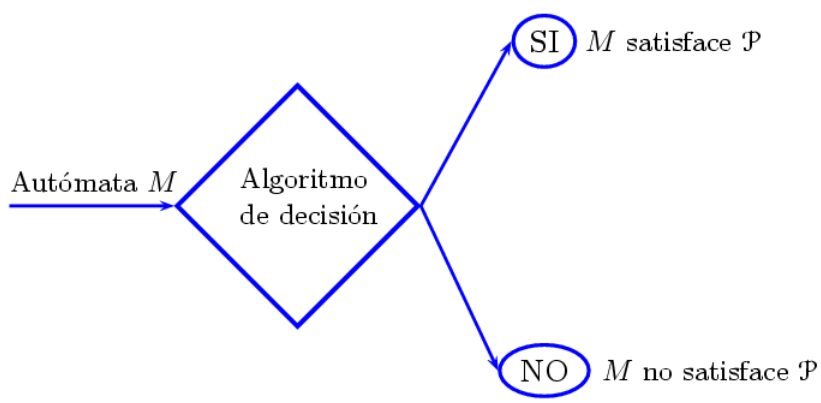
3.6. Algoritmos de decisión

Dada una propiedad $\mathcal{P}$, referente a autómatas sobre un alfabeto Σ, un **problema de decisión para $\mathcal{P}$** consiste en buscar un algoritmo[1], aplicable a un autómata arbitrario M, que responda SI o NO a la pregunta: ¿satisface M la propiedad $\mathcal{P}$? El siguiente diagrama ilustra la situación.



Si existe un algoritmo de decisión, se dice que el problema $\mathcal{P}$ es **decidible** o **resoluble**; en caso contrario, el problema $\mathcal{P}$ es **indecidible** o **irresoluble**.

Es importante tener presente que, para que un problema $\mathcal{P}$ sea decidible, no basta responder SI o NO a la pregunta "¿satisface M la propiedad $\mathcal{P}$?" para uno o varios autómatas aislados M; es necesario exhibir un algoritmo aplicable a *todos* los autómatas. Es posible que en algunos casos concretos se pueda decidir afirmativa o negativamente sobre la satisfabilidad de una propiedad $\mathcal{P}$ y, sin embargo, el problema $\mathcal{P}$ sea indecidible.

El primer problema de decisión que consideraremos es el problema de decidir si un autómata acepta o no *alguna* cadena.

3.6.1 Teorema. *Sea Σ un alfabeto dado. Existe un algoritmo para el siguiente problema de decisión referente a autómatas sobre Σ:*

> *Dado un autómata (AFD o AFN o AFN-λ) M, ¿Es $L(M) \neq \varnothing$? (es decir, ¿acepta M alguna cadena?).*

[1]La noción de algoritmo aquí considerada es la corriente: un algoritmo es un procedimiento claro y preciso para efectuar una determinada operación secuencialmente, en uno o más pasos.

Demostración: Podemos diseñar un algoritmo sencillo para encontrar los estados que son accesibles (o alcanzables) desde el estado inicial de M, es decir, los estados para los cuales existen trayectorias desde el estado inicial. Si algún estado de aceptación es alcanzable, se tendrá $L(M) \neq \varnothing$; en caso contrario $L(M) = \varnothing$. En el siguiente recuadro aparece el algoritmo para encontrar el conjunto **ALC** de estados alcanzables:

INICIALIZAR:

$\quad$ **ALC** $:= \{q_0\}$, donde q_0 es el estado inicial de M.

REPETIR:

$\quad$ Para cada $q \in$ **ALC** buscar los arcos que salen de q y añadir a **ALC** los estados p para los cuales haya un arco de q a p con cualquier etiqueta (puede ser λ).

HASTA:

$\quad$ No se añaden nuevos estados a **ALC**.

Claramente, el autómata M acepta alguna cadena (o sea, $L(M) \neq \varnothing$) si y solo si **ALC** contiene algún estado de aceptación. $\square$

3.6.2 Corolario. *Sea Σ un alfabeto dado. Existen algoritmos para los siguientes problemas de decisión referentes a autómatas sobre Σ:*

(1) *Dados dos autómatas M_1 y M_2 (AFD o AFN o AFN-λ), ¿$L(M_1) \subseteq L(M_2)$?*

(2) *Dados dos autómatas M_1 y M_2 (AFD o AFN o AFN-λ), ¿$L(M_1) = L(M_2)$?*

Demostración:

(1) Sea $L_1 = L(M_1)$ y $L_2 = L(M_2)$. Se tiene

$$L_1 \subseteq L_2 \Longleftrightarrow L_1 - L_2 = \varnothing.$$

Algoritmo: construir un AFD M' que acepte el lenguaje $L_1 - L_2$ (esto se puede hacer en razón de la parte (7) del teorema 3.3.1). Utilizar luego el procedimiento de la parte (1) para decidir si $L_1 - L_2$ es o no vacío.

(2) $L(M_1) = L(M_2) \Longleftrightarrow L(M_1) \subseteq L(M_2)$ y $L(M_2) \subseteq L(M_1)$. Por lo tanto, basta aplicar dos veces el algoritmo de la parte (1). También se puede encontrar un algoritmo de decisión teniendo en cuenta que $L_1 = L_2 \Longleftrightarrow L_1 \lhd L_2 = \varnothing$ junto con la parte (8) del teorema 3.3.1. $\square$

El argumento utilizado en la demostración del lema de bombeo sirve para establecer un criterio que permite decidir si el lenguaje aceptado por un autómata es o no infinito. El criterio aparece en el siguiente teorema.

3.6.3 Teorema. *Sea M un AFD con n estados y sea $L = L(M)$. L es infinito si y solo si M acepta una cadena w tal que $n \leq |w| < 2n$.*

Demostración: Si $w \in M$ y $n \leq |w| < 2n$, entonces por la demostración del lema de bombeo, w se puede descomponer como $w = uvx$, donde $|uv| \leq n$, $v \neq \lambda$. M acepta infinitas cadenas: $uv^i x$ para todo $i \geq 0$.

Recíprocamente, si L es infinito, existe $w \in L$ con $|w| \geq n$. Por la demostración del lema de bombeo, $w = uvx$ donde $|uv| \leq n$, $v \neq \lambda$. Si $|w| < 2n$ la demostración está terminada. Si $|w| \geq 2n$, puesto que $|v| \leq |uv| \leq n$, se tendrá $|ux| \geq n$ y $ux \in L$. De nuevo, si $|ux| < 2n$, la demostración termina; en caso contrario, se prosigue de esta forma hasta encontrar una cadena en L cuya longitud ℓ satisfaga $n \leq \ell < 2n$. □

3.6.4 Corolario. *Sea Σ un alfabeto dado. Existe un algoritmo para el siguiente problema de decisión referente a autómatas sobre Σ:*

Dado un autómata M (AFD o AFN o AFN-λ), ¿Es $L(M)$ infinito?

Demostración: Algoritmo: construir un AFD M' que acepte el lenguaje $L(M)$ y chequear la aceptación o rechazo de todas las cadenas $w \in \Sigma^*$ tales que $n \leq |w| < 2n$, donde $n = \#$ de estados de M'. □

Hemos trabajado con problemas de decisión referentes a autómatas, pero también podemos considerar problemas sobre lenguajes regulares. Dada una propiedad $\mathcal{P}$, referente a lenguajes regulares sobre un alfabeto Σ, un problema de decisión para $\mathcal{P}$ consiste en buscar un algoritmo, aplicable a todo lenguaje regular L (es decir, a toda expresión regular R), que responda SI o NO a la pregunta: ¿satisface L la propiedad $\mathcal{P}$? Puesto que conocemos algoritmos de conversión entre la representación por expresiones regulares y la representación por autómatas, los problemas decidibles sobre autómatas corresponden a problemas decidibles sobre lenguajes regulares y viceversa. Así por ejemplo, en razón del Corolario 3.6.4, el siguiente problema es decidible: "Dada una expresión regular R, ¿es $L(R)$ infinito?"

Ejercicios de la sección 3.6

1. Sea Σ un alfabeto dado. Encontrar algoritmos para los siguientes problemas de decisión referentes a autómatas sobre Σ:

(i) Dado un autómata M (AFD o AFN o AFN-λ), ¿es $L(M) = \Sigma^*$?

(ii) Dado un autómata M (AFD o AFN o AFN-λ), ¿acepta M alguna cadena de longitud 1250?

(iii) Dado un autómata M (AFD o AFN o AFN-λ), ¿acepta M alguna cadena de longitud mayor que 1250?

(iv) Dado un autómata M (AFD o AFN o AFN-λ), ¿acepta M *todas* las cadenas de longitud impar?

(v) Dados dos autómatas M_1 y M_2 (AFD o AFN o AFN-λ), ¿existe alguna cadena que no sea aceptada por ninguno de los autómatas M_1 y M_2?

(vi) Dado un autómata M (AFD o AFN o AFN-λ), ¿Es $L(M)$ cofinito? (Un conjunto es **cofinito** si su complemento es finito).

2. Encontrar algoritmos de decisión, que no utilicen autómatas, para resolver los siguientes problemas:

(i) Dada una expresión regular R, ¿es $L(R) \neq \varnothing$? Ayuda: puesto que las expresiones regulares se definen recursivamente, el algoritmo requiere descomponer la expresión R y utilizar criterios de vacuidad para la unión, la concatenación y la estrella de Kleene.

(ii) Dada una expresión regular R, ¿contiene $L(R)$ por lo menos 150 cadenas?

✎ Al considerar problemas de decisión lo importante es la *existencia* o no de algoritmos de decisión, no la *eficiencia* de los mismos.

✎ En el Capítulo 6 se mostrará que existen problemas indecidibles, es decir, problemas para los cuales no hay algoritmos de decisión.