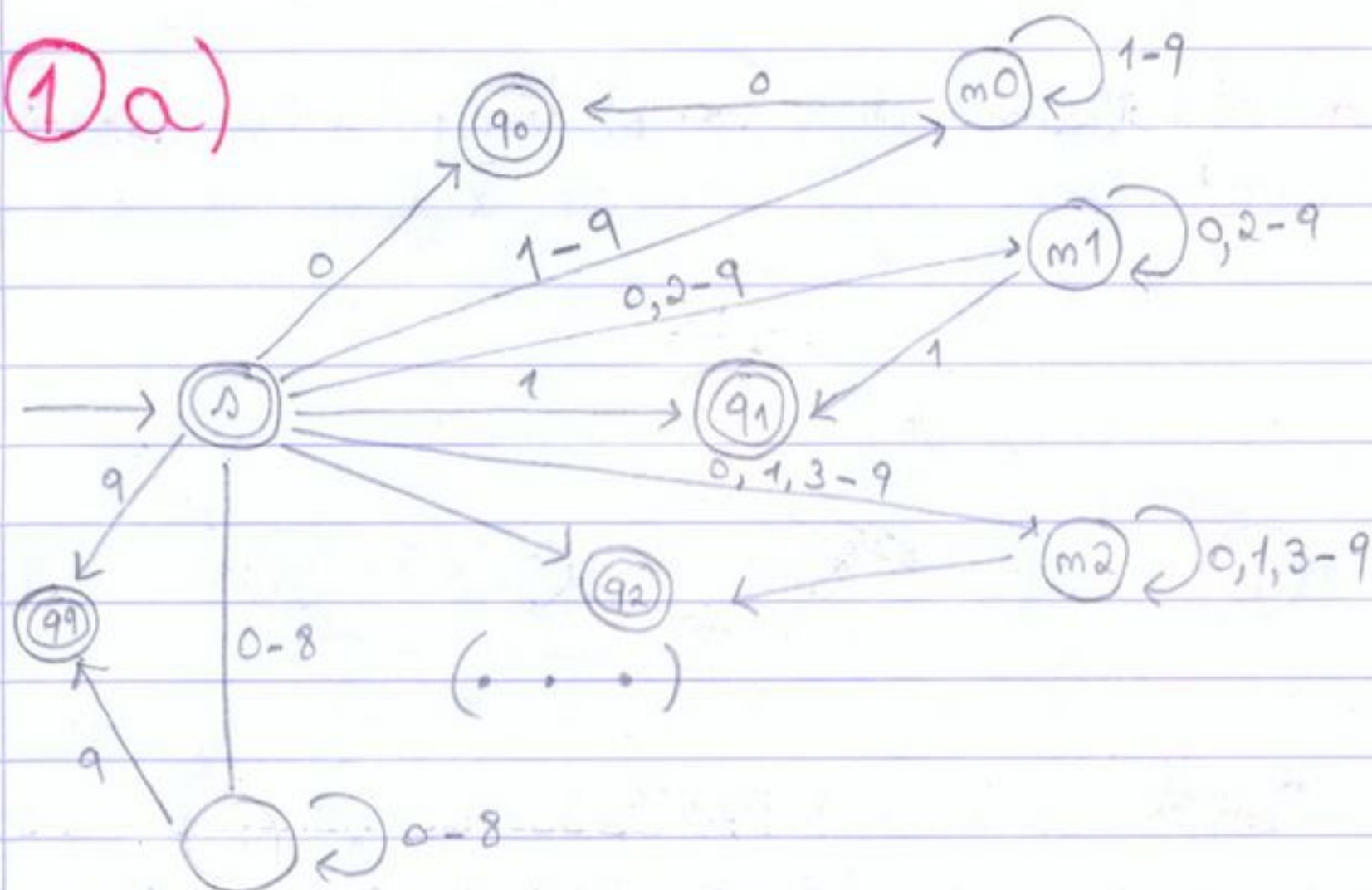
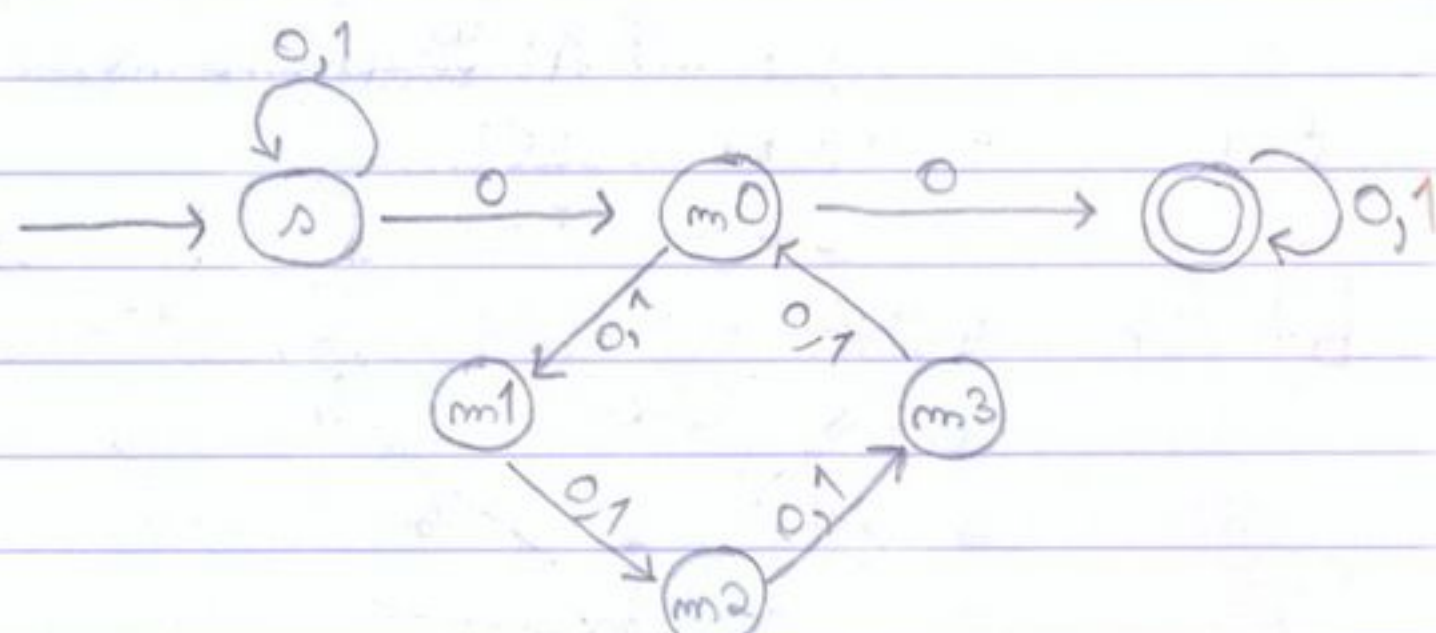


Prática 3 - NFAs (Non-Deterministic Finite Automata)

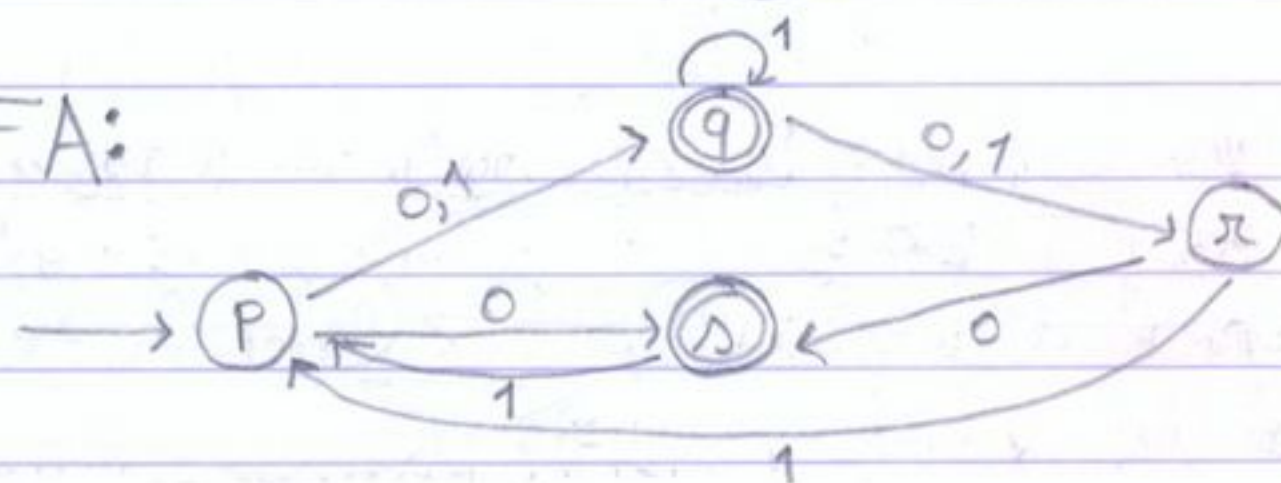
1a)



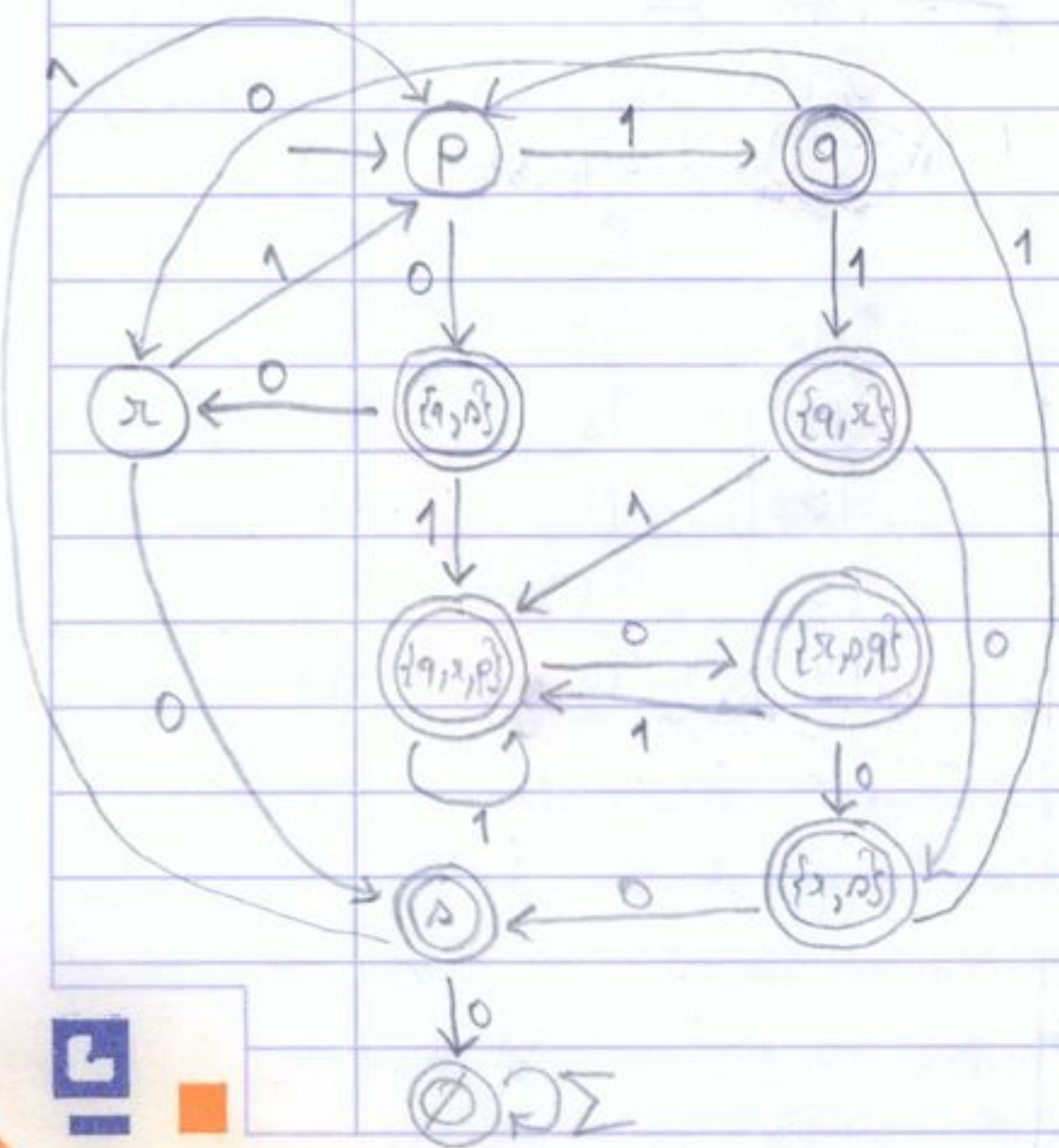
b)



2 NFA:

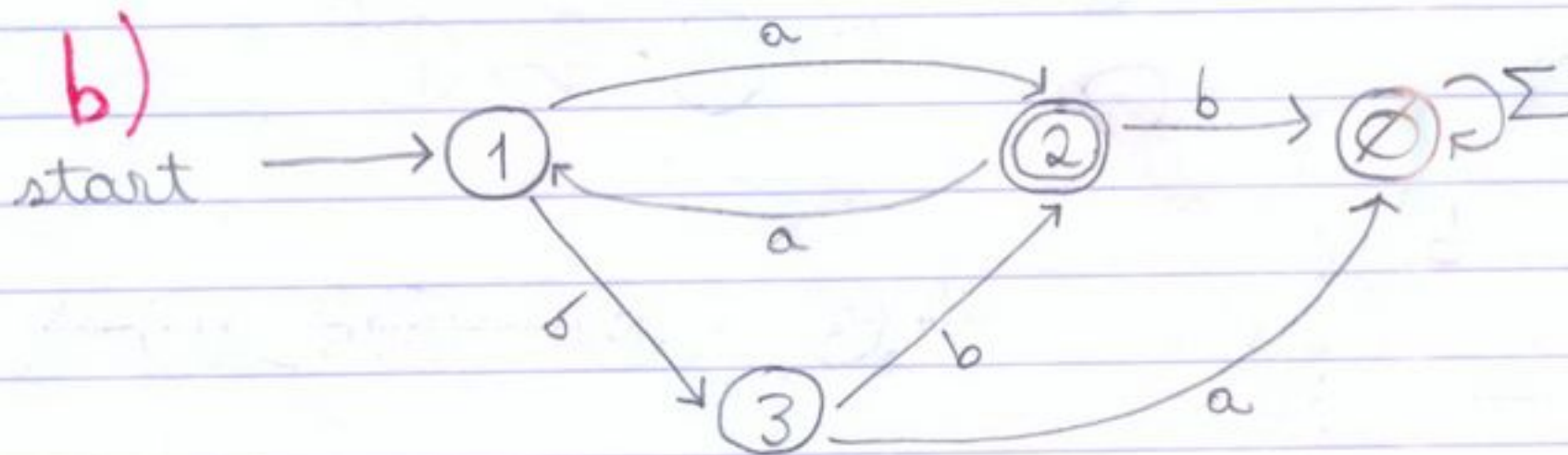


DFA:



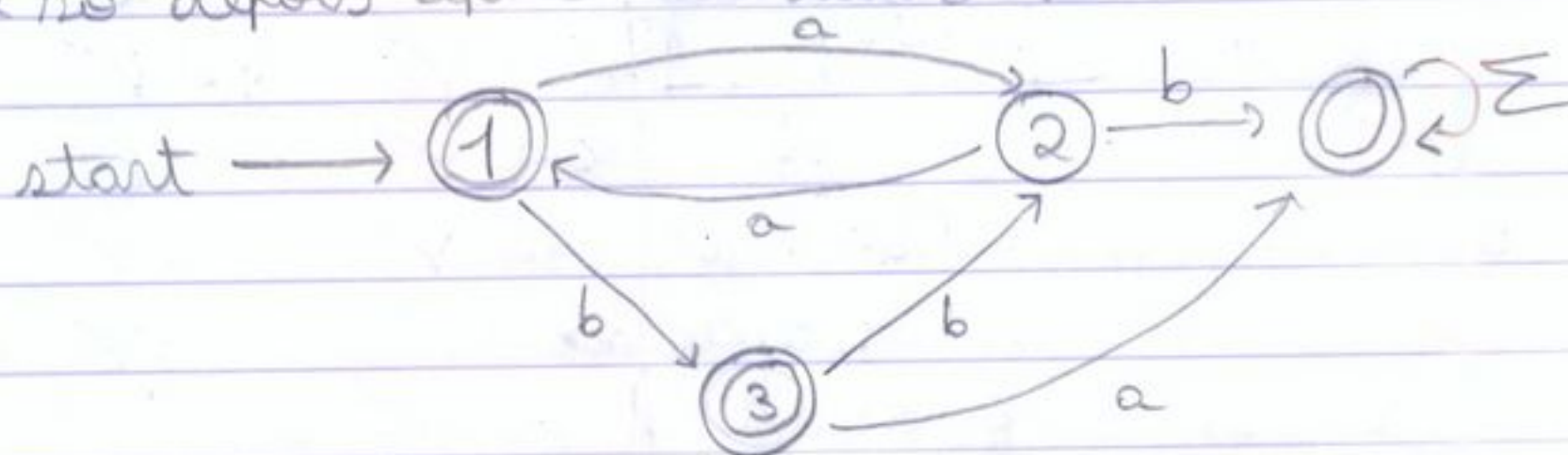
	0	1
→ p	{q, s}	q
* {q, s}	∅	{q, r, p}
* q	∅	{q, r}
r	s	p
* {q, r, p}	{r, s, q}	{q, r, p}
* {q, r}	{r, s}	{q, r, p}
* s	∅	p
* {r, s, q}	{r, s}	{q, r, p}
* {r, s}	s	p

③ a) Trata-se de um NFA, pois não possui transições para todos os pares (estado, símbolo de entrada).



e) Tática: Para DFA: Todos os estados de aceitação passam a ser de não-aceitação, e vice-versa.

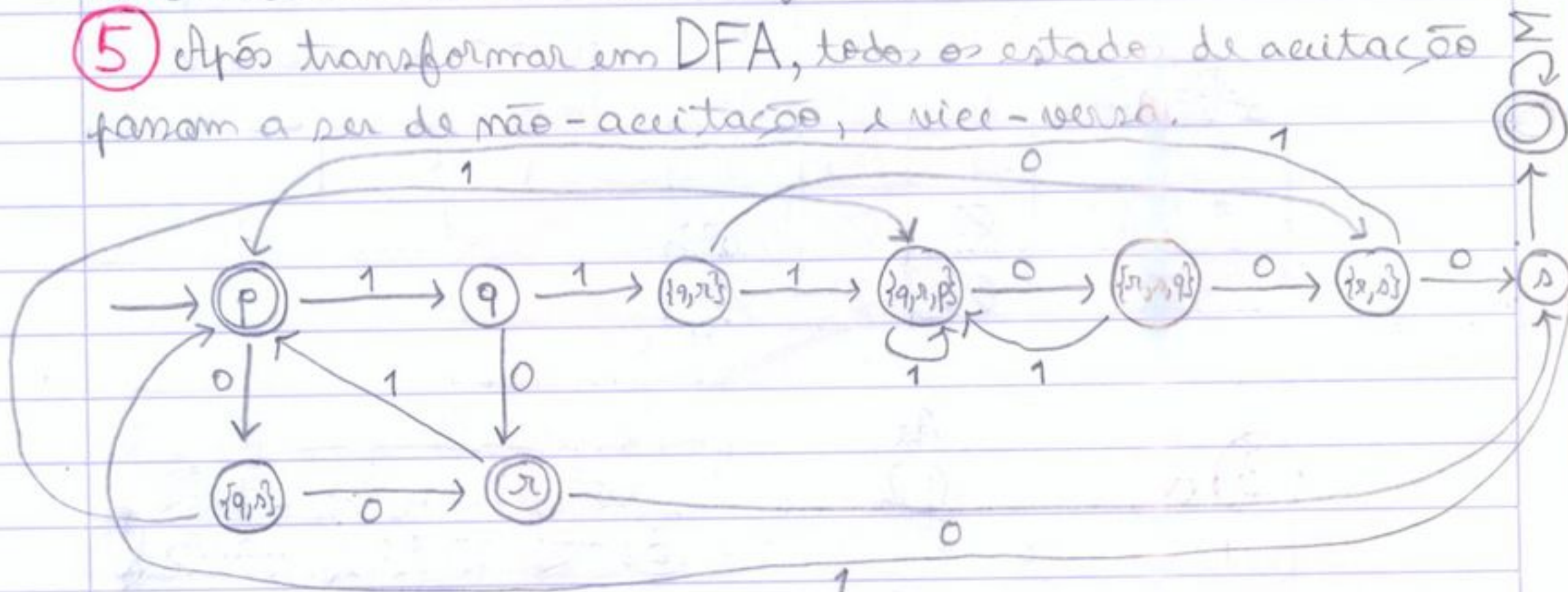
Para NFA: Transformar primeiro em DFA e só depois aplicar a tática.



④ a) Verdadeiro. Basta aplicar a técnica de construção de múltiplas cópias.

b) Falso. Um DFA pode conter "ciclos" e portanto reconhecer cadeias de caracteres cujo comprimento é superior ao n.º de estados.

⑤ Após transformar em DFA, todos os estados de aceitação passam a ser de não-aceitação, e vice-versa.



⑥ Partindo de um NFA N , em que para cada estado e símbolo de entrada há apenas um estado de destino no máximo. Para transformá-lo em DFA, resta criar novas transições até haver transições para todos os pares (estado, símbolo de entrada). Ora, resta adicionar o símbolo de entrada em falta, que deverão todos conduzir a um estado morto.