15.3 O Teorema de Cook

Vamos transcrever a demonstração de que CIRCUIT SAT é $\mathcal{NP}$-completo, deixando para as próximas Notas de Aula a redução de CIRCUIT SAT para SAT, como sistematizado em [32]. Destacamos, contudo, que o Teorema de Cook originalmente contempla ambas as variantes do *Problema da Satisfatibilidade Booleana*.

Teorema 15.4 (Teorema de Cook). *CIRCUIT SAT é $\mathcal{NP}$-completo.*

Demonstração. Já sabemos que CIRCUIT SAT é um problema da classe $\mathcal{NP}$ (cf. Exercício resolvido 14.4). Resta-nos mostrar apenas que se trata de um problema $\mathcal{NP}$-completo. Tomemos para tanto uma linguagem L, sobre um alfabeto Σ, qualquer de $\mathcal{NP}$. O que queremos mostrar é a existência duma redução R tal que $L \prec_R$ CIRCUIT SAT. Combinando a Definição 12.3 com o Exercício 15.2, temos que existe uma MTN $N = (Q, \Sigma, \Delta, q_0)$ e um inteiro positivo k tal que, para toda entrada x para N satisfazendo $|x| \geqslant 2$, $t_N(x) \leqslant |x|^k$. Podemos também assumir sem perda de generalidade que, para todo $q \in Q$ e todo $s \in \Sigma$, $|\Delta(q,s)| = 2$ (cf. Exercício 15.3), de modo que as escolhas em $\Delta(q,s)$ são indexadas por 0 e por 1.

Seja $x \in \Sigma^*$ uma entrada para N e $c \in \{0,1\}^{|x|^k+1}$ uma sequência de $|x|^k + 1$ escolhas que determina uma computação de N para x, ainda que a computação não use todas as escolhas da sequência. Observemos que de $\mathsf{T}(N,x,c)$, a tábua da computação de N para x sob c:

- a primeira linha (indexada por 0) é $\rhd \rhd_{q_0} x \sqcup^{|x|^k - |x|}$, a primeira célula de cada linha é sempre $\rhd$, e a última célula de cada linha é sempre $\sqcup$;
- a célula T_{ij} para $0 < i \leqslant |x|^k + 1$ e $0 < j < |x|^k + 1$ depende exclusivamente de apenas três células da linha superior — a saber, $\mathsf{T}_{i-1,j-1}$, $\mathsf{T}_{i-1,j}$ e $\mathsf{T}_{i-1,j+1}$ — e da escolha feita da linha $i-1$ para a linha i, ou seja, do *bit* c_i, sendo $c = c_1 c_2 \cdots c_{|x|^k+1}$. Ou seja, cada célula T_{ij} é *função* de $\mathsf{T}_{i-1,j-1}$, $\mathsf{T}_{i-1,j}$, $\mathsf{T}_{i-1,j+1}$ e c_i. Ou seja, existe uma função $f : \Gamma^3 \to \Gamma$ tal que

$$f(\mathsf{T}_{i-1,j-1}, \mathsf{T}_{i-1,j}, \mathsf{T}_{i-1,j+1}, c_i) = \mathsf{T}_{ij} \qquad \forall ij, 0 < i \leqslant |x|^k + 1, 0 < j < |x|^k + 1.$$

Vamos agora transformar $\mathsf{T}(N,x,c)$ numa tabela binária $\mathsf{S}(N,x,c)$. Isso é fácil se levarmos em consideração que cada símbolo $\gamma \in \Gamma$ pode ser codificado numa cadeia binária $\langle\gamma\rangle$ com $m = \lceil \lg|\Gamma| \rceil$ *bits*, sendo Γ o alfabeto de $\mathsf{T}(N,x,c)$. Assim, para obtermos $\mathsf{S}(N,x,c)$, trocamos cada célula T_{ij} em $\mathsf{T}(N,x,c)$ pelos m *bits* que codificam aquela célula: $\mathsf{S}_{ij1}, \mathsf{S}_{ij2}, \ldots, \mathsf{S}_{ijm}$. Logo, para cada $\ell \in [1..m]$, cada posição $\mathsf{S}_{ij\ell}$, se $0 < i \leqslant |x|^k + 1$ e $0 < j < |x|^k + 1$, é *função* de $\mathsf{S}_{i-1,j-1,1}, \ldots, \mathsf{S}_{i-1,j-1,m}$, de $\mathsf{S}_{i-1,j,1}, \ldots, \mathsf{S}_{i-1,j,m}$, de $\mathsf{S}_{i-1,j+1,1}, \ldots, \mathsf{S}_{i-1,j+1,m}$ e de c_i. Então, existem m funções booleanas $(3m+1)$-árias $F_1, \ldots, F_m$ tais que

$$F_\ell \colon (\mathsf{S}_{i-1,j-1,1}, \ldots, \mathsf{S}_{i-1,j-1,m}, \mathsf{S}_{i-1,j,1}, \ldots, \mathsf{S}_{i-1,j,m}, \mathsf{S}_{i-1,j+1,1}, \ldots, \mathsf{S}_{i-1,j+1,m}, c_i) \mapsto \mathsf{S}_{ij\ell},$$
$$\forall ij\ell, 0 < i \leqslant |x|^k + 1, 0 < j < |x|^k + 1, 1 \leqslant \ell \leqslant m.$$

Como toda função booleana pode ser computada por um circuito booleano, temos que existem m circuitos booleanos $C_1, \ldots, C_m$ que computam respectivamente as funções booleanas $F_1, \ldots, F_m$.

Finalmente, apresentamos a redução R no Algoritmo 15.1. A redução, ao receber $x \in \Sigma^*$ como entrada, constrói um circuito D juntando várias cópias dos circuitos $C_1, \ldots, C_m$, cada cópia para cada *bit* de cada posição da tábua da computação de N para x sob uma sequência de escolhas c. Note-se que a redução R recebe apenas x como entrada, não a sequência c. Portanto, as portas de entrada do circuito D correspondentes às escolhas de c são deixadas como variáveis, enquanto que todas as outras portas de entrada de D já recebem os valores-verdade correspondentes à codificação de x ou dos símbolos $\rhd$, $\rhd_{q_0}$ e $\sqcup$. O circuito D ainda incorpora um subcircuito E que testa se o estado final presente na última linha da tábua $\mathsf{T}(N,x,c)$ é q_{sim}, de modo que a porta de saída do circuito D é a porta de saída de E.

$R(x = x_1 x_2 \cdots x_n)$:

1 **se** $|x| \geqslant 2$, **então**:
2 encontre a função f;
3 transforme a função f nas m funções $F_1, \ldots, F_m$;
4 construa os m circuitos $C_1, \ldots, C_m$;
5 inicialize um circuito D sobre um conjunto de variáveis $X = \{c_1, \ldots, c_{n^k+1}\}$, criando uma porta de entrada para cada variável do circuito;
6 $s_0 \leftarrow \rhd$; $s_1 \leftarrow \rhd_{q_0}$; $s_2 \leftarrow x_1$; $\ldots$; $s_{n+1} \leftarrow x_n$; $s_{n+2} \leftarrow \sqcup$; $\ldots$; $s_{n^k+1} \leftarrow \sqcup$;
7 **para** j **de** 0 **até** $n^k + 1$, **faça**:
8 crie m portas de entrada $g_{0j1}, \ldots, g_{0jm}$ para D rotuladas não com variáveis, mas com os valores-verdade correspondentes aos m *bits* da codificação de s_j;
9 **para** i **de** 1 **até** $n^k + 1$, **faça**:
10 crie m portas de entrada $g_{i01}, \ldots, g_{i0m}$ para D rotuladas com os m valores-verdade correspondentes aos m *bits* da codificação de $\rhd$;
11 **para** j **de** 1 **até** n^k, **faça**:
12 **para** ℓ **de** 1 **até** m, **faça**:
13 crie uma cópia do circuito C_ℓ, chamando sua porta de saída de g_{ijl}, alimentando suas entradas com as portas $g_{i-1,j-1,1}, \ldots, g_{i-1,j-1,m}$, $g_{i-1,j,1}, \ldots, g_{i-1,j,m}$, $g_{i-1,j+1,1}, \ldots, g_{i-1,j+1,m}$ e a porta de entrada correspondente à variável c_i;
14 crie m portas de entrada $g_{i,n^k+1,1}, \ldots, g_{i,n^k+1,m}$ para D rotuladas com os m valores-verdade correspondentes aos m *bits* da codificação de $\sqcup$;
15 incorpore ao circuito D um circuito E alimentado por todas as portas $g_{n^k+1,j,\ell}$ $(0 < j < n^k + 1,\ 1 \leqslant \ell \leqslant m)$ que devolva **V** se e somente se existe algum j tal que os valores das portas $g_{n^k+1,j,1}, \ldots, g_{n^k+1,j,m}$ codificam algum símbolo $s_{q_{\text{sim}}}$ para $s \in \Sigma \cup \{\rhd, \sqcup\}$;
16 **senão**, construa o circuito D trivialmente;
17 **devolva** D.

Algoritmo 15.1

Vamos mostrar que a redução funciona. Se $x \in L$, então, existe alguma sequência de escolhas $c \in \{0,1\}^{|x|^k+1}$ tal que o estado final identificado na última linha da $\mathsf{T}(N,x,c)$ é q_{sim}. Valorando-se as variáveis do circuito D com os valores-verdade correspondentes às escolhas $c_1, \ldots, c_{|x|^k+1}$, obtemos o valor do circuito **V**, pela definição do circuito E. Em contrapartida, se $x \notin L$, então, toda valoração das variáveis de D faz o valor do circuito tornar-se **F**, pois não há sequência de escolhas c que faça aparecer na última linha da tábua $\mathsf{T}(N,x,c)$ o estado q_{sim}.

Encerramos a demonstração alegando que a redução R se trata de uma redução polinomial, o que se fundamenta pelas observações a seguir.

1. Podemos encontrar a função f na linha 2 construindo todas as possíveis tuplas

$$(T_{i-1,j-1}, T_{i-1,j}, T_{i-1,j+1}, c_i) \in \Gamma^3 \times \{0,1\}$$

e calculando a imagem T_{ij} de cada tupla pela função f através de uma consulta à tabela da função de transição Δ de N. Como $|\Gamma^3 \times \{0,1\}|$ e $|\Delta|$ são constantes independentes de $n = |x|$, tudo isso pode ser feito em tempo constante, bem como a transformação de f em $F_1, \ldots, F_m$ na linha 3 e a construção dos circuitos $C_1, \ldots, C_m$ na linha 4, já que m também é uma constante que independe de n.

2. Criar as $n^k + 1$ portas de entrada para D na linha 5 custa tempo $O(n^k)$, mesmo tempo da linha 6.

3. As linhas 7–8 custam tempo $O(n^k \cdot m) = O(n^k)$.

4. O laço da linha 9 é iterado $O(n^k)$ vezes, custando cada iteração:

   (a) o tempo $O(m) = O(1)$ da linha 10;

   (b) o tempo do laço da linha 11, que é $O(n^k \cdot m)$ vezes o tempo da linha 13, que é constante, pois nada na linha 13 depende de n;

   (c) o tempo $O(m) = O(1)$ da linha 14.

5. Como o tamanho do circuito E é proporcional a $n^k + 2$, o tempo da linha 15 é $O(n^k)$.

Sumarizando, o tempo da redução R é

$$T_R(n) \leqslant O(1) + O(n^k) + O(n^k)\Big(O(1) + O(n^k)\Big) + O(n^k) = O((n^k)^2),$$

polinomial como queríamos. ♦

15.4 Exercícios

Exercício 15.1. Mostre que o *Problema da Parada* é $\mathcal{NP}$-difícil.

Exercício 15.2. Mostre que, se M é uma Máquina de Turing — Determinística ou Não-determinística, tanto faz — com complexidade de tempo polinomial, então, existe um inteiro positivo k tal que, para toda entrada x para M satisfazendo $|x| \geqslant 2$, $t_M(x) \leqslant |x|^k$.

Consulte a bibliografia para este exercício!

Exercício 15.3. Mostre que para toda MTN N existe uma MTN N' com $d_{N'} = 2$ que decide $L(N)$.

Exercício 15.4. Construa uma Máquina de Turing M com uma só fita que decide se um número representado em binário é uma potência de 2. Construa $\mathsf{T}(M,x)$ sendo $x = 100$ (que representa o número 4). Construa também $\mathsf{S}(M,x)$.

Exercício 15.5. Mostre que a redução construída na demonstração do Teorema de Cook pode ser implementada em espaço logarítmico.