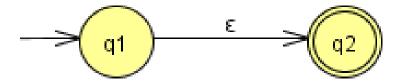
Construcción de Thompson de un AFN a partir de una expresión regular

Construcción de Thompson

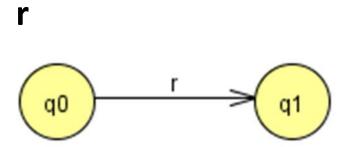
La construcción de Thompson construye a partir de una expresión regular r un AFND que reconoce el lenguaje definido por r, esto se realiza con el objetivo de que en un algoritmo siguiente se pueda generar un AFD mínimo equivalente.

Utiliza una notación estándar para generar el AFN

 \square Para la representación de una cadena vacía se utiliza el símbolo ε .

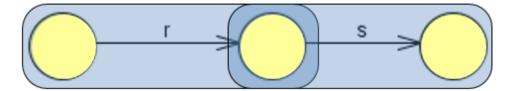


□ Para representar un símbolo, se utilizan dos estados y una transición para el movimiento con el símbolo.

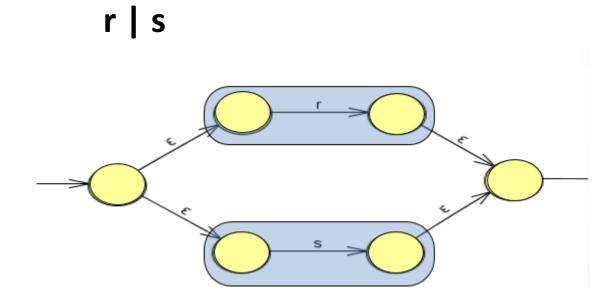


☐ Para la **concatenación** de dos símbolos únicamente se unen

rs



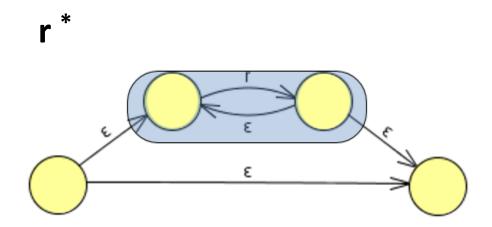
 \square Para la elección de **alternativas**, creamos transiciones ε para la unión de las transiciones.



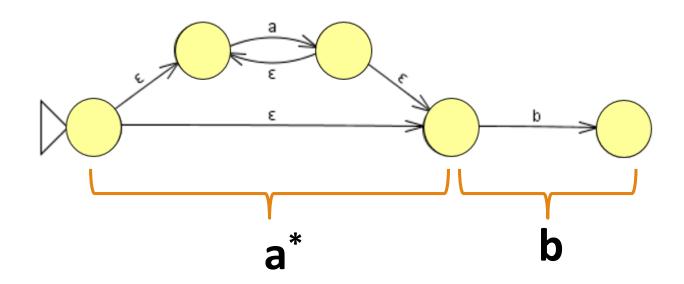
 \square Para la **cerradura positiva**, se agregan transiciones ε para retornar al estado previo, permitiendo agregar 1 o mas veces el símbolo.



 \square Para la **cerradura de Kleene**, se agregan transiciones ε para retornar a estado previo. Y otra transición ε para saltar la transición con r.

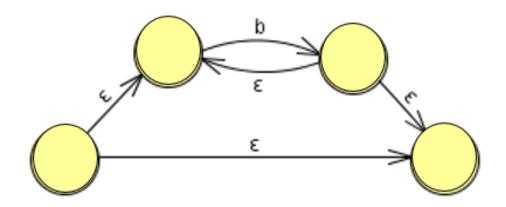


Ejemplo 1: Diagrama del AFN que representa la ER **a*b**

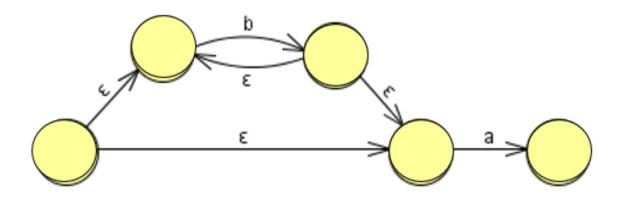


Ejemplo 2: Diagrama del AFN que representa la ER (b|(b*a)*)a

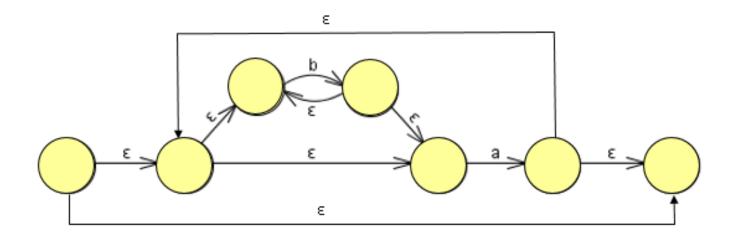
a) Primero hacemos la cerradura de Kleene para b*: (b | (b*a)*)a



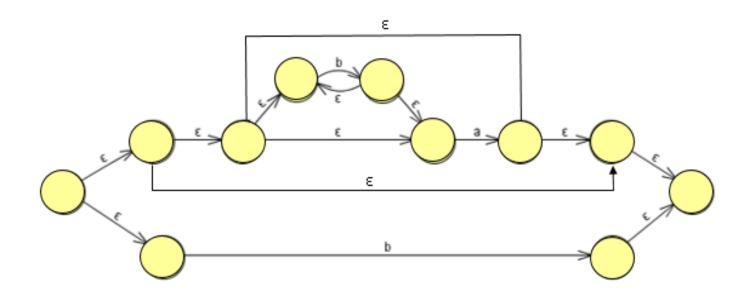
B) Ahora completamos el paréntesis interior (b^*a) : $(b|(b^*a)^*)a$



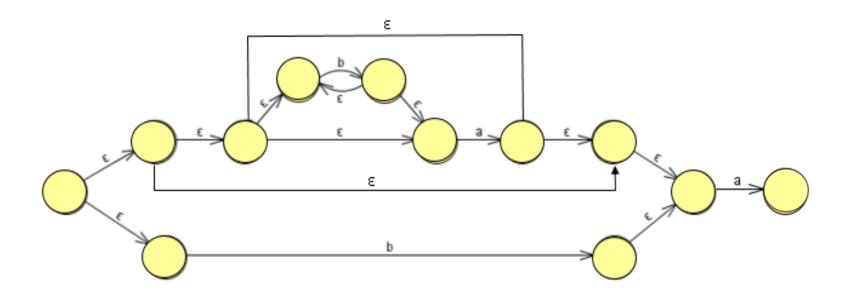
c) Aplicamos la cerradura de Kleene al paréntesis (b*a)*: (b | (b*a)*)a



c) Agregamos la alternativa b, **b|(b*a)***: **(b|(b*a)*)a**



d) Concatenamos a: (b | (b*a)*)a



Contruye los AFND

- 1. (abc)*
- 2. (b|bc)+
- 3. letra_(letra_|digito)*
- 4. (a|b)*abb
- 5. [(b|b*a)*]a
- 6. $(a^*|b^+)^+$