LINGUAGENS SENSÍVEIS AO CONTEXTO

Uma linguagem L é dita **sensível ao contexto** se $L - \{\epsilon\}$ é gerada por alguma gramática sensível ao contexto.

Formalmente, uma Máquina de Turing com fita limitada M é definida como:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, <, >, F)$$

onde:

- Q é o conjunto finito de estados;
- Σ é o alfabeto de entrada, composto por um conjunto finito de símbolos;
- Γ é o conjunto, também finito, de símbolos que podem ser lidos e/ou gravados na fita de trabalho. $\Sigma \subseteq \Gamma$;

- δ é a função parcial de transição, compreendendo os seguintes mapeamentos:
  - $Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{E, D\}}$
  - $Q \times \{<\} \rightarrow 2^{Q \times \{<\} \times \{D\}}$
  - $Q \times \{>\} \rightarrow 2^{Q \times \{>\} \times \{E\}}$
- q_0 é o estado inicial, $q_0 \in Q$;
- $<, > \notin \Gamma$ são símbolos respectivamente situados imediatamente à esquerda e imediatamente à direita da cadeia de entrada na configuração inicial;
- $F \subseteq Q$ é o conjunto de estados finais.

A **configuração** de uma Máquina de Turing com fita limitada é denotada através da tripla

$$(\alpha, q_k, \beta) \in \{<\}\Gamma^* \times Q \times \Gamma^*\{>\}$$

em que q_k é o estado corrente, $\alpha \in \{<\}\Gamma^*$ é a porção da cadeia de entrada que se encontra à esquerda do cursor de acesso e $\beta \in \Gamma^*\{>\}$ é a porção da cadeia de entrada que se encontra à direita do cursor de acesso, incluindo a posição por ele correntemente selecionada. Note-se que "<" e ">" podem ocorrer, cada um, no máximo uma vez em $\alpha\beta$, e sempre nos respectivos extremos.

A **configuração inicial** é $(<, q_0, \gamma >)$, onde q_0 é o estado inicial e $\gamma \in \Sigma^*$ é a cadeia de entrada a ser analisada. O cursor de acesso refere-se, portanto, ao símbolo inicial (mais à esquerda) da cadeia γ. A porção α da representação (α, q_k, β) corresponde, neste caso, apenas ao símbolo "$<$", pois não existe fita à esquerda deste delimitador. A **configuração final** é definida como (λ, q_f, μ), com $q_f \in F, \lambda \in \{<\}\Gamma^*$ e $\mu \in \Gamma^*\{>\}$.

O símbolo "$\vdash$" denota uma relação sobre as configurações de uma Máquina de Turing com fita limitada:

$$\vdash: \{<\}\Gamma^* \times Q \times \Gamma^*\{>\} \rightarrow \{<\}\Gamma^* \times Q \times \Gamma^*\{>\}$$

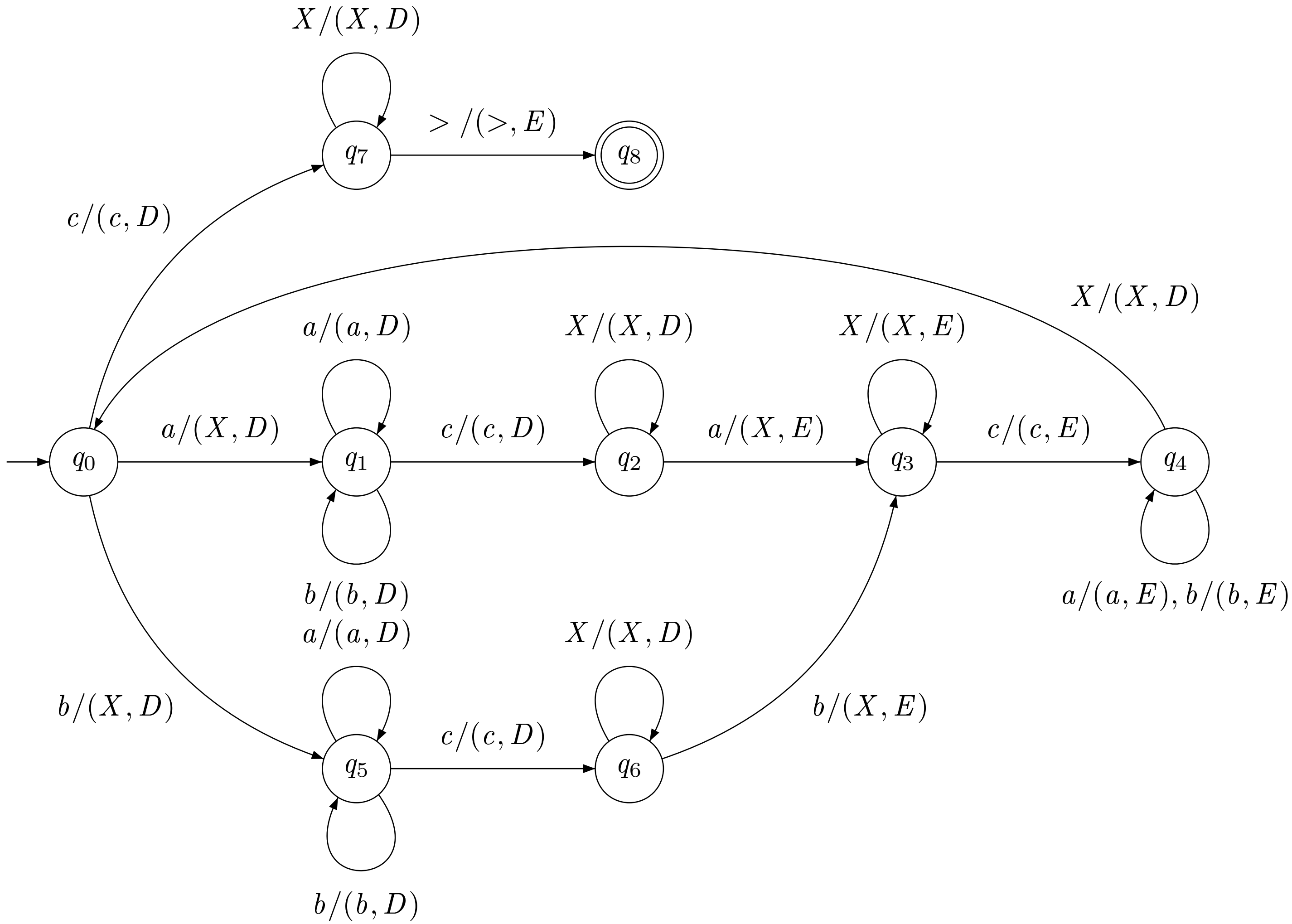
Portanto, a movimentação de uma configuração (α, q_k, β) para a configuração seguinte (α', q_m, β') é representada como:

$$(\alpha, q_k, \beta) \vdash (\alpha', q_m, \beta')$$

A linguagem aceita por uma Máquina de Turing com fita limitada é o conjunto de todas as cadeias que são capazes de conduzir o dispositivo desde a sua configuração inicial (única para cada cadeia de entrada) até uma configuração final qualquer, sem possibilidade de movimentação adicional. Formalmente:

$$L(M) = \{\gamma \in \Sigma^* \mid (<, q_0, \gamma >) \vdash^* (\lambda, q_f, \mu)$$

com $q_f \in F, \lambda \in \{<\}\Gamma^*$ e $\mu \in \Gamma^*\{>\}\}$.


$X/(X,D)$
q_7
$>/(>,E)$
q_8
$c/(c,D)$
$X/(X,D)$
$a/(a,D)$
$X/(X,D)$
$X/(X,E)$
$a/(X,D)$
$c/(c,D)$
$a/(X,E)$
$c/(c,E)$
q_0
q_1
q_2
q_3
q_4
$b/(b,D)$
$a/(a,E), b/(b,E)$
$a/(a,D)$
$X/(X,D)$
$b/(X,D)$
$b/(X,E)$
$c/(c,D)$
q_5
q_6
$b/(b,D)$

$$(<, q_0, abbcabb >) \vdash (< X, q_1, bbcabb >) \vdash (< Xb, q_1, bcabb >) \vdash$$
$$(< Xbb, q_1, cabb >) \vdash (< Xbbc, q_2, abb >) \vdash (< Xbb, q_3, cXbb >) \vdash$$
$$(< Xb, q_4, bcXbb >) \vdash (< X, q_4, bbcXbb >) \vdash (<, q_4, XbbcXbb >) \vdash$$
$$(< X, q_0, bbcXbb >) \vdash (< XX, q_5, bcXbb >) \vdash (< XXb, q_5, cXbb >) \vdash$$
$$(< XXbc, q_6, Xbb >) \vdash (< XXbcX, q_6, bb >) \vdash (< XXbc, q_3, XXb >) \vdash$$
$$(< XXb, q_3, cXXb >) \vdash (< XX, q_4, bcXXb >) \vdash (< X, q_4, XbcXXb >) \vdash$$
$$(< XX, q_0, bcXXb >) \vdash (< XXX, q_5, cXXb >) \vdash (< XXXc, q_6, XXb >) \vdash$$
$$(< XXXcX, q_6, Xb >) \vdash (< XXXcXX, q_6, b >) \vdash (< XXXcX, q_3, XX >) \vdash$$
$$(< XXXc, q_3, XXX >) \vdash (< XXX, q_3, cXXX >) \vdash (< XX, q_4, XcXXX >) \vdash$$
$$(< XXX, q_0, cXXX >) \vdash (< XXXc, q_7, XXX >) \vdash (< XXXcX, q_7, XX >) \vdash$$
$$(< XXXcXX, q_7, X >) \vdash (< XXXcXXX, q_7, >) \vdash (< XXXcXX, q_8, X >)$$

- Equivalência entre Máquinas de Turing com fita limitada e gramáticas sensíveis ao contexto;
- Toda linguagem livre de contexto também é sensível ao contexto;
- Existem linguagens que são sensíveis ao contexto mas não são livres de contexto;
- Existem linguagens que não são sensíveis ao contexto.

Considerar uma enumeração, em ordem lexicográfica crescente, das cadeias de Σ^+. Considere-se, em particular, $\Sigma = \{a, b\}$. Tal enumeração é correspondente à seqüência:

$$a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab, bba, bbb, ...$$

Tendo enumerado os elementos de $G = \{G_1,\ G_2,\ ...,\ G_n,\ ...\}$ e de $\Sigma^+ = \{\alpha_1,\ \alpha_2,\ ...,\ \alpha_n,\ ...\}$, passa-se agora a estabelecer uma função bijetora entre os dois conjuntos, de tal forma que os pares $(G_i, \alpha_i), i \geq 1$ sejam elementos dessa função:

$$\begin{array}{cccccc} G_1 & G_2 & G_3 & ... & G_n & ... \\ \updownarrow & \updownarrow & \updownarrow & & \updownarrow & \\ \alpha_1 & \alpha_2 & \alpha_3 & ... & \alpha_n & ... \end{array}$$

Defina-se agora a linguagem $L_R = \{\alpha_i \mid \alpha_i \notin L(G_i), \forall\, i \geq 1\}$. Em outras palavras, pertencem a L_R as cadeias α_i que não pertençam a $L(G_i)$. Tal verificação pode sempre ser feita para gramáticas sensíveis ao contexto, conforme demonstrado em teorema anterior sobre a equivalência deste tipo de gramáticas com as Máquinas de Turing com fita limitada.

Só existem duas hipóteses acerca de L_R: ou trata-se de uma linguagem que é sensível ao contexto, ou então trata-se de uma linguagem que não é sensível ao contexto.

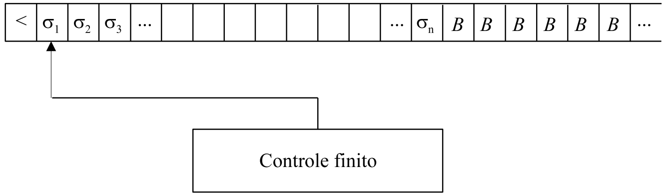
Considere-se que L_R seja uma linguagem sensível ao contexto. Se essa hipótese for verdadeira, deverá existir pelo menos uma gramática sensível ao contexto que a gere. Naturalmente, tal gramática deverá pertencer ao conjunto G. Seja G_i esta gramática. Se $L_R = L(G_i)$, então só existem duas possibilidades: α_i pertence ou não pertence a L_R.

- *Primeira possibilidade*: se $\alpha_i \notin L(G_i)$, então $\alpha_i \in L_R$, por hipótese, o que é uma contradição.

- *Segunda possibilidade*: se $\alpha_i \in L(G_i)$, então $\alpha_i \notin L_R$, por construção, o que também é uma contradição.

Logo, L_R não pode ser uma linguagem sensível ao contexto, e isso completa a demonstração.

LINGUAGENS RECURSIVAS



Formalmente, uma Máquina de Turing é definida como:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, <, B, F)$$

onde:

- Q é o conjunto finito não-vazio de estados;
- Σ é o alfabeto de entrada, formado por um conjunto finito não-vazio de símbolos;
- Γ é um conjunto, também finito e não-vazio, de símbolos que podem ser lidos e/ou escritos na fita de trabalho. $\Gamma \supseteq \Sigma$;

- δ é a função parcial de transição, $\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{E,D\}}$;
- q_0 é o estado inicial, $q_0 \in Q$;
- "$<$" $\in \Gamma$, "$<$" $\notin \Sigma$, é o símbolo que indica a primeira posição da fita de trabalho. Durante toda a operação da máquina, o símbolo "$<$" não pode ser gravado em nenhuma outra posição da fita;
- $B \in \Gamma, B \notin \Sigma$, é o símbolo utilizado para preencher inicialmente todas as posições à direta da cadeia de entrada na fita. Durante a operação da máquina, o símbolo B pode ser gravado em qualquer posição da fita;
- $F \subseteq Q$ é o conjunto de estados finais.

A **configuração** de uma Máquina de Turing é indicada por uma tripla

$$(\alpha, q_k, \beta) \in \Gamma^* \times Q \times \Gamma^*$$

onde q_k é o estado corrente, $\alpha \in \Gamma^*$ é a porção do conteúdo da fita de trabalho que se encontra à esquerda do cursor de acesso, e $\beta \in \Gamma^*$ é a porção do conteúdo da fita de trabalho que se encontra à direita do cursor de acesso, incluindo a posição correntemente apontada por ele.

A **configuração inicial** é $(<, q_0, \gamma)$, onde q_0 é o estado inicial e $\gamma \in \Sigma^*$ é a cadeia de entrada a ser analisada. O cursor de acesso aponta, portanto, exatamente o primeiro símbolo da cadeia de entrada γ. A porção α da representação (α, q_k, β) corresponde, neste caso, à cadeia unitária $<$. Uma **configuração final** é definida como (λ, q_f, μ), com $q_f \in F$ e $\lambda, \mu \in \Gamma^*$.

O símbolo "$\vdash$" denota uma relação sobre as configurações de uma Máquina de Turing:

$$\vdash: \Gamma^* \times Q \times \Gamma^* \rightarrow \Gamma^* \times Q \times \Gamma^*$$

Portanto, a movimentação de uma configuração para a seguinte é representada por:

$$(\alpha, q_k, \beta) \vdash (\alpha', q_m, \beta')$$

Em caso de tentativa de deslocamento do cursor de acesso para a esquerda da primeira posição da fita, a computação encerra-se anormalmente. Neste caso, a cadeia de entrada é rejeitada, não importando se o estado em que a máquina se encontra é final ou não.

A linguagem aceita por uma Máquina de Turing com fita infinita é o conjunto das cadeias que são capazes de conduzir o dispositivo desde a sua configuração inicial (única para uma determinada cadeia de entrada) até uma configuração final qualquer na qual ele esteja parado — sem possibilidade de movimentação. Formalmente,

$$L(M) = \{\gamma \in \Sigma^* \mid (<, q_0, \gamma) \vdash^* (\lambda, q_f, \mu), q_f \in F, \lambda, \mu \in \Gamma^*\}$$

admitindo-se, como condição de parada, que $\mu = \sigma\pi, \sigma \in \Gamma, \pi \in \Gamma^*$ e δ não seja definida para (q_f, σ).

Seja L a linguagem aceita por uma Máquina de Turing M. Se $w_1 \in L$, então M pára e aceita w_1. Considere-se, porém, a cadeia $w_2 \in \Sigma^* - L$. Neste caso, M pode tanto parar, rejeitando a entrada, quanto entrar em loop infinito, sem nunca atingir uma condição de parada.

Uma linguagem L é dita **recursiva** se existir pelo menos uma Máquina de Turing M tal que:

1. Para toda cadeia $w \in L$, M pára e aceita w;
2. Para toda cadeia $z \in \Sigma^* - L$, M pára e rejeita z.

Ou seja, M atinge a condição de parada para toda e qualquer cadeia que lhe é submetida, não importando se esta pertence ou não a L.

Uma linguagem recursiva é também dita **linguagem decidível**. Este termo denota o fato de que, para essa classe de linguagens, sempre existe pelo menos uma Máquina de Turing que aceita a referida linguagem, qualquer que seja a cadeia usada como entrada (não importa se ela pertence ou não à linguagem), e que sempre pára. A parada pode ocorrer com aceitação ou rejeição da cadeia.

- Não existe um formalismo gramatical para representar as linguagens recursivas;
- Toda linguagem sensível ao contexto é também recursiva;
- Existem linguagens recursivas que não são sensíveis ao contexto;
- Existem linguagens que não são recursivas.

A linguagem

$$L_R = \{\alpha_i \mid \alpha_i \notin L(G_i), \forall\, i \geq 1\}$$

não é sensível ao contexto. Por outro lado, qualquer que seja a cadeia $\beta \in \Sigma^*$, será sempre possível determinar mecanicamente se β pertence ou não a L_R, o que pode ser feito conforme o método descrito a seguir.

- Entrada: uma cadeia $\beta \in \Sigma^*$;
- Saída: SIM, se $\beta \in L_R$; NÃO, se $\beta \notin L_R$;
- Método:
  1. Compara-se β com cada uma das cadeias α_i, até que haja coincidência entre ambas;
  2. Seleciona-se a gramática G_i correspondente à cadeia α_i (lembrar que foi estabelecida uma função bijetora entre o conjunto das gramáticas e o conjunto das cadeias);

3. Determina-se se $\alpha_i \in L(G_i)$. Como se trata de linguagens sensíveis ao contexto, tal determinação pode sempre ser efetuada;
4. Se o resultado for que $\alpha_i \in L(G_i)$, então, por construção, $\alpha_i \notin L_R$ e a resposta é NÃO. Caso o resultado seja que $\alpha_i \notin L(G_i)$, então, por construção, $\alpha_i \in L_R$ e a resposta é SIM.

Logo, é sempre possível determinar se uma cadeia β qualquer pertence a L_R. Em outras palavras, L_R é uma linguagem decidível e, portanto, recursiva, sem ser sensível ao contexto.

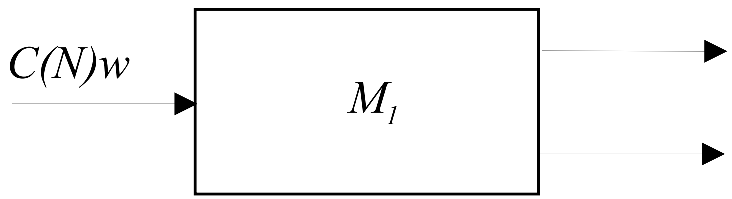
A classe das linguagens recursivas não é a mais abrangente que se conhece. Ao contrário, é possível demonstrar que existe pelo menos uma linguagem não-recursiva.

É suficiente provar que existe pelo menos uma linguagem não-recursiva. Essa demonstração consiste na definição da linguagem L_P, conforme abaixo, e na prova, por contradição, de que a mesma não pode pertencer à classe das linguagens recursivas. Em outras palavras, que esta linguagem não é decidível.

$$L_P = \{C(M)w \in \Sigma^* \mid M \text{ pára com a cadeia } w\}$$

onde:

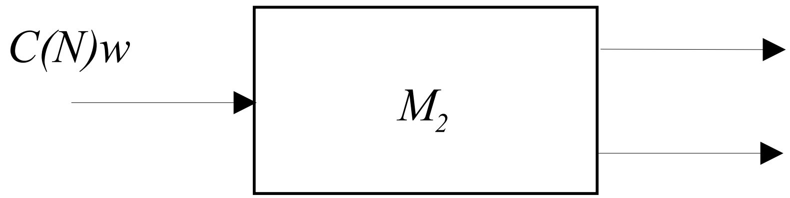
- $C(M)$ é a codificação de uma Máquina de Turing com fita limitada, cujo alfabeto de entrada é Σ, como cadeia sobre o próprio alfabeto Σ;
- $w \in \Sigma^*$ é uma cadeia de entrada qualquer para M.



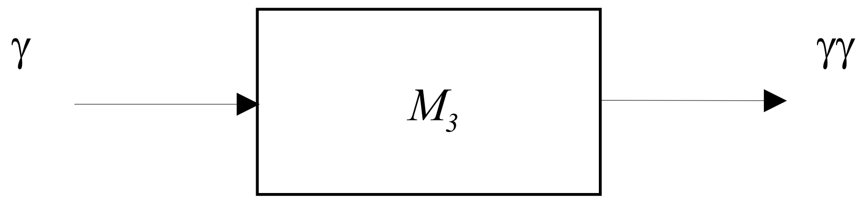
M_1 pára e aceita $C(N)w$:
N pára com a entrada w

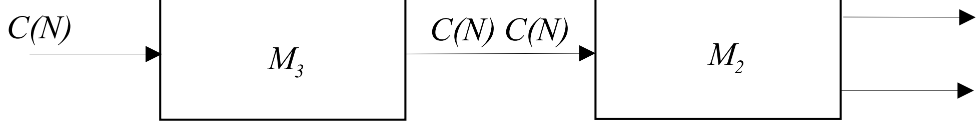
M_1 pára e rejeita $C(N)w$:
N não pára com a entrada w

M_2 executa uma seqüência infinita de movimentações:
N pára com a entrada *w*



M_2 pára e rejeita *C(N)w*:
N não pára com a entrada *w*

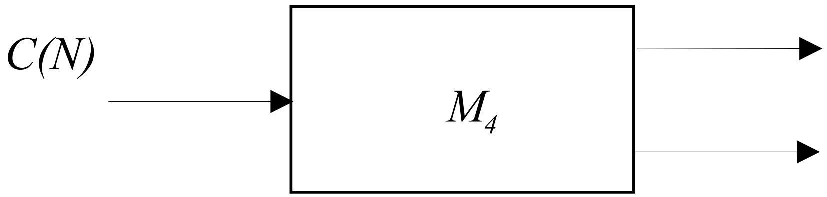

γ
M_3
γγ



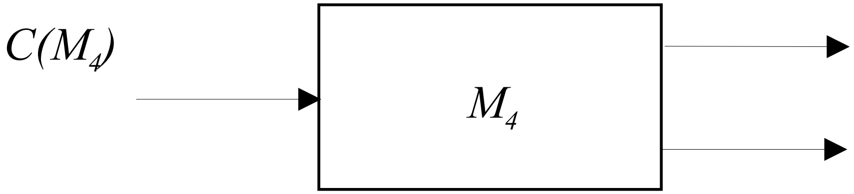
***M_2* executa uma seqüência infinita de movimentações:**
N pára com a entrada *C(N)*

***M_2* pára:**
N não pára com a entrada *C(N)*

***M_4* executa uma seqüência infinita de movimentações:**
N pára com a entrada *C(N)*



***M_4* pára:**
N não pára com a entrada *C(N)*



***M_4* executa uma seqüência infinita de movimentações:**
M_4 pára com a entrada $C(M_4)$

***M_4* pára:**
M_4 não pára com a entrada $C(M_4)$

Contradição: por um lado, temos a informação de que, ao analisar a cadeia $C(M_4)$, se a máquina M_4 parar, então M_4 executa uma seqüência infinita de movimentações. Por outro, que ao analisar a cadeia $C(M_4)$, se M_4 não parar, então M_4 pára. Tem-se, portanto, uma contradição. Logo, a nossa hipótese inicial não é válida, ou seja, L_P não pode ser uma linguagem recursiva.

Não existe solução para se determinar se uma certa máquina (ou programa) pára ao processar uma dada entrada. Ou seja, a linguagem

$$L_P = \{C(M)w \mid M \text{ pára com a entrada } w\}$$

é indecidível.

Naturalmente, esses resultados são válidos apenas para os casos de máquinas e entradas arbitrárias, não conhecidas *a priori*. Eventualmente, esses problemas podem ser resolvidos para combinações de máquinas e/ou entradas predeterminadas.

LINGUAGENS RECURSIVAMENTE ENUMERÁVEIS

Uma linguagem L é dita **recursivamente enumerável** (ou simplesmente **irrestrita**) se for aceita por pelo menos uma Máquina de Turing M. Ou seja:

1. Para toda cadeia $w \in L, M$ pára e aceita w;
2. Para toda cadeia $z \in \Sigma^* - L, M$ pára e rejeita z ou executa uma seqüência infinita de movimentações.

Uma linguagem L é dita **estritamente recursivamente enumerável** se, para toda e qualquer Máquina de Turing M que aceita L, existir pelo menos uma cadeia $z \in \Sigma^* - L$, tal que M inicie uma seqüência interminável de movimentações em seu processamento.

- Equivalência entre Máquinas de Turing e gramáticas irrestritas;
- Toda linguagem recursiva é também recursivamente enumerável;
- Existem linguagens que são recursivamente enumeráveis mas não são recursivas;
- Existem linguagens que não são recursivamente enumeráveis.

A linguagem L_P, conforme demonstrado anteriormente, é uma linguagem não-recursiva. A demonstração de que L_P é uma linguagem recursivamente enumerável, feita a seguir, basta para provar que a classe das linguagens recursivas constitui subconjunto próprio da classe das linguagens recursivamente enumeráveis.

$$L_P = \{C(M)w \in \Sigma^* \mid M \text{ pára com a entrada } w\}$$

onde

- $C(M)$ é a codificação de uma Máquina de Turing com fita limitada, cujo alfabeto de entrada é Σ, como cadeia sobre o próprio alfabeto Σ;
- $w \in \Sigma^*$ é uma cadeia de entrada qualquer de M;

Para provar que L_P é recursivamente enumerável, basta mostrar que L_P é aceita por uma Máquina de Turing N, a qual pode ser obtida conforme o algoritmo a seguir.

- Entrada: a linguagem L_P;
- Saída: uma Máquina de Turing N que aceita L_P;
- Método:
  1. N simula a operação da Máquina M no processamento da cadeia de entrada w, onde $C(M)w$ é a cadeia de entrada de N. N possui duas fitas: na primeira é gravada a cadeia $\gamma = C(M)w$ que se pretende analisar; a segunda é usada como área de trabalho de N, e servirá para guardar, em cada instante, o estado corrente de M;
  2. Duas possibilidades podem acontecer durante o processamento de N:

- M pára com a entrada w. Neste caso, N deverá encerrar sua operação aceitando $C(M)w$;
- M inicia uma seqüência infindável de movimentações com w. Neste caso, N inevitavelmente também iniciará, correspondentemente, com $C(M)w$, uma seqüência interminável de movimentações.

Assim, M aceita w se e somente se N aceita w. Logo, L_P é uma linguagem recursivamente enumerável.