

Computabilidad y algoritmia

Práctica 5: Autómatas finitos en JFLAP

Alejandro Pérez Álvarez
alu0101215310@ull.edu.es

Diseño de DFAs

3

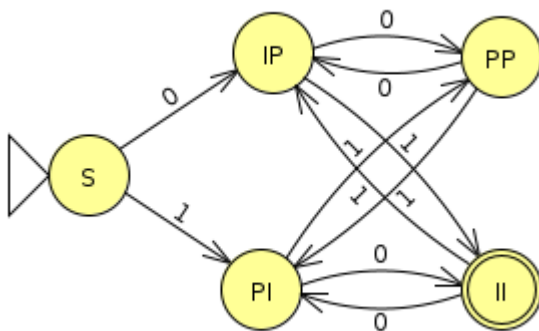
1. Diseñar un autómata finito determinista que reconozca cadenas binarias que contengan un número impar de unos y un número impar de ceros. 3
2. Diseñar un autómata finito determinista que reconozca cadenas binarias de longitud par. 5
3. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b, c\}$ que no contengan la subcadena abc. 8
4. Diseñar un autómata finito determinista que acepte números reales. El alfabeto que usa el autómata se define como $\Sigma = \{+, -, ., E, e, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ 11

Diseño de DFAs

1. Diseñar un autómata finito determinista que reconozca cadenas binarias que contengan un número impar de unos y un número impar de ceros.

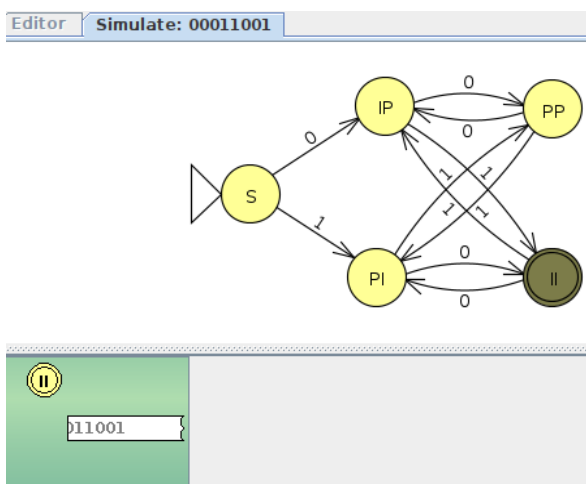
Para el diseño del autómata, suponiendo que el número cero sea un número par, se ha tenido en cuenta la existencia de cuatro estados diferente (más el de arranque): un primero donde el número de de ceros es impar pero de unos es par (IP); un segundo que representa el caso opuesto, ceros pares y unos impares (PI); un tercero que indica que ambos son pares (PP); y el estado de aceptación en el que ambos son impares (II).

Diseños del autómata en JFLAP:



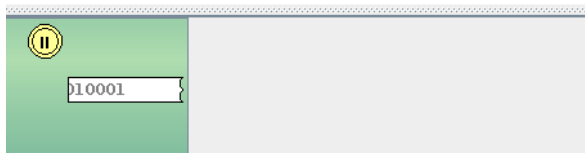
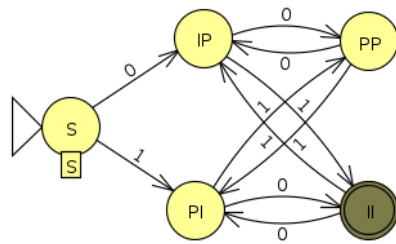
Ejemplos de aceptación del autómata:

- 00011001



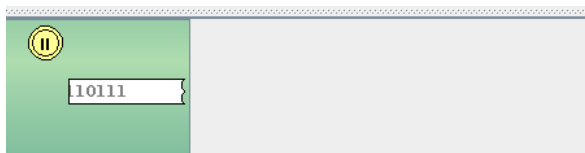
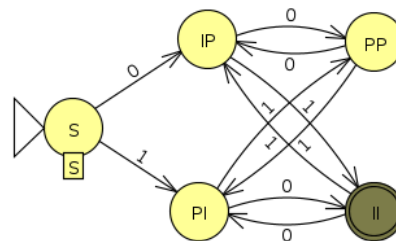
- 01010001

Editor Simulate: 01010001



- 110111

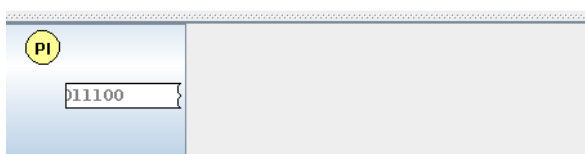
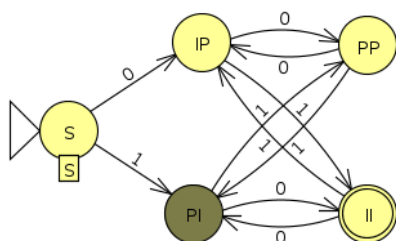
Editor Simulate: 110111



Ejemplos de rechazos del autómata:

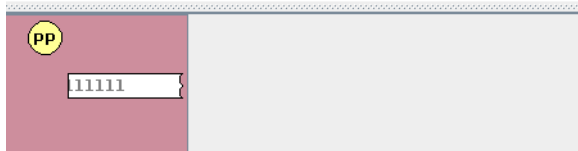
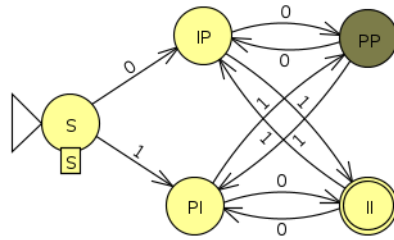
- 0011100

Editor Simulate: 0011100



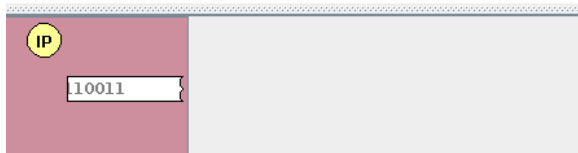
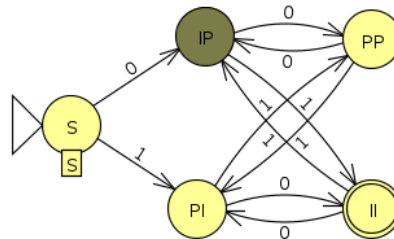
- 111111

Editor Simulate: 111111



- 01010110011

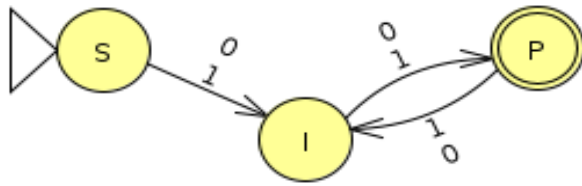
Editor Simulate: 01010110011



2. Diseñar un autómata finito determinista que reconozca cadenas binarias de longitud par.

Para el diseño del autómata se han contemplado dos estados posibles, uno en el que la cadena es impar (I), y el de aceptación en el que la cadena es par (P), independientemente del contenido de la misma.

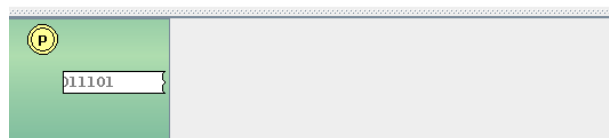
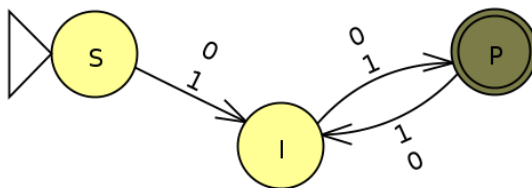
Diseño del autómata en JFLAP:



Ejemplos de aceptación del autómata:

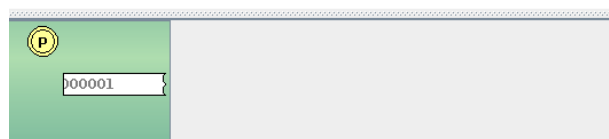
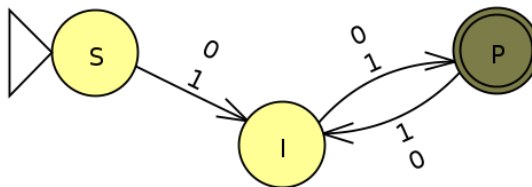
- 01011101

Editor Simulate: 01011101



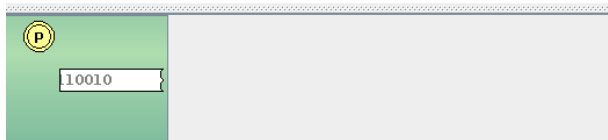
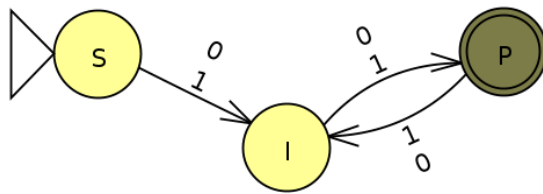
- 000001

Editor Simulate: 000001



- 111000110010

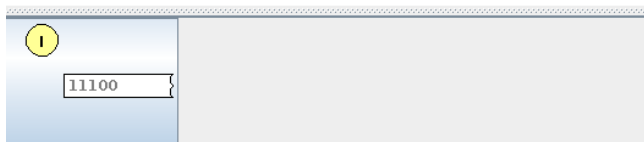
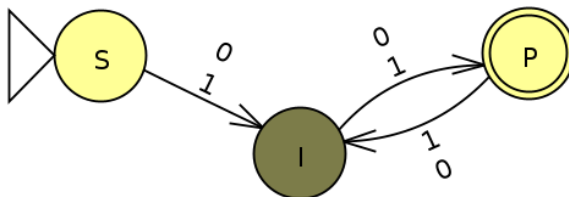
Editor Simulate: 111000110010



Ejemplos de rechazo del autómata:

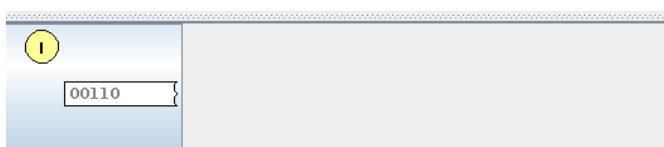
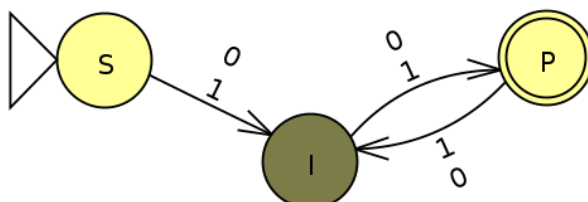
- 11100

Editor Simulate: 11100

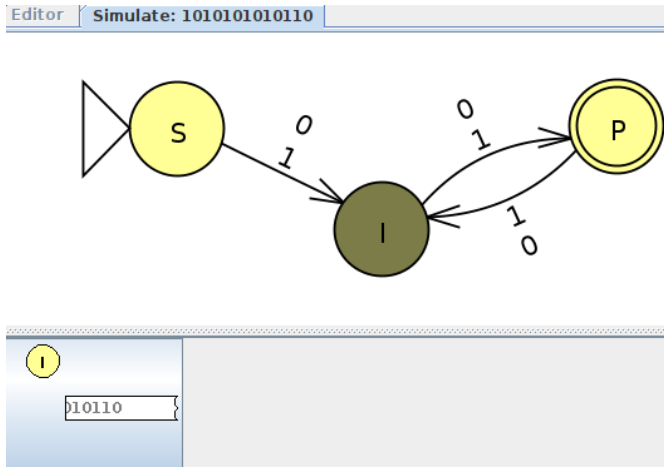


- 00110

Editor Simulate: 00110



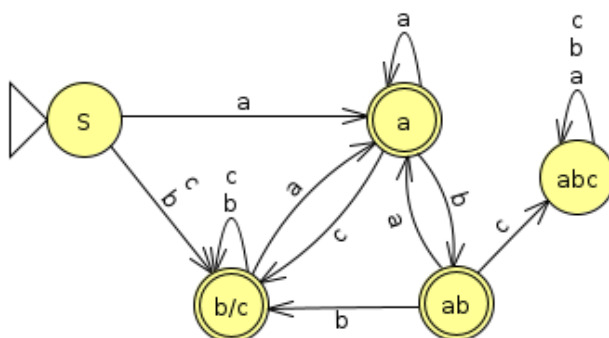
- 1010101010110



3. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b, c\}$ que no contengan la subcadena abc.

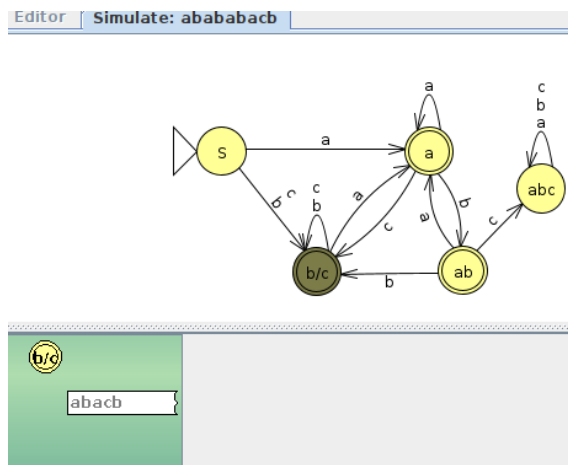
Para el diseño del autómata se ha tenido en cuenta tres estados de aceptación posibles: un primero donde ha entrado una a (a), y existe la posibilidad de incumplir la condición; un segundo donde, sin ir precedido de una a, ha llegado una b o una c, y por lo tanto no hay posibilidad de incumplimiento; y un tercero donde ha llegado una b justo después de una a y existe la opción de incumplimiento de validez. Además, existirá un estado de absorción al cual se llegará en caso de haber incumplido la condición, invalidando de cualquier manera la cadena.

Diseño del autómata en JFLAP:

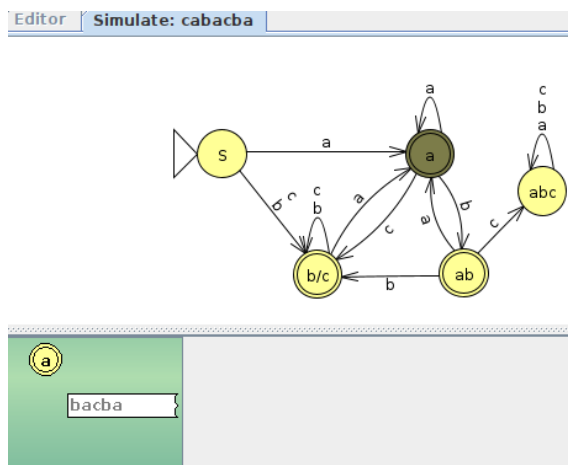


Ejemplos de aceptación del autómata:

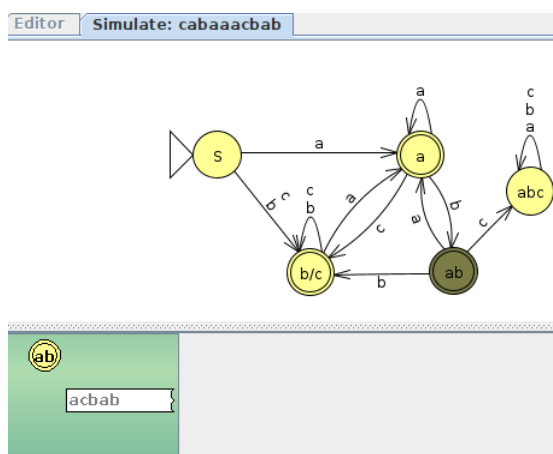
- abababacb



- cabacba

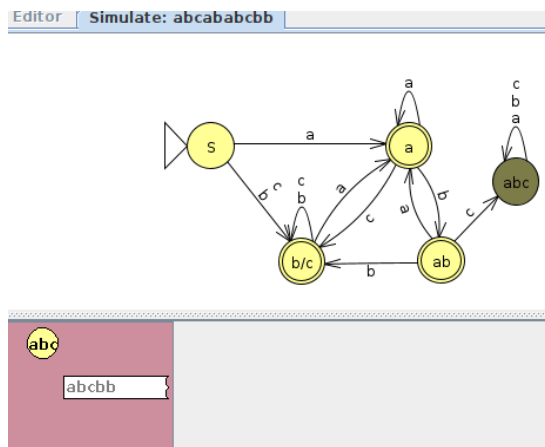


- cabaaacbab

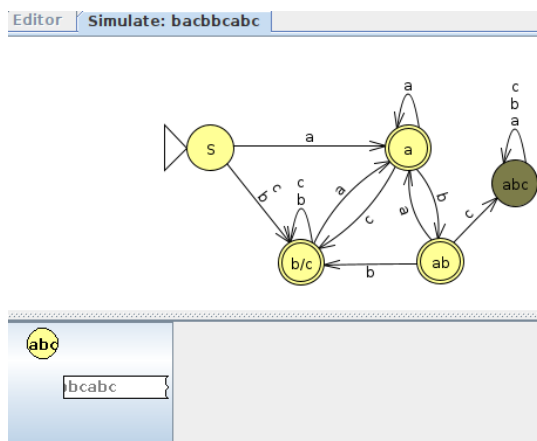


Ejemplos de rechazo del autómata:

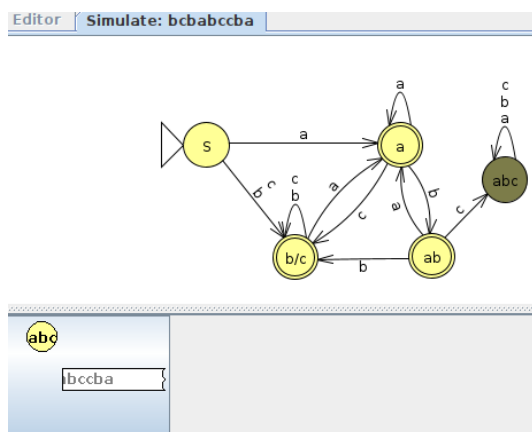
- abcababcbb



- bacbbcabc



- bcbabccba

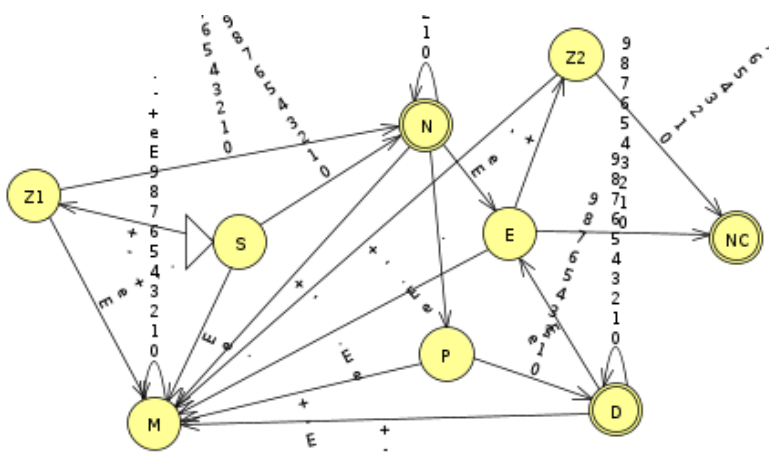


4. Diseñar un autómata finito determinista que acepte números reales. El alfabeto que usa el autómata se define como $\Sigma = \{+, -, ., E, e, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

- La cadena comienza opcionalmente por el símbolo “+” o “-”
- A continuación la cadena contiene uno o varios símbolos en el rango [0 – 9]
- Posteriormente, y de forma opcional, aparece en la cadena el símbolo “.”. Si aparece este símbolo, la cadena debe continuar con uno o más símbolos entre el rango [0 – 9].
- Opcionalmente la cadena puede ir seguida del símbolo “E” o “e” para indicar un número en notación científica. En este caso, tendrá que ir seguido de un símbolo “+” o “-” (opcional) y una cadena de uno o más símbolos entre el rango [0-9] que representa el exponente.

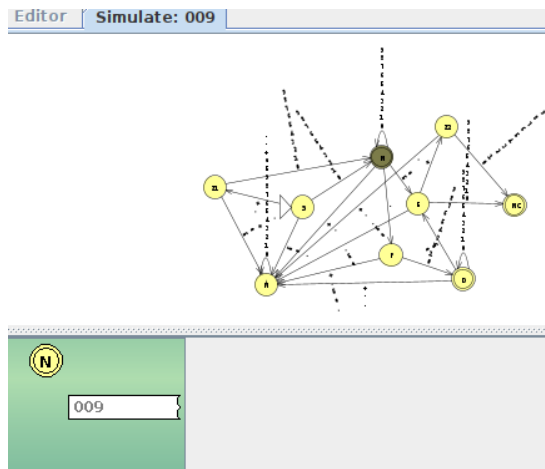
Para el diseño del autómata tendremos que tener en cuenta una serie de estados algo elevada respecto al resto de ejercicio, y llevar el análisis de las cadenas por partes, diferenciando en primer lugar los tres estados de aceptación que representarán a los número naturales (N), a los números decimales (D) y a los números expresados en notación científica (NC). A continuación añadiremos dos estados para los signos (Z1 y Z2) a los cuales se accede de forma opcional, un estado para el punto (P), también opcional salvo que se trate de un número decimal, un estado para la e o la E de la notación científica (E). Por último contaremos con el estado de arranque y un estado de absorción al que se llegará en caso de que la cadena no cumpla con el formato (M).

Diseño del autómata en JFLAP:

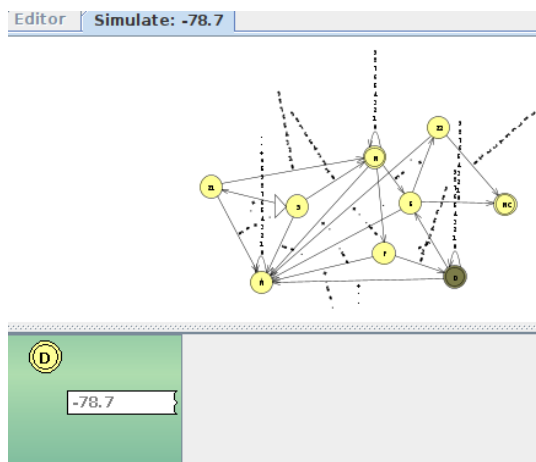


Ejemplos de aceptación del autómata:

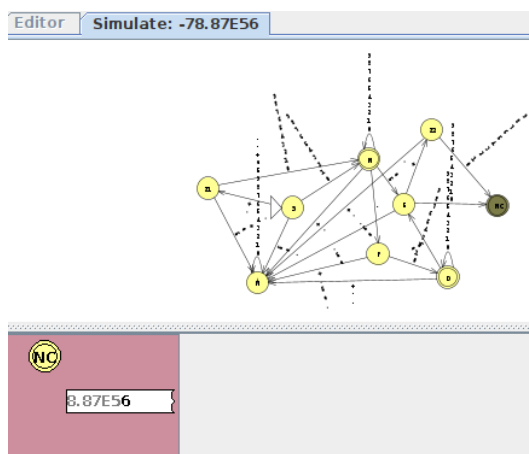
- 009



- -78.7

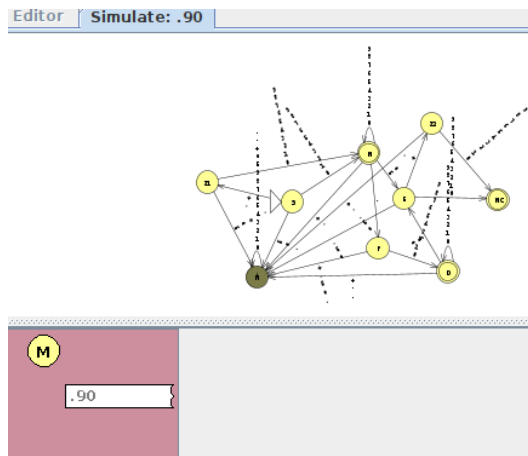


- -78.87E56

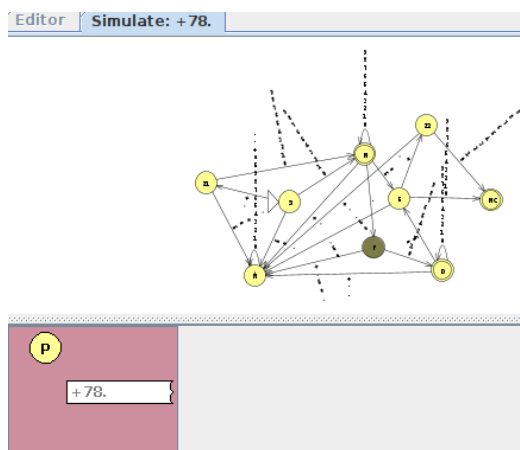


Ejemplos de rechazo del autómata:

- .90



- +78.



- +78.90E

