PCS2046- Lógica Computacional

Aula 13 | Exercício 11

Professor Doutor Ricardo Rocha

9783640 - Luís Henrique Barroso Oliveira

9835623 - Rodrigo Vali Cebrian

11259715 - Vanderson da Silva dos Santos

São Paulo, 10 de abril de 2023

Sumário

Exercício 1

> Mostre que a classe de linguagens Turing-decidíveis é fechada sobre união, concatenação, estrela de Kleene, complemento e interseção.

Solução

A classe de linguagens Turing-decidíveis podem também ser chamadas de linguagens recursivas. Sendo assim, quando são recursivas e enumeráveis, são fechadas em relação ao Fecho de Kleene, à concatenação, à união e à interseção. Porém, em relação ao complemento, basta ser recursiva para seu complemento também ser.

Exercício 2

Mostre que o problema da parada da máquina de Turing, o conjunto de pares (*M*, *w*) tais que a MT *M* pára quando fornecida uma entrada *w*, é recursivamente enumerável mas não recursivo.

Solução

Recursivamente enumerável: problema parcialmente decidível (linguagem é reconhecida pela Máquina de Turing) → linguagem reconhecível

Recursivo: problema decidível (a Máquina de Turing sempre para quando aceita ou rejeita uma cadeia) → linguagem decidível

Demonstração por absurdo

Assumimos que seja a linguagem seja decidível.

Suponha que exista um decisor *H* para a linguagem, que se comporte da seguinte forma:

H(<M,w>) = aceite, se M aceita w,
rejeite, se M não aceita w

Se o decisor H existe, é possível construir uma nova MT *D* que utiliza *H* como uma subrotina.

A MT *D* recebe uma MT *M* como entrada e pergunta ao decisor *H* se *M* para ou entra em *loop* quando *M* recebe sua própria descrição *<M>* como entrada.

O que o decisor responder, a MT *D* inverte seu comportamento.

D = "Sobre a entrada <M>, onde M é uma MT:
1. Rode H sobre a entrada <M, <M>>.
2. Dê como saída o oposto do que H dá como saída: se H aceita, rejeita; se H rejeita, aceite."

Note que *D* terá o seguinte comportamento:

D(<M>) = aceite, se M não aceita <M>,

rejeite, se M aceita <M>

Assim verificamos o que acontece quando rodamos a MT *D* com sua própria descrição <*D*> como entrada.

O comportamento de *D* com <*D*> como entrada será:

D(<D>) = aceite, se D não aceita <D>,
rejeite, se D aceita <D>

Mas note que isso é um absurdo:

D aceita a cadeia <*D*> se, e somente se, *D* não aceita a cadeia <*D*>

Evidentemente é impossível *D* aceitar e não aceitar uma mesma entrada ao mesmo tempo, logo chegamos a uma contradição e devemos concluir que nossa hipótese era falsa: ou seja, não pode existir decisor *H* para o linguagem.

Exercício 3

Sejam $L_1, L_2, \ldots L_k$ linguagens sobre um alfabeto Σ tais que:

1. $\forall i, j\ L_i \cap L_j = \emptyset$.
2. $L_1 \cup L_2 \cup \cdots \cup L_k = \Sigma^*$.
3. $\forall i, L_i$ é recursivamente enumerável.

Mostre que cada uma das linguagens L_i é recursiva.

Solução

Também chamadas por Turing-Acceptable ou decidíveis, as linguagens recursivas são aquelas que uma máquina de Turing é capaz de aceitar.

Como não existam intersecções entre as linguagens Li (item 1) e a união de todas as Li gera Σ∗ (item 2), então o complemento de uma Lj qualquer é a união das outras Li, em que i ≠ j. E do item 3, todas as Lis são recursivas, portanto a união delas também será.

Qualquer linguagem Lx, representada por uma cadeia qualquer w, poderá ser aceita pela máquina de Turing de Li ou de seu complemento, ou qualquer união de Li’s. Portanto, cada linguagem Li é recursiva.