# Limitantes Inferiores de Espaço e Tempo para SAT

Nicollas Sdroievski

3 de Junho de 2019

## 1 Ideia Geral

Hoje veremos um tipo de resultado na fronteira do que conhecemos sobre a complexidade de SAT e outros problemas NP-completos. Mostraremos que é impossível resolver SAT em tempo linear e espaço logarítmico ao mesmo tempo.

Para provar esse resultado, aplicaremos algumas ideias relacionadas à Hierarquia Polinomial. Mostraremos, incondicionalmente, que é possível tornar computações limitadas em tempo e espaço ao mesmo tempo mais rápidas ao introduzir quantificadores. Na sequência, assumindo que SAT possui um algoritmo que usa pouco tempo e espaço, mostramos que é possível remover um quantificador sem que haja uma perda significativa de tempo. Combinando esses resultados, obteremos uma contradição, garantindo que tal algoritmo para SAT não pode existir.

## 2 Preliminares

Os resultados que vamos apresentar se aplicam tanto para máquinas de Turing com acesso sequencial quanto máquinas de Turing com acesso indexado (estilo RAM). Um ingrediente importante é uma versão especial do Teorema de Cook-Levin (várias provas para essa versão existem, e deixamos a sugestão da Seção 2.3.1 de [1] e o vídeo do curso de complexidade da Carnegie Mellon University `https://www.youtube.com/watch?v=aR6kEDPBXbE`).

Definimos as classe QLIN e NQLIN, dos problemas decidíveis em tempo determinístico e não-determinístico $n \log^c(n)$ para alguma constante $c$ fixa, respectivamente. É comum se referir a esse limite de tempo como quasi-linear ou $n \cdot \mathsf{polylog}(n)$.

**Definição 1.**

$$\text{QLIN} = \bigcup_{c>0} \mathsf{TIME}(n \log^c(n)).$$
$$\text{NQLIN} = \bigcup_{c>0} \mathsf{NTIME}(n \log^c(n)).$$

Podemos "apertar" o Teorema de Cook-Levin para mostrar que $\mathsf{SAT}$ é completo para a classe NQLIN através de reduções determinísticas quasi-lineares.

**Teorema 1.** $\mathsf{SAT}$ *é* $\leq_m^{\text{QLIN}}$*-completo para* NQLIN. *Além disso, cada bit da fórmula é computável em tempo e espaço* $\mathsf{polylog}(n)$*.*

Definimos também as classes $\mathsf{DTISP}(t(n), s(n))$, que limitam simultaneamente o tempo e o espaço necessários para decidir linguagens.

**Definição 2.** *Sejam* $t, s : \mathbb{N} \to \mathbb{N}$ *duas funções. Dizemos que uma linguagem* $L \in \mathsf{DTISP}(t(n), s(n))$ *se existe uma máquina de Turing determinística* $M$ *de tempo* $t(n)$ *e espaço* $s(n)$ *que decide* $L$*.*

# 3 Teorema e Demonstração

**Teorema 2.** $\mathsf{SAT} \notin \mathsf{DTISP}(n^{1.1}, n^{0.1})$.

*Demonstração.* Primeiro apontamos que é suficiente mostrar que $\mathsf{NTIME}(n) \notin \mathsf{DTISP}(n^{1.2}, n^{0.2})$.

**Afirmação 1.** *Caso* $\mathsf{NTIME}(n) \notin \mathsf{DTISP}(n^{1.2}, n^{0.2})$*, então* $\mathsf{SAT} \notin \mathsf{DTISP}(n^{1.1}, n^{0.1})$.

*Demonstração.* Por contraposição: assuma $\mathsf{SAT} \in \mathsf{DTISP}(n^{1.1}, n^{0.1})$ e considere $L \in \mathsf{NTIME}(n)$. Pela versão do Teorema de Cook-Levin apresentada anteriormente, podemos, a partir de $x$ com $|x| = n$, construir $\phi_x$ com $|\phi_x| \leq n \cdot \mathsf{polylog}(n)$ tal que

$$x \in L \iff \phi_x \in \mathsf{SAT}.$$

Usando o fato de que $\mathsf{SAT} \in \mathsf{DTISP}(n^{1.1}, n^{0.1})$, podemos decidir se $\phi_x \in \mathsf{SAT}$ em tempo $(n \cdot \mathsf{polylog}(n))^{1.1} = O(n^{1.2})$ e espaço $(n \cdot \mathsf{polylog}(n))^{0.1} = O(n^{0.2})$, concluindo que $L \in \mathsf{DTISP}(n^{1.2}, n^{0.2})$. □ (afirmação)

Agora mostramos que é possível acelerar uma computação limitada em tempo e espaço ao introduzir dois quantificadores.

**Afirmação 2.** $\mathsf{DTISP}(n^{12}, n^2) \subseteq \Sigma_2\mathsf{TIME}(n^8)$.

*Demonstração.* Seja $L \in \mathsf{DTISP}(n^{12}, n^2)$, $x \in \{0,1\}^n$ e $M$ a MT determinística de tempo $n^{12}$ e espaço $n^2$ que decide $L$. Considere o grafo de configurações $G_{M,x}$. A string $x$ pertence a $L$ se e somente se existe um caminho de tamanho $n^{12}$ nesse grafo da configuração inicial para a configuração de aceitação. Quebramos esse caminho em "pedaços" de tamanho $n^6$, ou seja, também é verdade que $x \in L$ se e somente se existem $n^6$ configurações $C_1, C_2, \ldots, C_{n^6}$ tal que $C_{n^6}$ é uma configuração de aceitação e para todo $1 \leq i \leq n^6$, $C_{i-1}$ alcança $C_i$ em até $n^6$ passos. Reescrevendo:

$$x \in L \iff \exists(C_1, C_2, \ldots, C_{n^6})\forall(1 \leq i \leq n^6) \mid C_{i-1} \text{ alcança } C_i \text{ em até } n^6 \text{ passos e } C_{n^6} \text{ é uma configuração de aceitação.}$$

Perceba que todas as $n^6$ configurações $C_1, C_2, \ldots, C_{n^6}$ podem ser representadas usando $cn^8$ bits para alguma constante $c$, enquanto os índices $i$ podem ser representados usando $\lceil \log(n^6) \rceil = \lceil 6 \log n \rceil$ bits. Além disso, temos que as condições necessárias podem ser computadas em tempo $O(n^8)$ por um algoritmo $M'$, logo:

$$x \in L \iff \exists u \in \{0,1\}^{cn^8} \forall v \in \{0,1\}^{\lceil 6 \log n \rceil} M'(x, u, v) = 1.$$

Concluindo que $L \in \Sigma_2\mathsf{TIME}(n^8)$. □ (afirmação)

Usando a hipótese $\mathsf{NTIME}(n) \subseteq \mathsf{DTISP}(n^{1.2}, n^{0.2}) \subseteq \mathsf{DTIME}(n^{1.2})$, mostramos que é possível remover o quantificador $\forall$ de $\Sigma_2\mathsf{TIME}(n^8)$ com um *overhead* de tempo pequeno.

**Afirmação 3.** *Se* $\mathsf{NTIME}(n) \subseteq \mathsf{DTIME}(n^{1.2})$*, então* $\Sigma_2\mathsf{TIME}(n^8) \subseteq \mathsf{NTIME}(n^{9.6})$.

*Demonstração.* Seja $L \in \Sigma_2\mathsf{TIME}(n^8)$, então existe uma MT determinística $M$ de tempo $O(n^8)$ tal que, para todo $x \in \{0,1\}^*$

$$x \in L \iff \exists u \in \{0,1\}^{c|x|^8} \forall v \in \{0,1\}^{d|x|^8} M(x, u, v) = 1$$

para valores de $c$ e $d$ constantes. Considere a linguagem $L'$ definida da seguinte maneira:

$$\langle x, u \rangle \in L' \iff \forall v \in \{0,1\}^{d|x|^8} M(x, u, v) = 1$$

Perceba que como o tempo de execução de $M$ é $O(|x|^8)$ e $|\langle x, u\rangle| = O(|x|^8)$, $L' \in \mathsf{coNTIME}(n)$. Pela hipótese, $\mathsf{coNTIME}(n) \subseteq \mathsf{DTIME}(n^{1.2})$, pois $\mathsf{DTIME}(n^{1.2})$ é fechada sob complemento. Logo, existe uma MT determinística $D$ de tempo $O(n^{1.2})$ que decide $L'$. Assim

$$x \in L \iff \exists u \in \{0,1\}^{c|x|^8} \langle x, u\rangle \in L' \iff \exists u \in \{0,1\}^{c|x|^8} D(x,u) = 1.$$

Como o tamanho da entrada de $D$ é $O(|x|^8)$, temos que seu tempo de execução em função de $|x|$ é $O((|x|^8)^{1.2}) = O(|x|^{9.6})$. Concluímos então que $L \in \mathsf{NTIME}(n^{9.6})$. □ (afirmação)

Combinando os resultados das afirmações provadas, mostramos que a hipótese $\mathsf{NTIME}(n) \in \mathsf{DTISP}(n^{1.2}, n^{0.2})$ leva a uma contradição. Usando um argumento de preenchimento junto da hipótese (**Exercício**), podemos mostrar que $\mathsf{NTIME}(n^{10}) \in \mathsf{DTISP}(n^{12}, n^2)$. Então aplicamos a seguinte sequência de resultados:

$$\begin{aligned} \mathsf{NTIME}(n^{10}) &\subseteq \mathsf{DTISP}(n^{12}, n^2) \\ &\subseteq \Sigma_2\mathsf{TIME}(n^8) \text{ (Afirmação 2)} \\ &\subseteq \mathsf{NTIME}(n^{9.6}) \text{ (Afirmação 3).} \end{aligned}$$

O que contradiz o Teorema de Hierarquia de Tempo não-Determinístico. □

# Referências


[1] Dieter van Melkebeek. A Survey of Lower Bounds for Satisfiability and Related Problems. *Foundations and Trends in Theoretical Computer Science*, 2(3):197–303, 2006.