Intersección

- La intersección de dos AFD es un nuevo AFD que reconoce solamente las palabras reconocidas por ambos autómatas iniciales
- Formalmente si $M_1 = \{Q_1, \Sigma, T_1, q_1, F_1\}$ y $M_2 = \{Q_2, \Sigma, T_2, q_2, F_2\}$
- Entonces $M = M_1 \cap M_2$ se forma con los pares ordenados del primer autómata con el segundo, es decir:
- $Q = Q_1 x Q_2$
- $q_0 = (q_1, q_2)$
- $F = F_1 x F_2$
- $T((p,q),x) = (T_1(p,x), T_2(q,x)) \quad \forall\ p \in Q_1 \wedge \forall\ q \in Q_2$

Método

- **Observación**: Tener en cuenta que al ser pares ordenados, si un estado del par es erróneo, el par también lo es.

1 Obtener el estado inicial: es el par ordenado con los dos estados iniciales originales. Colocarlo en la primer fila.

2 Obtener para cada carácter del alfabeto el par ordenado hacia donde se llega desde el estado del que se partió, usando la TT correspondiente para cada componente del par. (Ojo, no olvidar la observación de arriba)

3 Se agrega cada nuevo estado no erróneo (par ordenado) y se repite el proceso.

4 Los estados finales son aquellos donde **AMBOS** elementos del par ordenado sean finales en sus autómatas originales.

Ejemplo

TT1	a	b	c
0-	1	0	1
1+	-	1	1

TT2	a	b	c
2-	2	-	3
3+	2	3	-

Renombrando los estados

0 ←
1 ←
2 ←

TT∩	a	b	c
(0,2)-	(1,2)	-	(1,3)
(1,2)	-	-	(1,3)
(1,3)+	-	(1,3)	-

TT∩	a	b	c
0-	1	-	2
1	-	-	2
2+	-	2	-

Complemento

- El complemento de un lenguaje se hace con respecto al lenguaje universal (Σ^*)
- Es decir que el autómata del complemento debe reconocer todas las cadenas de Σ^* que NO son reconocidas por el autómata original y rechazar las que si eran reconocidas originalmente.
- Para ello basta con **"invertir"** los estados finales. Formalmente si
  $M = (Q, \Sigma, T, q_0, F)$ entonces el complemento será
  $M^c = (Q, \Sigma, T, q_0, Q - F)$
- Tener en cuenta que si hay transiciones no definidas estaría rechazando antes de terminar de analizar la cadena. Por eso es necesario **COMPLETAR** la tabla de transición **ANTES** de complementar. De este modo el estado erróneo podrá ser tomado como estado final.

Ejemplo

Tabla Original

TT	a	b	c
0-	1	-	2
1	-	-	2
2+	-	2	-

Completando

TT	a	b	c
0-	1	3	2
1	3	3	2
2+	3	2	3
3	3	3	3

Invirtiendo Estados Finales

TT^c	a	b	c
0±	1	3	2
1+	3	3	2
2	3	2	3
3+	3	3	3

AFD Mínimo

- Dado un LR hay varios AFD equivalentes que lo reconocen. El ADF mínimo es aquel con la menor cantidad de estados.
- Se puede demostrar que dado un LR hay un único autómata mínimo que lo reconoce, por eso encontrarlo permite
  - Saber si dos AFD son equivalentes (reconocen el mismo lenguaje)

    Implica, via conversión a AFD saber si dos ER son equivalentes
  - Optimizar el reconocimiento de un LR dado

AFD Mínimo - Conceptos

- El algoritmo que se explica en el libro de la cátedra es el de Moore, el cual se basa en el teorema de Myhill-Nerode que caracteriza los LR a partir de relaciones de equivalencia.
  - Nota: varios algoritmos se basan en relaciones de equivalencia para obtener el mínimo. El de Hopcroft es de los más conocidos por tener peor caso O(n log n)
- La idea es establecer una partición basado en la relación de equivalencia más gruesa (y obvia) posible y luego ir refinando esta relación hasta obtener la más fina. Esa nos da el AFD mínimo

AFD Mínimo – Conceptos

- La primer partición consiste en agrupar los estados en dos clases, según sean estados finales o no.
- Para cada estado hacemos “un salto” de un solo carácter para refinar la partición. Solo quedarán en la misma clase los estados que se comporten igual (indistinguible a nivel de clase)
- Al no poder refinar más la partición se tiene el AFD mínimo.

El algoritmo – parte previa

1. El primer paso, previo al algoritmo en si, es eliminar los estados inalcanzables, ya que el refinamiento de particiones los dejará en un clase separada (e inalcanzable) pero no los eliminará.
2. Para acelerar el proceso podemos eliminar los estados “equivalentes inmediatos” es decir aquellos que tienen exactamente el mismo comportamiento y ambos son finales o no finales.
   a. Al reducir, por convención, nos quedamos en el estado de menor número y luego renombramos en toda la tabla
   b. Puede que al reducir aparezcan otros estados equivalentes, seguir reduciendo hasta que no haya más estados equivalentes inmediatos

El algoritmo

3. Armo la TTC (tabla de transición por clases) reemplazando cada estado por el nombre de su clase y si dentro de una clase hay estados de distinto comportamiento los separo en clases nuevas
4. Rearmo la TTC tomando en cuenta las nuevas clases y vuelvo a controlar si hay que si hay que separar en nuevas clases.
5. Repito hasta que no surjan nuevas clases.
6. De cada clase tomo un estado como representante de la misma y elimino el resto
   a. Como con los inmediatos puedo tomar el más bajo y reemplazar e todos lados
   b. Alternativamente puedo tomar el número de clase como número de estado

Ejemplo

Tabla Original

TT	a	b
0-	2	4
1	5	3
2	2	3
3+	5	2
4	4	0
5+	6	2
6+	5	2

Inalcanzable → **1**

Equivalentes → **3+** y **6+**

Separando en Finales y no Finales

Clase	TT	a	b
C_0	**0-**	2	4
	2	2	3
	4	4	0
C_1	**3+**	5	2
	5+	3	2

El estado 6 se eliminó y es representado por el estado 3

Ejemplo

Tabla de Transición Inicial

Clase	TT	a	b
C_0	**0-**	2	4
	2	2	3
	4	4	0
C_1	**3+**	5	2
	5+	3	2

Tabla de Transición por Clases

Clase	TT	a	b
C_0	**0-**	C_0	C_0
	2	C_0	C_1
	4	C_0	C_0
C_1	**3+**	C_1	C_0
	5+	C_1	C_0

Separo el estado 2 en nueva clase y rearmo la

Clase	TT	a	b
C_0	**0-**	C_2	C_0
	4	C_0	C_0
C_2	**2**	C_2	C_1
C_1	**3+**	C_1	C_2
	5+	C_1	C_2

Debo volver a separar

Separo el estado 4 y ya no se puede separar más

Clase	TT	a	b
C_0	**0-**	C_2	C_3
C_2	**2**	C_2	C_1
C_3	**4**	C_3	C_0
C_1	**3+**	C_1	C_2
	5+	C_1	C_2

Ejemplo

Tabla de Transición Inicial

Clase	TT	a	b
C_0	**0-**	2	4
	2	2	3
	4	4	0
C_1	**3+**	5	2
	~~**5+**~~	~~3~~	~~2~~

Tabla de Transición por Clases

Clase	TT	a	b
C_0	**0-**	C_2	C_3
C_2	**2**	C_2	C_1
C_3	**4**	C_3	C_0
C_1	**3+**	C_1	C_2
	5+	C_1	C_2

Opción 1: elimino el estado 5 y lo reemplazo por el 3 en la tabla

TT	a	b
0-	2	4
2	2	3
4	4	0
3+	3	2

Tablas iguales solo que
4 <---------->3
3 <---------->1

Opción 2: uso los nros de clase como nros de estado.

TT	a	b
0-	2	3
2	2	1
3	3	0
1+	1	2

Licencia