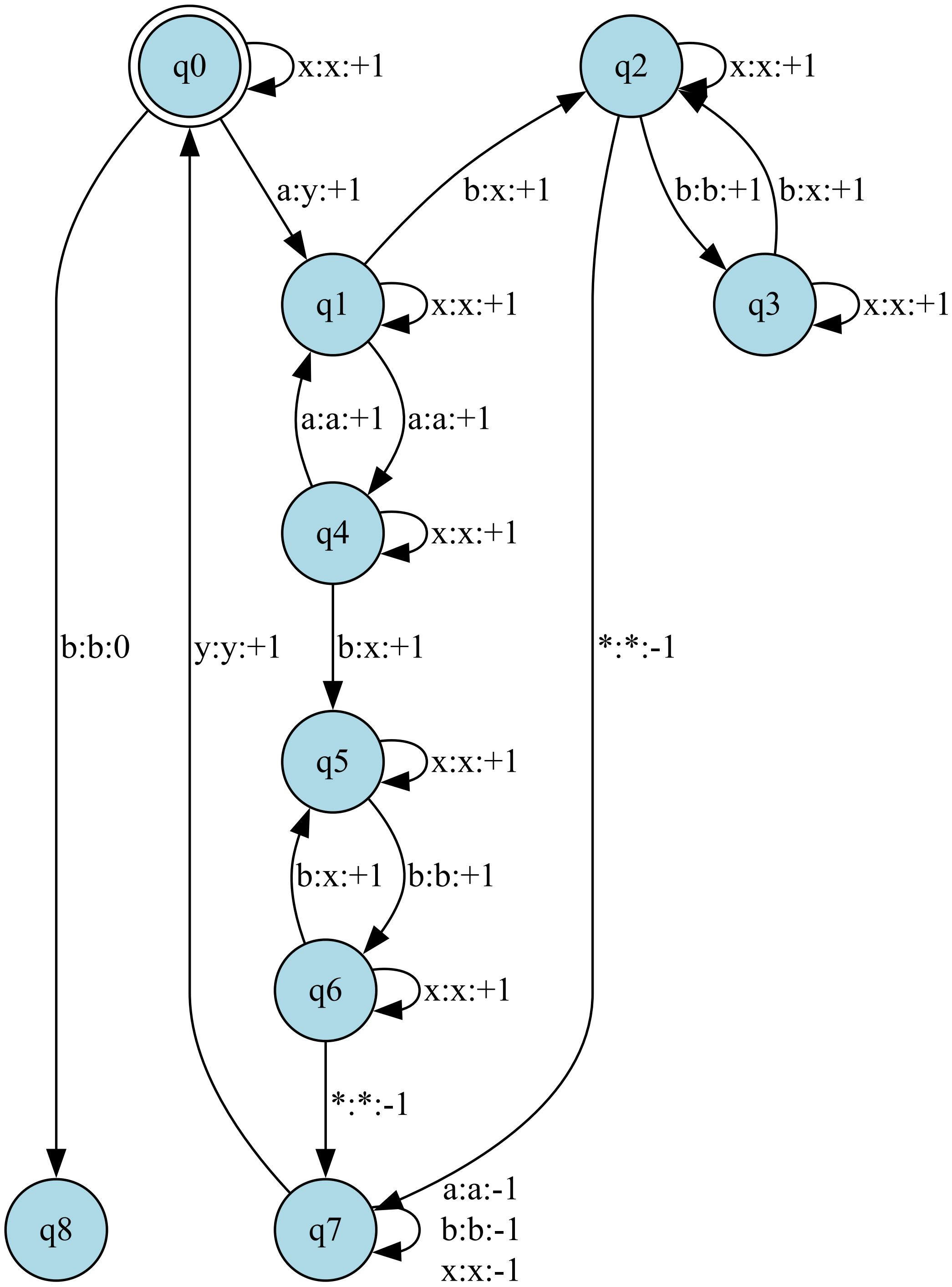
Turing Machine

Ejercicio 2.b

> ** Dada la siguiente maquina de turing determine que lenguaje reconoce:



El lenguaje que reconoce es $a^n b^{S(n)}$ donde se cumple que:

- $S(n) = 2S(n-1)$ si n es par
- $S(n) = 2S(n-1) + 1$ si n es impar

- $S(1) = 1(\text{la cadena } ab \in L)$

Nótese que entre cadenas $\subset \{b\}*$ no pueden existir cadenas con a porque en los estados donde han sido leído previamente el caracter b no existen transiciones con a, y como las cadenas reconocidas deben empezar por a porque es la transición que no tranca el autómata desde $q_0 \implies$ estas cadenas tienen la forma $a^n b^k$.

Nótese también que las transiciones x son equivalentes a obviar dicho caracter en la cinta, ya que al reconocerlo la MT se mantiene en el estado y avanza, lo cual, es equivalente a que no lo hubiese leído y saltara esa posición, por lo que, al marcar la primera a como y y una cantidad m de b, la cinta a analizar por la MT es equivalente a analizar la cadena $a^{n-1}b^{k-m}$

Nótese además que la cantidad de a y de b tienen la misma paridad:

- Supongamos que la cantidad de a es par $\implies$ al terminar de leerlas termina la MT en q_4, luego, al leer las b, para que la cadena sea reconocida, esta subcadena de b debe terminar en q_6, lo cual solo es posible si la longitud de b es par.
- Supongamos que la cantida de a es impar $\implies$ dicha subcadena termina en q_1, y al comenzar a leer las b pasaría la MT a los estados q_2, q_3, de los cuales hay salida $\iff$ la cantidad de b es impar.

Procedamos por inducción, tomemos como casos base las cadenas ab y $aabb$, supongamos que para $n = k - 1$, donde n es la cantidad de a en la cadena se cumple y analicemos que ocurre con k. Diferenciemos dos casos:

- k es par: entonces si inicialmente hay una cantidad par de a la MT termina de leerlas en q_4, luego, al leer los caracteres b y marcarlos alternadamente con x lo que ocurre es que, si k es par y t es el número de b que había $\implies$ el número de b resultante es $t - \frac{t}{2} = r \implies t = 2r$, y al leer $*$ la MT volverá al estado donde reconoce una a, por lo que caeríamos en el caso anterior ($a^{k-1}b^r$). Tomando $r = S(n-1)$ la cantidad de b que hay en el momento $n - 1$ y $t = S(n)$ como la cantidad que hay en el momento n (nótese que los "momentos" son contados como la cantidad de a que existen en la cadena, que disminuyen de 1 en 1), nos queda la primera expresión de la función S.
- k es impar: entonces al terminar de leer los caracteres a la MT termina en q_1, lo cual, indica que, si t es la cantidad de b, en la próxima vuelta de la MT dicha cantidad será $t - \frac{t+1}{2} = r \implies t = 2r + 1$, por lo que, resulta en $a^{k-1}b^r$ que es

nuevamente un caso anterior y la expresión para este caso de la recurrencia es la segunda

Ejercicio 3

> * Construya una maquina de turing que reconozca 1^n con n primo

111111111111 ****111&xxxx11111111