

# FUNDAMENTOS TEÓRICOS DA COMPUTAÇÃO

--- MÁQUINAS DE TURING ---

Exemplo (3)

$\omega\omega$

Consider the language  $M_3 = \{\omega\omega : \omega \in \{0,1\}^*\}$



Use the arsenal you have and be creative!

$\omega\omega$

$M_3 =$  Sobre a cadeia  $w$  de entrada:

- 1) Enquanto houver símbolos não marcados, vá e volte na fita marcando o **primeiro** e o **último** símbolos ainda não marcados. Se não for possível, rejeite. Se não houver símbolos, aceite ( $\varepsilon$ ).
- 2) A partir da **primeira** posição da **segunda** metade, desmarque os símbolos até chegar no primeiro "branco"
- 3) Para cada símbolo não marcado, copie os símbolos uma posição para a direita
- 4) Escreva o símbolo "#" na primeira posição da segunda metade e desmarque os símbolos até a extremidade esquerda
- 5) Execute  $M_1$  sobre a entrada. Aceite se  $M_1$  aceitar; e rejeite de  $M_1$  rejeitar.

# $\omega\omega$

U	0	1	0	1	0	1	0	U	U
U	0	1	0	1	0	1	0	i	U
U	0	i	0	1	0	1	0	i	U
U	0	i	i	0	1	0	i	0	i
U	0	i	i	0	i	0	i	0	i
U	0	i	i	0	i	0	i	0	i
U	0	i	i	0	1	0	1	0	1
U	0	i	i	0	0	1	0	1	0
U	0	i	i	1	#	0	1	0	1
U	0	1	0	1	#	0	1	0	1
U	x	1	0	1	#	x	1	0	1
...									
U	x	x	x	x	#	x	x	x	U

Inicial

Marque o 1º e último símbolos não marcados

Segunda metade, desmarque até o primeiro "U"

Copie os símbolos uma posição para a direita

Escreva o símbolo "#" na segunda metade

Desmarque os símbolos até a extrema esquerda

Execute  $M_1$  sobre a entrada modificada

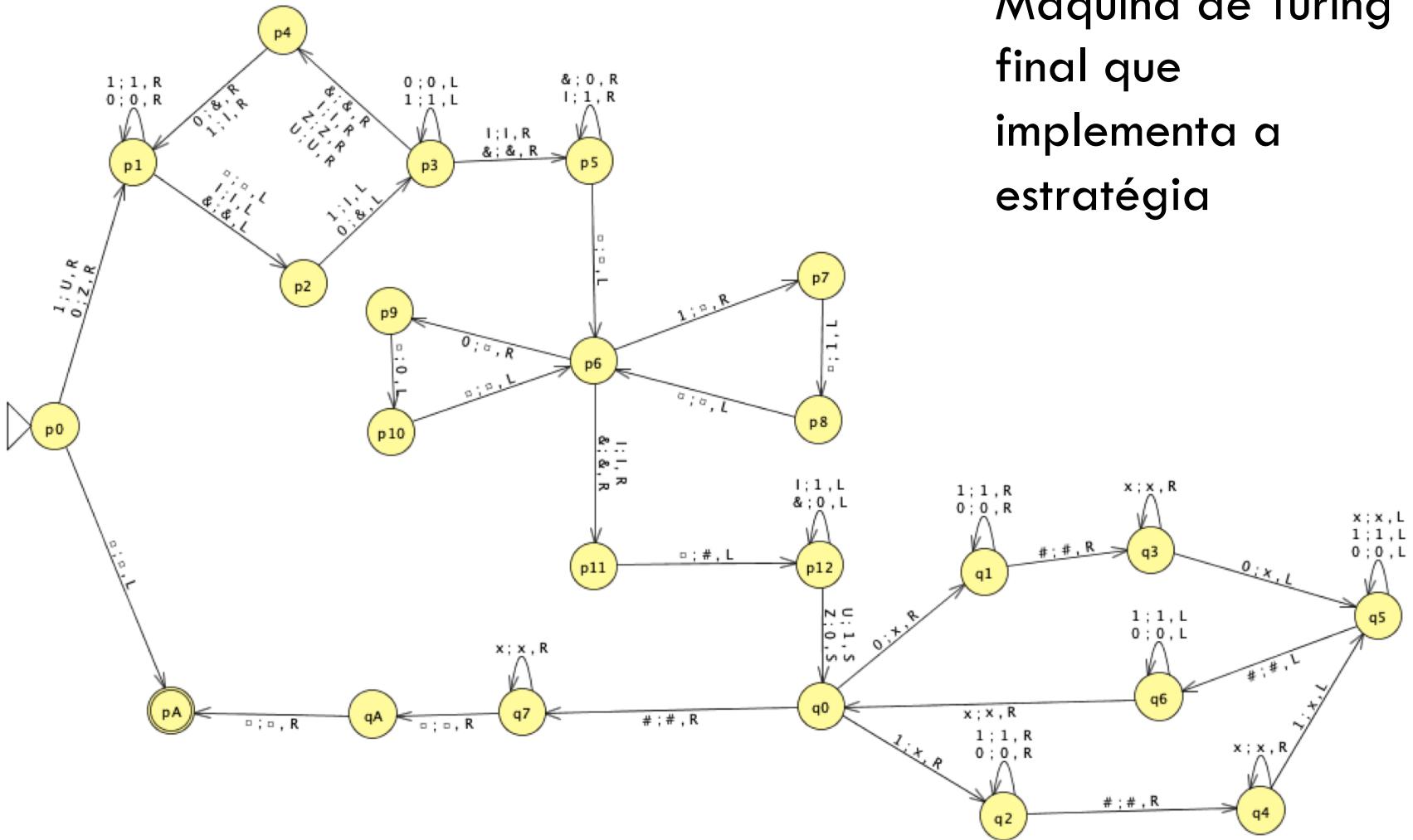
Execute  $M_1$  sobre a entrada modificada

$\omega\omega$

Na implementação foram feitas duas pequenas alterações

- 1) Ao invés de adotar os símbolos “ $\dot{0}$ ” e “ $\dot{1}$ ”, foram usados os símbolos “&” e “I”, respectivamente
- 2) Só para identificar qual é o primeiro símbolo, ao invés de usar os símbolos “&” e “I”, foram usados os símbolos “Z” e “U” para representar os símbolos “ $\dot{0}$ ” e “ $\dot{1}$ ”

$\omega\omega$

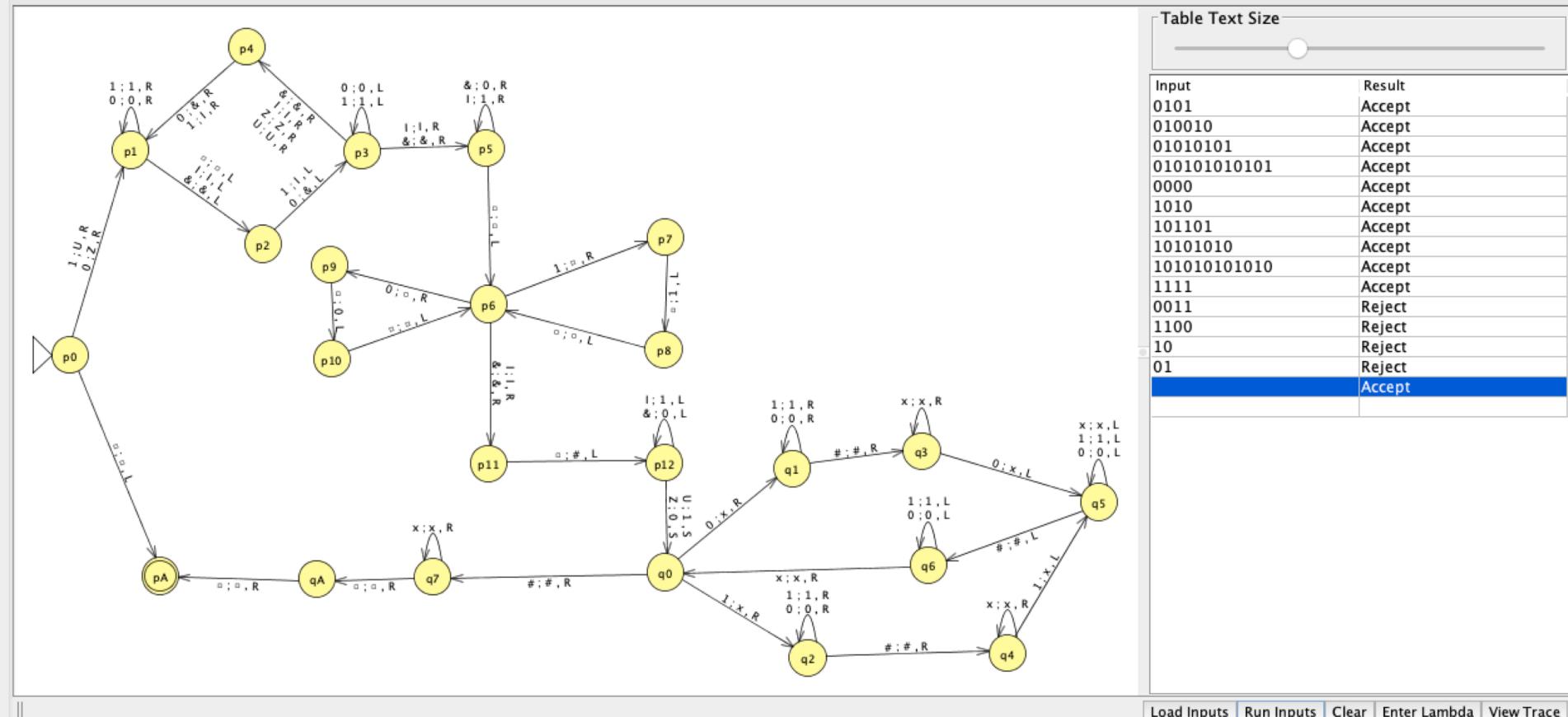


$\omega\omega$

## Multiple Run

File Input Test View Convert Help

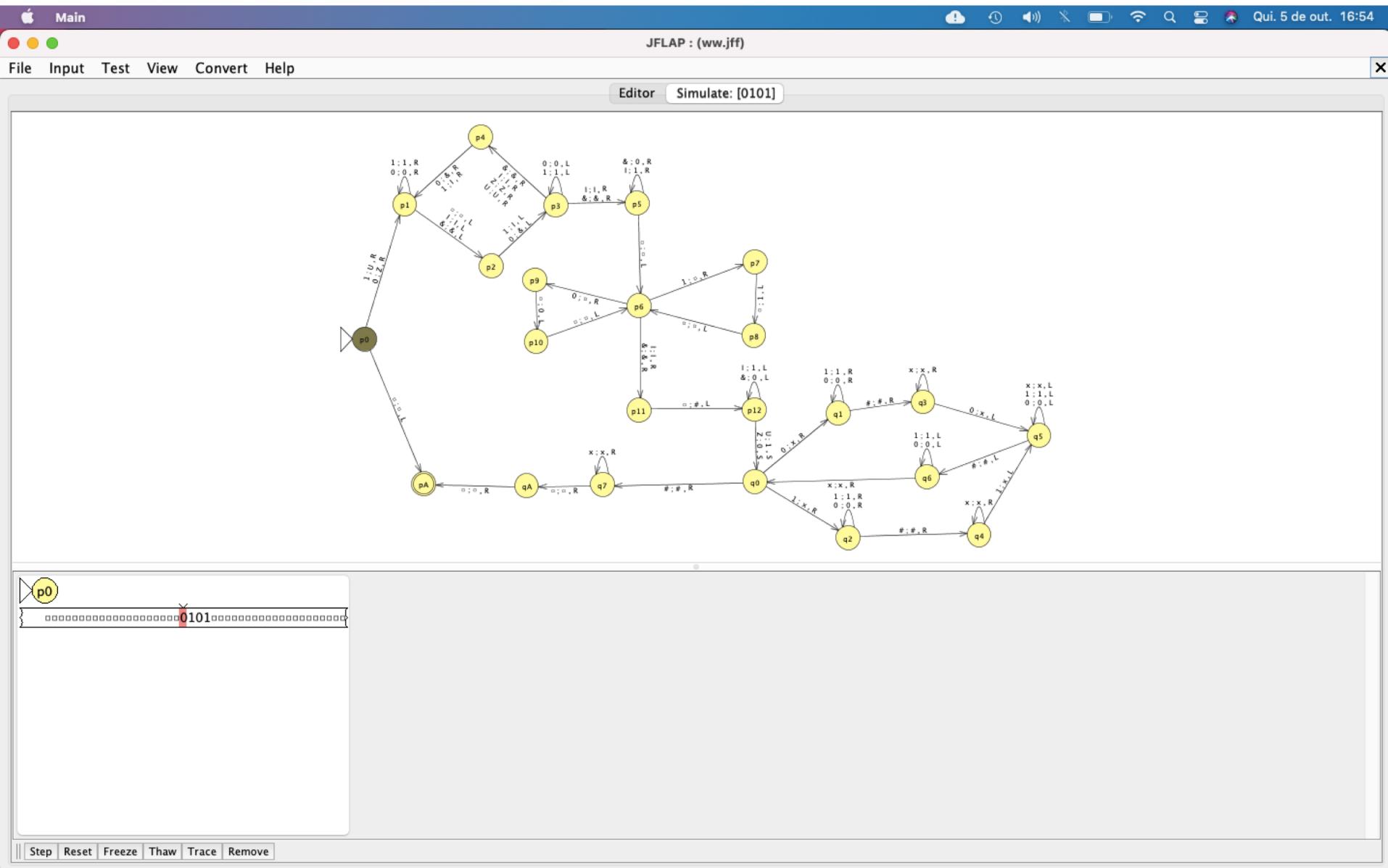
Editor Multiple Run



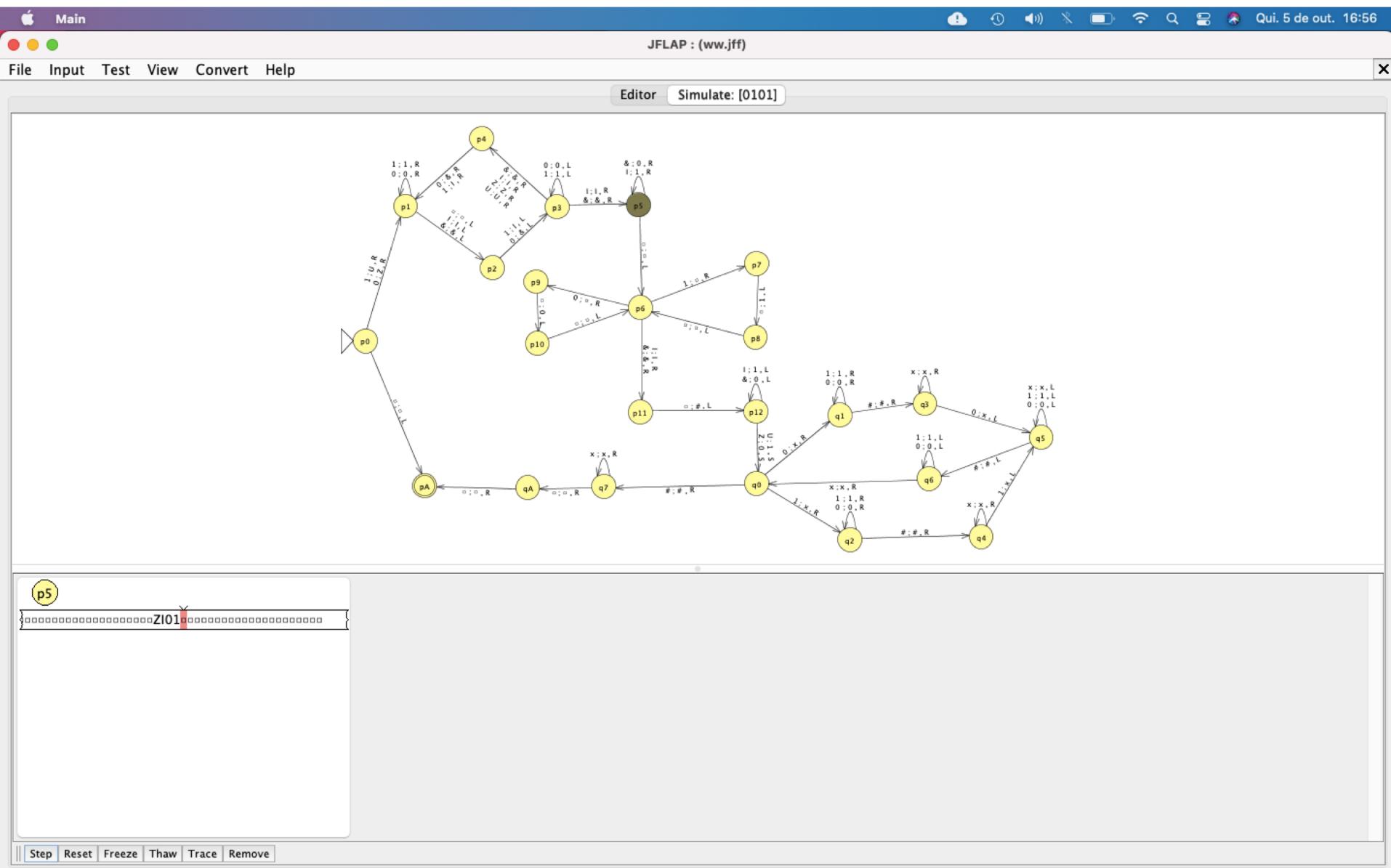


# Simulação de cadeia aceita

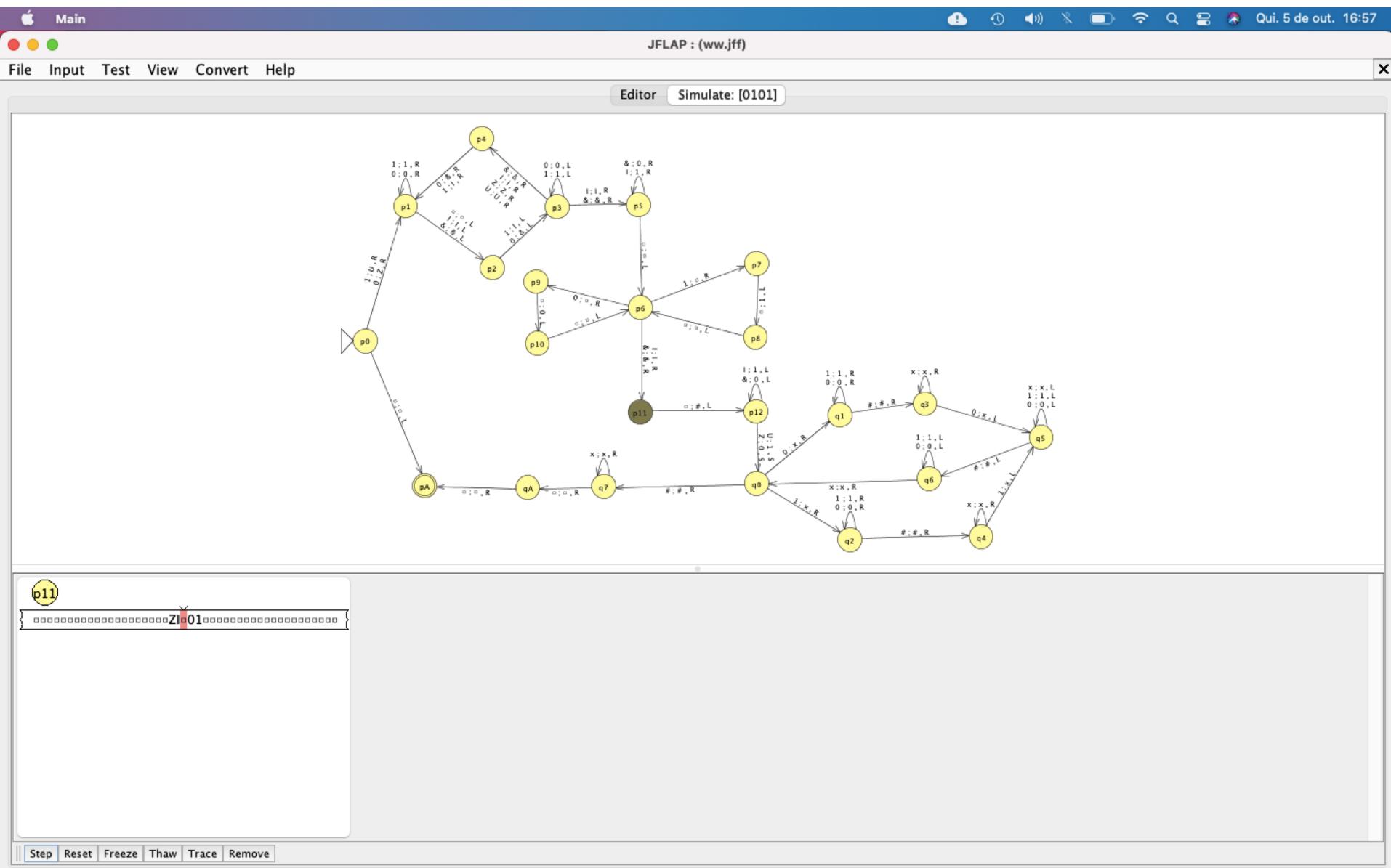
# $\omega\omega$



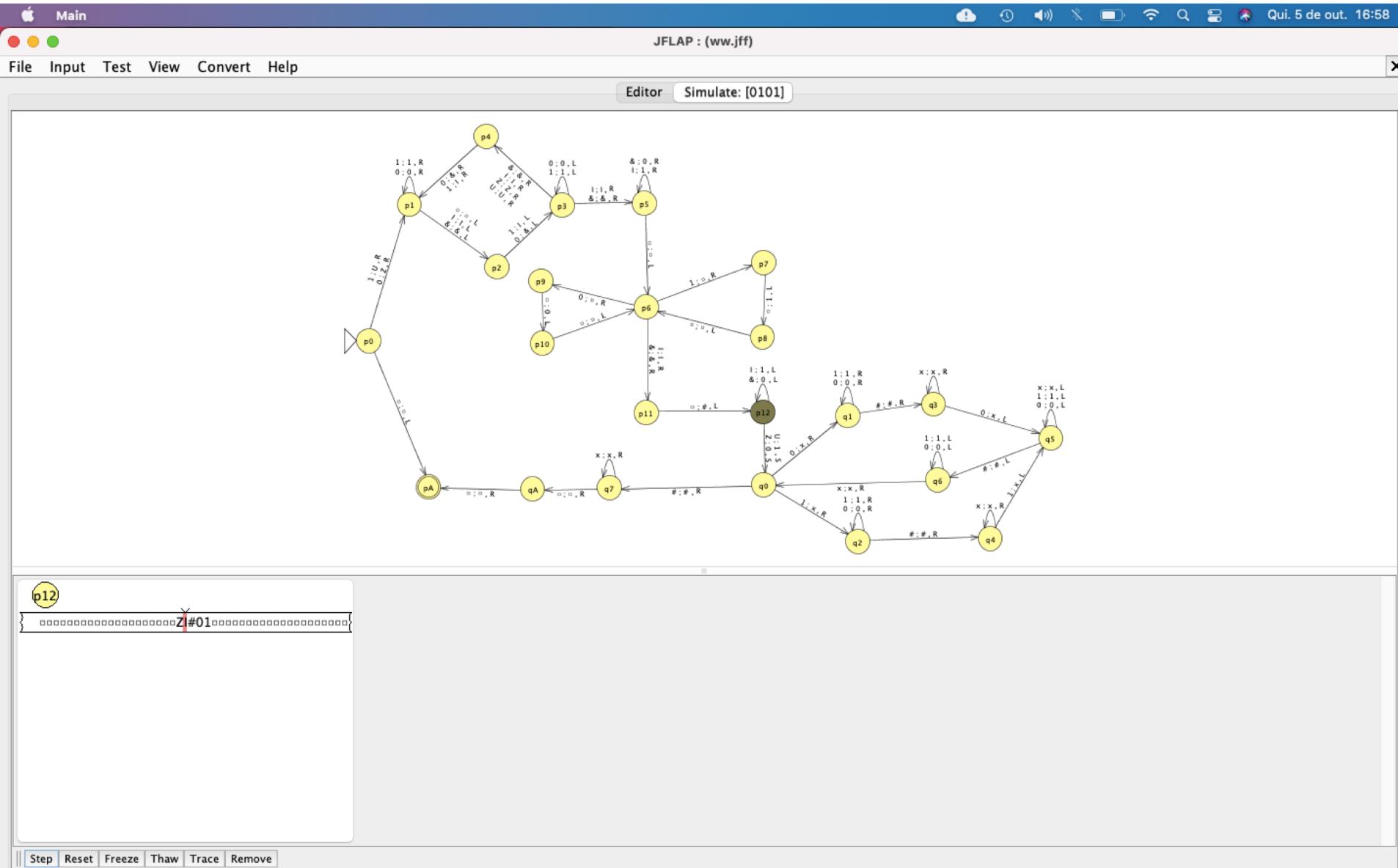
$\omega\omega$



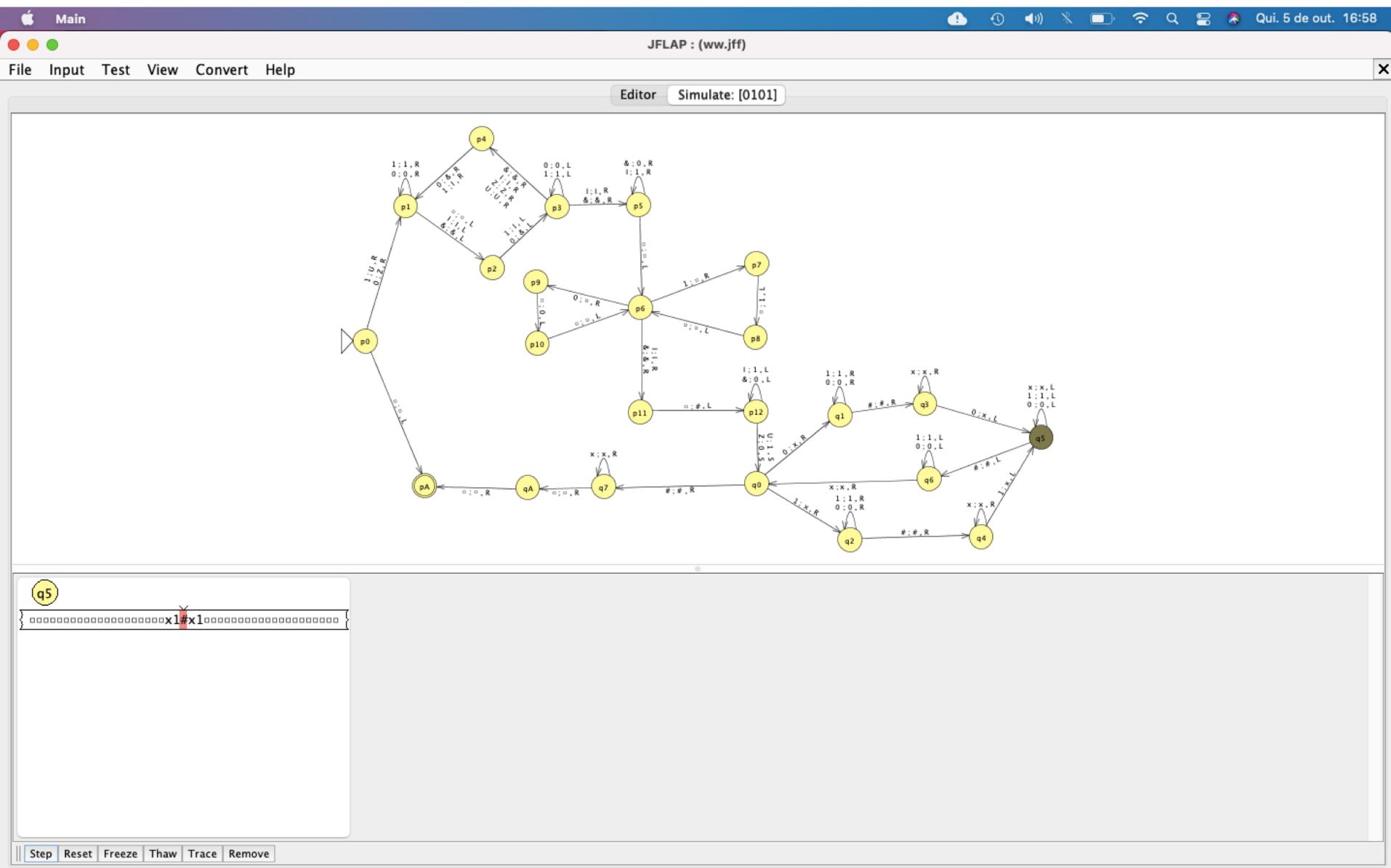
$\omega\omega$



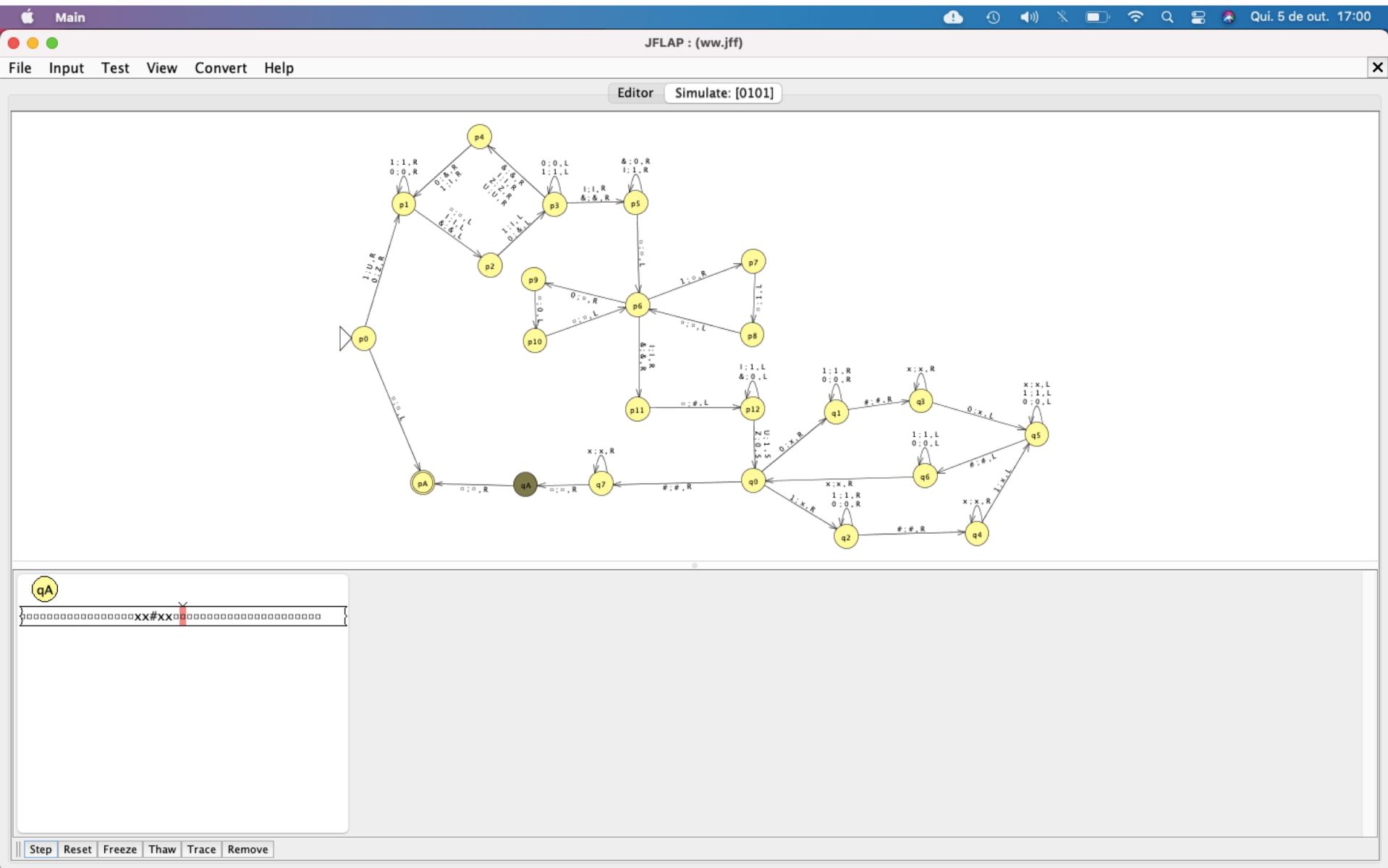
$\omega\omega$



$\omega\omega$

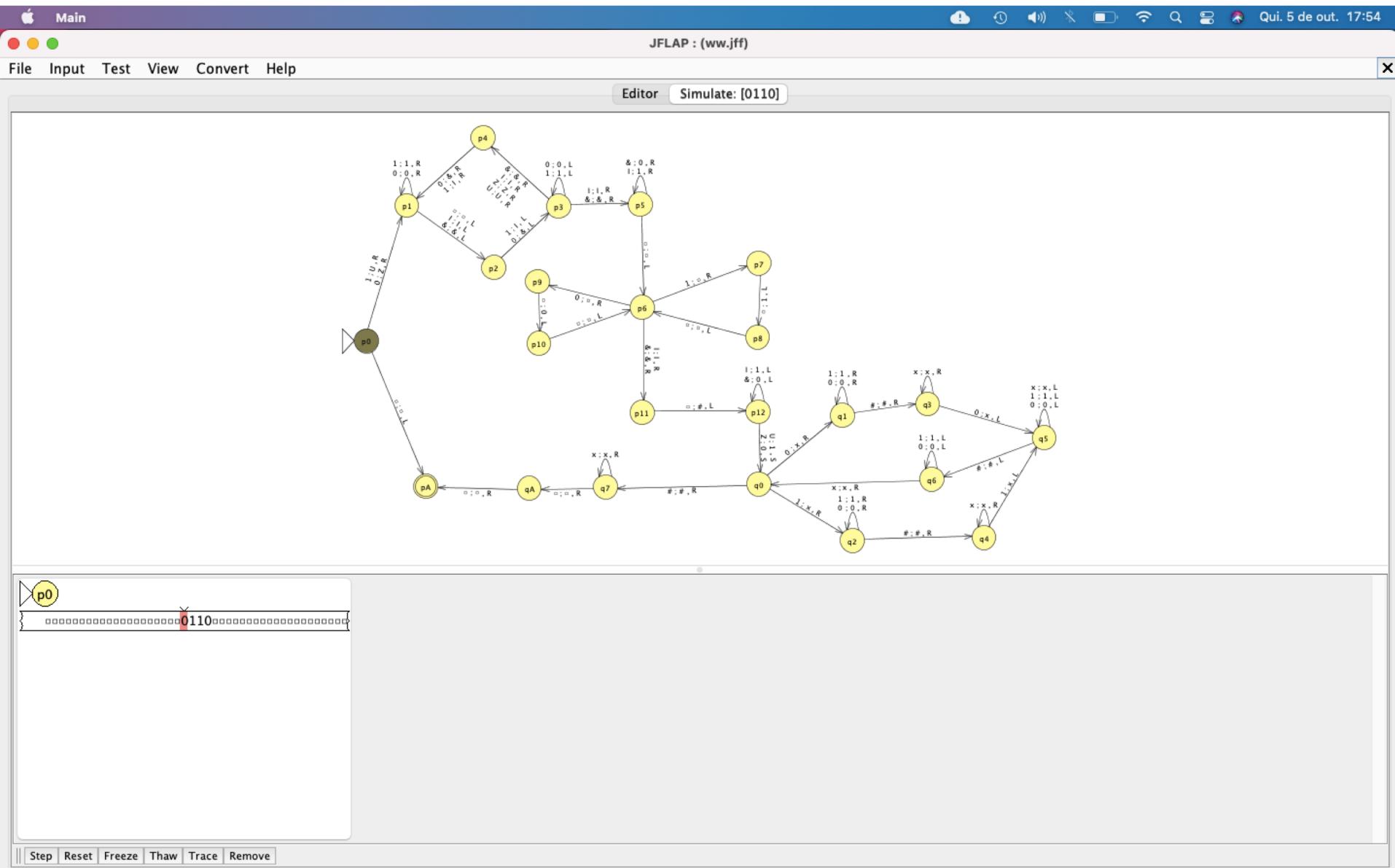


$\omega\omega$

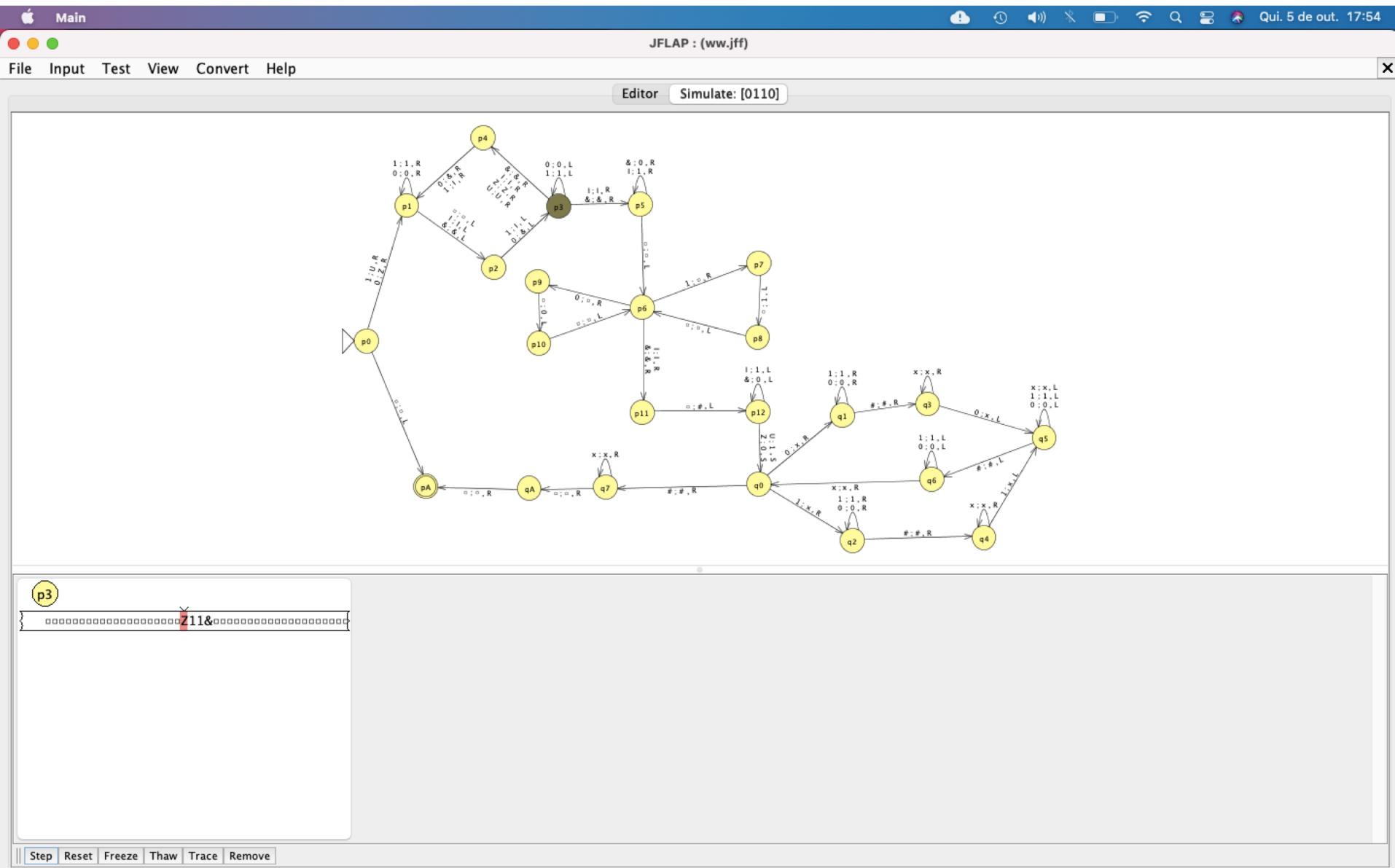


# Simulação de cadeia rejeitada

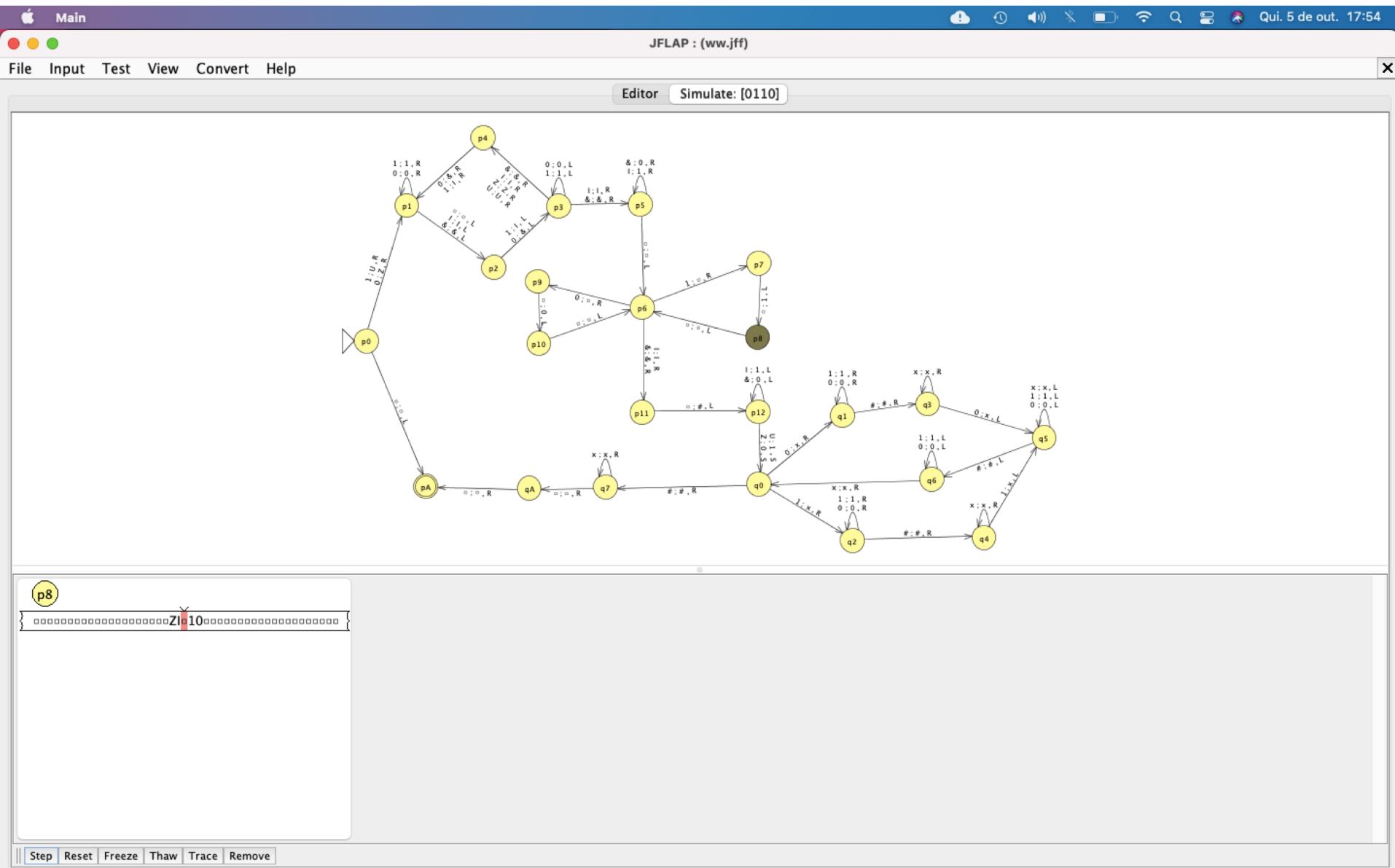
$\omega\omega$



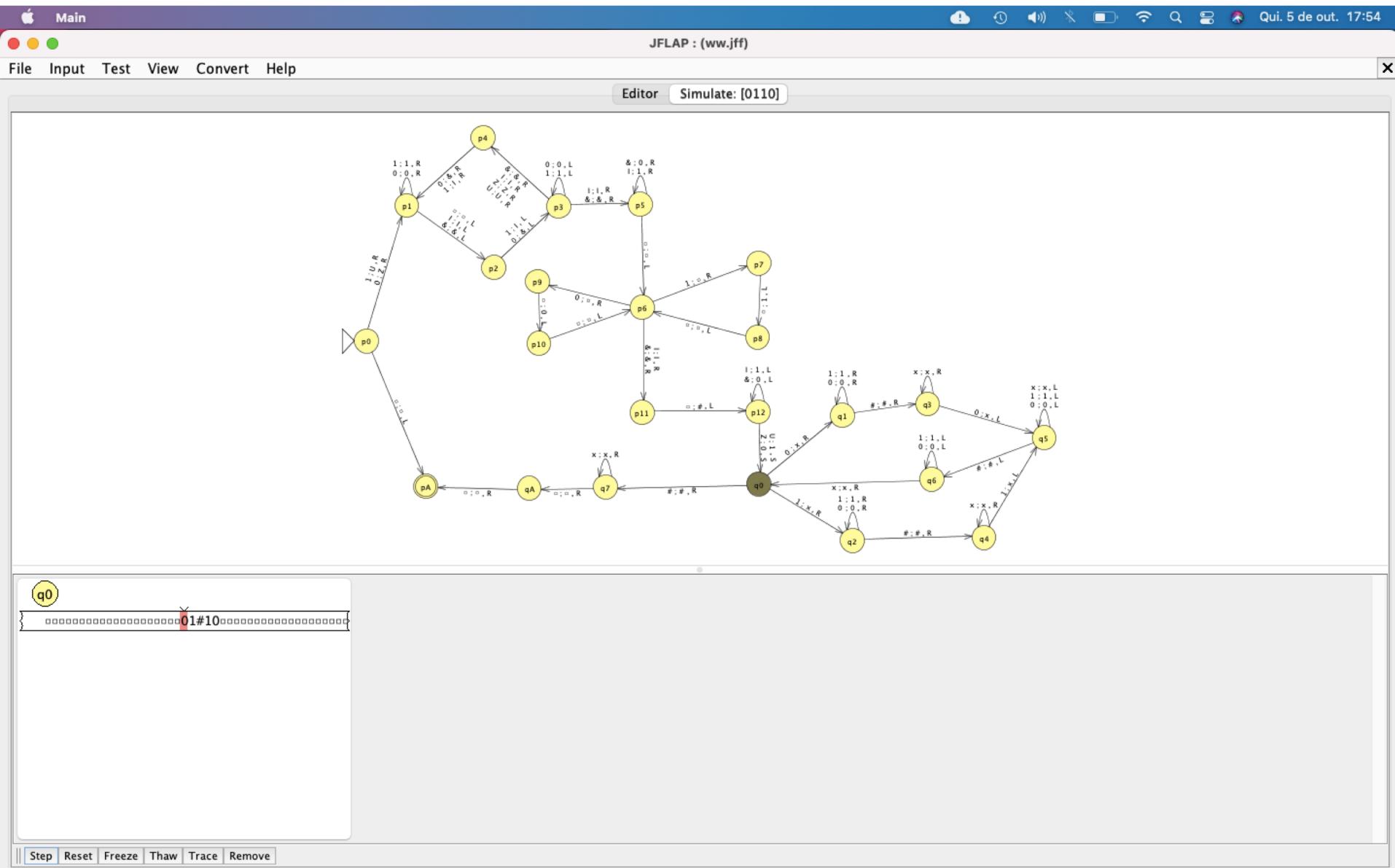
$\omega\omega$



$\omega\omega$



$\omega\omega$



$\omega\omega$

