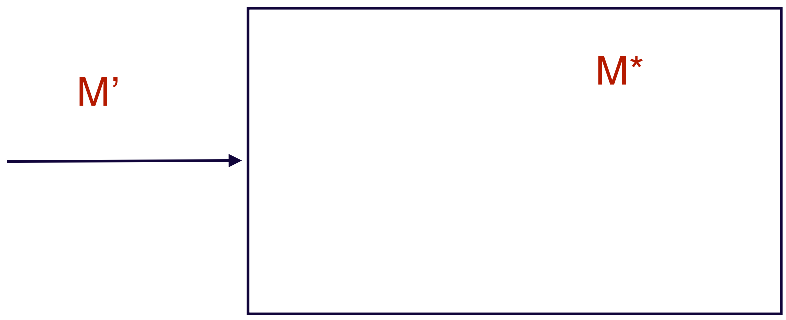
Uma discussão interessante sobre uma linguagem RE e o seu complemento

Definições

Considere $\langle M \rangle$ como a codificação de uma MT M sobre o alfabeto $\{0, 1\}$. Então:

- $L_e = \{\langle M \rangle | L(M) = \emptyset\}$
- $L_{ne} = \{\langle M \rangle | L(M) \neq \emptyset\}$
- $L_e = \overline{L_{ne}}$



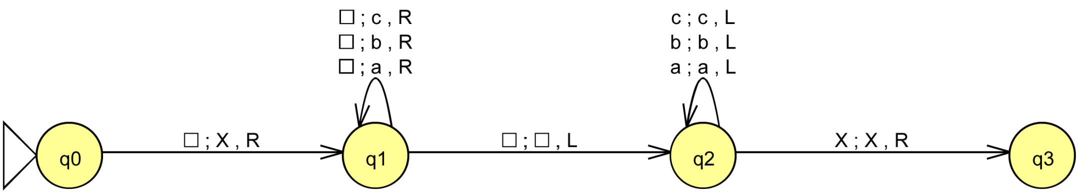
L_{ne} é RE

Teorema: A linguagem L_{ne} é recursivamente enumerável.
Prova:

1. Construir uma MT M^* que aceita como entrada a codificação de uma outra MT M';
2. M^* opera de forma não-determinística, fazendo escolhas de cadeias arbitrárias para serem testadas em M';
3. Em cada ramo da sua execução não-determinística, M^* gera uma cadeia e testa se M' aceita a mesma;
4. Para isso, M^* simula a máquina U que aceita a linguagem L_u;
5. Se algum caminho de M' for de aceitação, então M^* pára e aceita a sua entrada (M');

L_{ne} é RE

Exemplo: Geração não-determinística de cadeias arbirtrárias sobre o alfabeto $\{a, b, c\}$ para serem posteriormente testadas:



L_{ne} é RE

Em resumo:

- Se M' aceita alguma cadeia, $\overset{*}{M}$ "adivinha" essa cadeia e aceita M';
- Se M' não aceita nenhuma cadeia, então não há cadeia que conduza à aceitação em M' e $\overset{*}{M}$ não aceita M' (nesse caso, $\overset{*}{M}$ pode rejeitar M' ou entrar em loop);
- Portanto, $L(\overset{*}{M}) = L_{ne}$.

L_{ne} não é recursiva

Idéia geral:

- Fazer uma redução de L_u para L_{ne};
- Construir M' a partir de $\langle M, w\rangle$ tal que:
  - Se $w \in L(M)$, então $L(M') \neq \emptyset$;
  - Se $w \notin L(M)$, então $L(M') = \emptyset$;
- M' ignora a sua entrada e simula M com a entrada w;
- Se M aceita w, M' também aceita a sua entrada, qualquer que seja ela.

L_{ne} não é recursiva

Teorema: A linguagem L_{ne} não é recursiva.
Prova:

1. É suficiente provar a existência de um algoritmo que efetua a redução de L_u para L_{ne};
2. O algoritmo deve mapear $\langle M, w \rangle$ em M' de tal forma que $w \in L(M) \Leftrightarrow L(M') \neq \emptyset$;
3. A construção de M' a partir de $\langle M, w \rangle$ é detalhada a seguir;

L_{ne} não é recursiva

4. M' ignora a sua entrada x, qualquer que seja ela. M' substitui x por $\langle M, w \rangle$, tomando o cuidado de trocar os símbolos finais de x por brancos, caso $|x| > |\langle M, w \rangle|$;
5. M' posiciona a cabeça de leitura/escrita sobre o primeiro símbolo da cadeia $\langle M, w \rangle$;
6. M' simula a Máquina Universal U com a entrada $\langle M, w \rangle$;
7. Se U aceita $\langle M, w \rangle$, então M' pára e aceita a sua entrada, qualquer que seja ela e $L(M') \neq \emptyset$ (e se U não aceita $\langle M, w \rangle$, então M' não aceita nenhuma entrada e $L(M') = \emptyset$).

L_{ne} não é recursiva

Em resumo:

- Existe um algoritmo que reduz L_u para L_{ne};
  - M' aceita qualquer cadeia de entrada (e portanto $\langle M' \rangle \in L_{ne}$) sse $w \in L(M)$ (ou seja, se $\langle M, w \rangle \in L_u$);
  - M' não aceita nenhuma cadeia de entrada (e portanto $\langle M' \rangle \notin L_{ne}$) sse $w \notin L(M)$ (ou seja, se $\langle M, w \rangle \notin L_u$);
- Como L_u é indecidível, então L_{ne} é indecidível.

L_{ne} não é recursiva

Suponha que L_{ne} fosse decidível. Então seria possivel decidir L_u, da seguinte forma:

- Fazer a redução de $\langle M, w\rangle$ para M';
- Decidir se $L(M') \neq \emptyset$, ou seja, se $\langle M'\rangle \in L_{ne}$;
- Em caso afirmativo, $\langle M, w\rangle \in L_u$, ou seja, $w \in L(M)$;
- Em caso negativo, $\langle M, w\rangle \notin L_u$, ou seja, $w \notin L(M)$;

Mas como é sabido que L_u não é recursiva, então a suposição de que L_{ne} é recursiva é falsa.

L_e é não-RE

Teorema: L_e não é recursivamente enumerável.
Prova:

1. Suponha que L_e seja recursivamente enumerável;
2. Portanto, de acordo com um teorema anterior, tanto L_e quanto $\overline{L_e}$ devem ser recursivas;
3. Mas $\overline{L_e} = L_{ne}$;
4. Além disso, foi demonstrado que L_{ne} não é recursiva;
5. Logo, L_e não é recursivamente enumerável.