportamiento: cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_5]$ y cuando se lee el símbolo 1, se pasa a un estado que pertenece al conjunto $[q_2]$. Lo cual quiere decir que no hay que partir el conjunto $[q_1]$. Por tanto, no surge ninguna partición nueva.

7.7.7.4 AFDC minimal

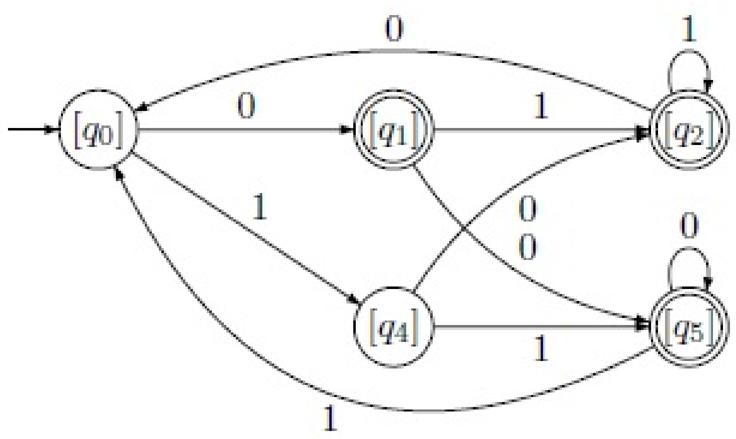
Los conjuntos definitivos son estos cinco:

$$
\begin{aligned}
[q_0] &= \{q_0, q_3\} \\
[q_1] &= \{q_1, q_6\} \\
[q_2] &= \{q_2\} \\
[q_4] &= \{q_4\} \\
[q_5] &= \{q_5\}
\end{aligned}
$$

Cada conjunto representa un estado. Teniendo en cuenta la última tabla, se calculan las transiciones:

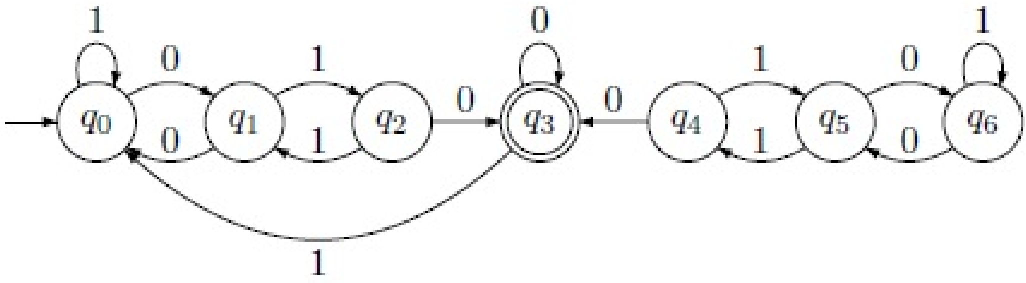
	0	1
q_0	$[q_1]$	$[q_4]$
q_1	$[q_5]$	$[q_2]$
q_2	$[q_0]$	$[q_2]$
q_4	$[q_2]$	$[q_5]$
q_5	$[q_5]$	$[q_0]$

$[q_0]$ será el estado inicial porque contiene al estado inicial q_0 del AFDC original. Por otra parte, $[q_1]$, $[q_2]$ y $[q_5]$ serán los estados con doble círculo porque están formados por estados que pertenecen al conjunto Y del AFDC original.



7.7.8 Minimización: Ejercicio 8

Minimizar el siguiente AFDC:



La tabla correspondiente a la función de transición $\tau_{\mathbb{A}}$ es la siguiente:

$\tau_{\mathbb{A}}$	0	1
q_0	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_3	q_0
q_4	q_3	q_5
q_5	q_6	q_4
q_6	q_5	q_6

7.7.8.1 Primera partición

La primera partición siempre genera dos conjuntos. Por una parte, ponemos en un conjunto los estados que pertenecen a Y (los estados que responden "Sí") y, por otra parte, ponemos en otro conjunto los que no pertenecen a Y (los estados que responden "No"). A cada conjunto lo llamaremos $[q_j]$ donde j es el menor de los índices que aparecen en el conjunto en cuestión.

$$[q_0] = \{q_0, q_1, q_2, q_4, q_5, q_6\}$$
$$[q_3] = \{q_3\}$$

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_2 leyendo el símbolo 1 se pasa al estado q_1, en vez de q_1 vamos a poner su grupo, es decir, $[q_0]$. Procederemos de la misma forma en todos los casos:

	0	1
q_0	$[q_0]$	$[q_0]$
q_1	$[q_0]$	$[q_0]$
q_2	$[q_3]$	$[q_0]$
q_3	$[q_3]$	$[q_0]$
q_4	$[q_3]$	$[q_0]$
q_5	$[q_0]$	$[q_0]$
q_6	$[q_0]$	$[q_0]$

7.7.8.2 Segunda partición

Vamos a comprobar si hay que partir los conjuntos $[q_0]$ y $[q_3]$.

Puesto que el conjunto $[q_3]$ contiene solo el estado q_3, se queda como está. En el conjunto $[q_0]$ vemos dos comportamientos diferentes: en primer lugar, los estados q_0, q_1, q_5 y q_6 tienen el mismo comportamiento: tanto cuando se lee el símbolo 0 como cuando se lee el símbolo 1 se pasa a un estado que pertenece al conjunto $[q_0]$; en segundo lugar, los estados q_2 y q_4 tienen el mismo comportamiento pero distinto con respecto al comportamiento de q_0, q_1, q_5 y q_6: cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_3]$ y cuando se lee el símbolo 1 se pasa a un estado que pertenece al conjunto $[q_0]$. Esto significa que hay que partir el conjunto $[q_0]$ en dos conjuntos: $[q_0] = \{q_0, q_1, q_5, q_6\}$ y $[q_2] = \{q_2, q_4\}$. Consecuentemente, nos quedan tres conjuntos:

$$
\begin{aligned}
&[q_0] = \{q_0, q_1, q_5, q_6\} \\
&[q_2] = \{q_2, q_4\} \\
&[q_3] = \{q_3\}
\end{aligned}
$$

IMPORTANTE: Es importante darse cuenta de que aunque q_2 y q_4 tengan el mismo comportamiento que el estado q_3, no hay que ponerlos en el mismo conjunto. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar.

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_4 leyendo el símbolo 1 se pasa al estado q_5, en vez de q_5 vamos a poner su grupo, es decir, $[q_0]$. Procederemos de la misma forma en todos los casos:

	0	1
q_0	$[q_0]$	$[q_0]$
q_1	$[q_0]$	$[q_2]$
q_2	$[q_3]$	$[q_0]$
q_3	$[q_3]$	$[q_0]$
q_4	$[q_3]$	$[q_0]$
q_5	$[q_0]$	$[q_2]$
q_6	$[q_0]$	$[q_0]$

7.7.8.3 Tercera partición

Como el conjunto $[q_3]$ solo tiene un elemento, no se puede partir. Vamos a centrarnos en averiguar si hay que partir los conjuntos $[q_0]$ y $[q_2]$.

En el conjunto $[q_0]$ vemos dos comportamientos distintos. Por una parte, q_0 y q_6 tienen el mismo comportamiento: tanto con el símbolo 0 como con el símbolo 1 se pasa a un estado que pertenece al conjunto $[q_0]$; en cambio, desde los estados q_1 y q_5 cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_0]$ y cuando se lee el símbolo 1, se pasa a un estado que pertenece al conjunto $[q_2]$. Esto significa que hay que partir el conjunto $[q_0]$ en dos: $[q_0] = \{q_0, q_6\}$ y $[q_1] = \{q_1, q_5\}$. En el conjunto $[q_2]$ vemos que los estados q_2 y q_4 tienen el mismo comportamiento: cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_3]$ y cuando se lee el símbolo 1, se pasa a un estado que pertenece al conjunto $[q_0]$. Por tanto, no hay que partir el conjunto $[q_2]$. Consecuentemente, nos quedan cuatro conjuntos:

$$
\begin{aligned}
&[q_0] = \{q_0, q_6\}\\
&[q_1] = \{q_1, q_5\}\\
&[q_2] = \{q_2, q_4\}\\
&[q_3] = \{q_3\}
\end{aligned}
$$

IMPORTANTE: Es importante darse cuenta de que aunque q_0 y q_6 tengan el mismo comportamiento que el estado q_3, no hay que ponerlos en el mismo conjunto. El método consiste en ir partiendo los conjuntos en subconjuntos, pero una vez separados, nunca se han de mezclar.

Ahora vamos a recalcular esa tabla, pero poniendo grupos en vez de estados concretos. Así, como del estado q_4 leyendo una b se pasa al estado q_5, en vez de q_5 vamos a poner su grupo, es decir, $[q_1]$. Procederemos de la misma forma en todos los casos:

	0	1
q_0	$[q_1]$	$[q_0]$
q_1	$[q_0]$	$[q_2]$
q_2	$[q_3]$	$[q_1]$
q_3	$[q_3]$	$[q_0]$
q_4	$[q_3]$	$[q_1]$
q_5	$[q_0]$	$[q_2]$
q_6	$[q_1]$	$[q_0]$

7.7.8.4 No hay cuarta partición

El conjunto $[q_3]$ no se puede partir porque es unitario, es decir, contiene un único elemento, por lo cual, solo nos queda averiguar si hay que partir los conjuntos $[q_0]$, $[q_1]$ y $[q_2]$.

En el conjunto $[q_0]$ tenemos que q_0 y q_6 tienen el mismo comportamiento: desde q_0 leyendo el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_1]$ y leyendo el símbolo 1, se pasa a un estado que pertenece al conjunto $[q_0]$. Esto significa que no hay que partir el conjunto $[q_0]$.

En el conjunto $[q_1]$ tenemos que los estados q_1 y q_5 tienen el mismo comportamiento: cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_0]$ y cuando se lee el símbolo 1, se pasa a un estado que pertenece al conjunto $[q_2]$. Por tanto, no hay que partir el conjunto $[q_1]$. Por último, en el conjunto $[q_2]$ tenemos que los estados q_2 y q_4 tienen el mismo comportamiento: cuando se lee el símbolo 0 se pasa a un estado que pertenece al conjunto $[q_3]$ y cuando se lee el símbolo 1, se pasa a un estado que pertenece al conjunto $[q_1]$. Esto quiere decir que tampoco hay que partir el conjunto $[q_2]$. Consecuemtemente, no surge ninguna partición nueva.

7.7.8.5 AFDC minimal

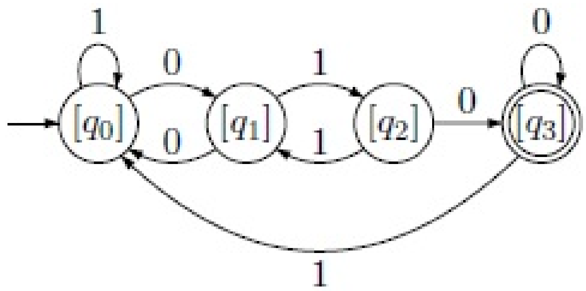
Los conjuntos definitivos son estos cinco:

$$
\begin{aligned}
&[q_0] = \{q_0, q_6\} \\
&[q_1] = \{q_1, q_5\} \\
&[q_2] = \{q_2, q_4\} \\
&[q_3] = \{q_3\}
\end{aligned}
$$

Cada conjunto representa un estado. Teniendo en cuenta la última tabla, se calculan las transiciones:

	0	1
q_0	$[q_1]$	$[q_0]$
q_1	$[q_0]$	$[q_2]$
q_2	$[q_3]$	$[q_1]$
q_3	$[q_3]$	$[q_0]$

$[q_0]$ será el estado inicial porque contiene al estado inicial q_0 del AFDC original. Por otra parte, $[q_3]$ será el estado con doble círculo porque está formado por el único estado que pertenece al conjunto Y del AFDC original.



7.8.

Símbolos griegos

Es habitual utilizar símbolos del alfabeto griego para representar fórmulas de la lógica matemática y también para denominar funciones. Debido a ello, en la tabla 7.8.1 de la página 124, se muestran los símbolos griegos que se pudieran utilizar.

Alfabeto griego moderno

	Mayúscula	minúscula	Nombre (en castellano)
1	$A, \boldsymbol{A}$	$\alpha, \boldsymbol{\alpha}$	alfa
2	$B, \boldsymbol{B}$	$\beta, \boldsymbol{\beta}$	beta
3	$\Gamma, \boldsymbol{\mathit{\Gamma}}$	$\gamma, \boldsymbol{\gamma}$	gamma
4	$\Delta, \boldsymbol{\mathit{\Delta}}$	$\delta, \boldsymbol{\delta}$	delta
5	$E, \boldsymbol{E}$	$\epsilon, \boldsymbol{\epsilon}, \varepsilon, \boldsymbol{\varepsilon}$	épsilon
6	$Z, \boldsymbol{Z}$	$\zeta, \boldsymbol{\zeta}$	dseta
7	$H, \boldsymbol{H}$	$\eta, \boldsymbol{\eta}$	eta
8	$\Theta, \boldsymbol{\mathit{\Theta}}$	$\theta, \boldsymbol{\theta}, \vartheta$	ceta
9	$I, \boldsymbol{I}$	$\iota, \boldsymbol{\iota}$	iota
10	$K, \boldsymbol{K}$	$\kappa, \boldsymbol{\kappa}, \varkappa, \boldsymbol{\varkappa}$	kappa
11	$\Lambda, \boldsymbol{\mathit{\Lambda}}$	$\lambda, \boldsymbol{\lambda}$	lambda
12	$M, \boldsymbol{M}$	$\mu, \boldsymbol{\mu}$	mu
13	$N, \boldsymbol{N}$	$\nu, \boldsymbol{\nu}$	nu
14	$\Xi, \boldsymbol{\mathit{\Xi}}$	$\xi, \boldsymbol{\xi}$	ksi
15	$O, \boldsymbol{O}$	$o, \boldsymbol{o}$	ómicron
16	$\Pi, \boldsymbol{\mathit{\Pi}}$	$\pi, \boldsymbol{\pi}, \varpi$	pi
17	$P, \boldsymbol{P}$	$\rho, \boldsymbol{\rho}, \varrho, \boldsymbol{\varrho}$	ro
18	$\Sigma, \boldsymbol{\mathit{\Sigma}}$	$\sigma, \boldsymbol{\sigma}, \varsigma$	sigma
19	$T, \boldsymbol{T}$	$\tau, \boldsymbol{\tau}$	tau
20	$\Upsilon, \boldsymbol{\mathit{\Upsilon}}$	$\upsilon, \boldsymbol{\upsilon}$	ípsilon
21	$\Phi, \boldsymbol{\mathit{\Phi}}$	$\phi, \boldsymbol{\phi}, \varphi, \boldsymbol{\varphi}$	fi
22	$X, \boldsymbol{X}$	$\chi, \boldsymbol{\chi}$	ji
23	$\Psi, \boldsymbol{\mathit{\Psi}}$	$\psi, \boldsymbol{\psi}$	psi
24	$\Omega, \boldsymbol{\mathit{\Omega}}$	$\omega, \boldsymbol{\omega}$	omega

Letras griegas no pertenecientes al alfabeto griego moderno

	Mayúscula	minúscula	Nombre (en castellano)
25	Ϝ	ϝ	digamma
26	Ϙ	ϙ	qoppa
27	Ϟ	ϟ	koppa
28	Ϡ	ϡ	sampi
29	Ϛ	ϛ	estigma

Tabla 7.8.1. Símbolos griegos.