A Linguagem da diagonalização

Supondo que exista uma M*i,* a “*i*-éssima máquina de Turing”, a TM M cujo código é Wi, o i-ésimo string binário.

Isto é, para tal i, esse Mi é uma Máquina de Turing que para imediatamente quando aplicada sobre qualquer entrada. Desse modo, L(Mi) é {} se Wi não for válido para TM.

* A linguagem Ld, a **linguagem da diagonalização**, é o conjunto de strings wi tais que wi não está em L(Mi).

Ou seja, Ld, são todos os strings w tais que a TM M cujo código é w não aceita quando recebe w como entrada.

A razão do nome linguagem de “diagonalização” pode ser entendida através da figura X.

A tabela informa, para todo *i* (linha) e *j* (coluna), se a máquina de Turing Mi aceita a cadeia de entrada Wj: 1 significa “sim” e 0 significa “não”.

Pode-se pensar na *i­-*ésima linha como o vetor característico para a linguagem L(Mi); ou seja, os 1’s nesta linha indicam as cadeias que são elementos desta linguagem.

Os valores na diagonal dizem se Mi aceita Wi.

Para construir a Ld, complementa-se a diagonal.

Ou seja, se a figura X for a tabela a ser seguida, então a diagonal complementada seria 1,0,0,0, ...

Portanto, Ld contém W1 = λ, não contém W2 a W4, que são 0, 1 e 00, e assim por diante.

O fato de complementar a diagonal para construir o vetor característico de uma linguagem que não pode ser a linguagem que aparece em nenhuma linha é chamado de **diagonalização.**

Isto funciona porque o complemento da diagonal é ele próprio um vetor característico que descreve a pertinência em alguma linguagem, a Ld.

Este vetor caratrístico discorda em alguma coluna com toda linha da tabela.

Portanto, o comportamento da diagonal não pode ser o vetor característico de nenhuma máquina de Turing.

**Teorema:** Ld não é uma linguagem recursivamente enumerável (RE). Ou seja, não à nenhuma máquina de Turing que aceite Ld.

**Prova:** Suponha que Ld seja L(M) para alguma máquina de Turing M. Como Ld é uma linguagem sobre o alfabeto {0,1}, M estaria na lista das máquinas de turing, já que esta lista inclui todas as máquinas de Turing com alfabeto de entrada {0,1}. Portanto, há pelo menos um código para M, por exemplo, M = Mi. Agora será que Wi está em Ld?

Se Wi está em Ld, então Mi aceita W­i. Mas então, pela definição de Ld, Wi não está em Ld, porque Ld contém apenas aqueles Wj tal que Mj **não** aceita Wj.

Similarmente, se Wi não está em Ld, então Mi não aceita Wi. Portanto, por definição de Ld, Wi **está** em Ld.

Como Wi não pode estar e não estar em Ld ao mesmo tempo, há uma contradição na suposição de que M existe. Ou seja, Ld não é uma linguagem recursivamente enumerável.