ITC: Introdução à Teoria da Computação

Marcos Castilho

DInf/UFPR

4 de agosto de 2021

SAT está em $\mathcal{NP}$

Primeiramente, definimos uma representação para FBF's sobre um conjunto de variáveis booleanas $\{x_1, x_2, \ldots, x_n\}$.

- Variável: codificada pela representação em binário do seu subescrito;
- Literal: é a variável codificada seguido de #1 se o literal é positivo e #0 se negativo.

Literal	Codificação
x_i	$\bar{i}\#1$
$\neg x_i$	$\bar{i}\#0$

SAT está em $\mathcal{NP}$

- O número que segue a codificação da variável especifica o valor booleano que satisfaz o literal;
- Uma FBF é codificada concatenando-se os literais com os símbolos representando conjunção e disjunção.
- Exemplo:
  - $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$
  - $1\#1 \vee 10\#0 \wedge 1\#0 \vee 11\#1$.

SAT está em $\mathcal{NP}$

- A entrada da MT consiste da codificação das variáveis na fórmula seguida de ## e depois a codificação da fórmula.
- Exemplo para a fórmula anterior:
  - $1\#10\#11\#\#1\#1 \vee 10\#0 \wedge 1\#0 \vee 11\#1$.

SAT está em $\mathcal{NP}$

- Representação de uma instância do problema:
  - string sobre $\Sigma = \{0, 1, \wedge, \vee, \#\}$.
- A linguagem L_{SAT} consiste
  - são todas as strings sobre Σ que representam FBF's em FNC.

SAT está em $\mathcal{NP}$

A máquina de Turing não determinística com duas fitas M:

- M usa *chutar e verificar*;
- O chute gera não determinísticamente uma atribuição de valores verdade;
- M inicia com a entrada na fita 1 e BB na fita 2.

SAT está em $\mathcal{NP}$

Passo 1:

- Se a entrada não tem a forma certa, a computação termina e rejeita.

  Fita 2 BB

  Fita 1 $B1\#10\#11\#\#1\#1 \vee 10\#0 \wedge 1\#0 \vee 11\#1B.$

SAT está em $\mathcal{NP}$

Passo 2:

- A codificação da primeira variável da fita 1 é copiada na fita 2;
- Segue um 0 ou 1 gerado não deterministicamente;
- Se não for a última variável, coloca ## e repete para as outras;
- Não deterministicamente, escolher um valor para cada variável define a atribuição de valor verdade t;
- O valor verdade para x_i é denotado $t(x_i)$.

Fita 2 $B1\#t(x_1)\#\#10\#t(x_2)\#\#11\#t(x_3)B$

Fita 1 $B1\#10\#11\#\#1\#1 \vee 10\#0 \wedge 1\#0 \vee 11\#1B$.

SAT está em $\mathcal{NP}$

- Reposicionar o cabeçote da fita 2 no início;
- O cabeçote da fita 1 é posicionado após $\#\#$ em uma posição para ler a primeira variável da fórmula.

  Fita 2 $B1\#t(x_1)\#\#10\#t(x_2)\#\#11\#t(x_3)B$

  Fita 1 $B1\#10\#11\#\#1\#1 \vee 10\#0 \wedge 1\#0 \vee 11\#1B.$

SAT está em $\mathcal{NP}$

- A geração da atribuição de valores é a única parte não determinística de M;
- O restante da computação determina se a fórmula é satisfeita pela atribuição não determinística.

SAT está em $\mathcal{NP}$

Passo 3:

- Assuma que a codificação da variável x_i foi lida na fita 1;
- A codificação de x_i é encontrada na fita 2;
- M então obtém o resultado da comparação de $t(x_i)$ na fita 2 com o valor booleano que segue x_i na fita 1.

SAT está em $\mathcal{NP}$

Passo 4:

- Se os valores não casarem, o literal atual não é satisfeito;
- Se o símbolo que segue o literal não é B nem $\wedge$, todo literal nesta cláusula foi examinado e falhou;
- Quando isso ocorre, a atribuição de valores verdade não satisfaz a fórmula e a entrada é rejeitada;
- Se $\vee$ é encontrado, os cabeçotes são reposicionados para examinar o próximo literal (conforme passo 3)

SAT está em $\mathcal{NP}$

- Se os valores casam, o literal e a cláusula atual são satisfeitas;
- O cabeçote na fita 1 se move para a direita do próximo $\wedge$ ou B;
- Se um B é encontrado, a computação termina e a entrada é aceita;
- Senão, a próxima cláusula é processada (conforme passo 3);

SAT está em $\mathcal{NP}$

- O passo 3 determina a taxa de crescimento do tamanho da computação;
- No pior caso, o casamento requer comparar a variável da fita 1 com cada uma das variáveis da fita 2 para descobrir o casamento;
- Isso pode ser feito em tempo $O(kn^2)$, onde:
  - n é o número de variáveis; e
  - k é o número de literais da entrada.

Fim da prova

SAT está em $\mathcal{NP}$-difícil

- Como provar que todo problema em $\mathcal{NP}$ pode ser reduzido para SAT?
- Sequer sabemos quais são todos os problemas!!!
- E eles têm, inclusive, linguagens diferentes!

SAT está em $\mathcal{NP}$-difícil

- A ideia é modelar a MT que resolve um problema em $\mathcal{NP}$ como uma instância SAT.
- Na verdade, um computador nada mais é do que um circuito baseado em lógica proposicional!
- Basta mostrar que esta modelagem pode ser feita em tempo polinomial.

SAT está em $\mathcal{NP}$-difícil

- Seja M uma MT não determinística cujas computações são limitadas por um polinômio p.
- Assumimos que esta MT tem um único estado inicial e um único estado final.
- A transformação a seguir assume que todas as computações com tamanho de entrada n contém $p(n)$ configurações.

SAT está em $\mathcal{NP}$-difícil

Assumimos:

- $Q = \{q_0, q_1, \ldots, q_m\}$
- $\Gamma = \{B = a_0, a_1, \ldots, a_s, a_{s+1}, a_t\}$
- $\Sigma = \{a_{s+1}, a_{s+2}, \ldots, a_t\}$
- $F = \{q_m\}$
- O B é o símbolo na fita com número zero
- O estado de rejeição é q_{m-1}

SAT está em $\mathcal{NP}$-difícil

- Seja $u \in \Sigma^*$ uma palavra de tamanho n.
- Definiremos uma fórmula $f(u)$ que codifica M com entrada u.
- O tamanho de $f(u)$ depende de $p(n)$.
- A modelagem é feita para que exista uma atribuição de valores que satisfaz $f(u)$ se, e somente se, $u \in L(M)$.

SAT está em $\mathcal{NP}$-difícil

Sejam as seguintes classes de variáveis:

Variável		Interpretação
$Q_{i,k}$	$0 \leq i \leq m$ $0 \leq k \leq p(n)$	M está no estado q_i no tempo k
$P_{j,k}$	$0 \leq j \leq p(n)$ $0 \leq k \leq p(n)$	M está escaneando a posição i no tempo k
$S_{j,r,k}$	$0 \leq j \leq p(n)$ $0 \leq r \leq t$ $0 \leq k \leq p(n)$	A posição j contém o símbolo a_r no tempo k

- Seja V o conjunto de variáveis como sendo a união das três classes.

SAT está em $\mathcal{NP}$-difícil

- Uma computação de M define um assinalamento de verdade em V
- Exemplo: Se a posição 3 na fita inicialmente contém a_i, então $S_{3,i,0}$ e verdadeiro.
- Logo, $S_{3,j,0}$ deve ser falso para todo $j \neq i$.
- Um assinalamento de verdade assim obtido especifica o estado, a posição do cabeçote e os símbolos da fita em cada tempo k.

SAT está em $\mathcal{NP}$-difícil

- Problema: um assinalamento de verdade não corresponde necessariamente à uma situação válida.
- Exemplo: Se $P_{0,0}$ e $P_{1,0}$ são verdadeiros, significa que a máquina está ao mesmo tempo em duas posições distintas no tempo 0.

SAT está em $\mathcal{NP}$-difícil

- Portanto, temos que impor restrições na fórmula para impedir estas situações inválidas.
- Vamos especificar oito conjuntos de fórmulas para capturar a string u e as transições de M.

SAT está em $\mathcal{NP}$-difícil

(1) Para cada tempo k, M está em pelo menos um estado e M está em no máximo um estado

Cláusulas	Condições
$\bigvee_{i=0}^{m} Q_{i,k}$	$0 \leq k \leq p(n)$
$\neg Q_{i,k} \vee \neg Q_{i',k}$	$0 \leq i < i' \leq m$ $0 \leq k \leq p(n)$

- Note que $\neg Q_{i,k} \vee \neg Q_{i',k}$ é equivalente a $Q_{i,k} \rightarrow \neg Q_{i',k}$

SAT está em $\mathcal{NP}$-difícil

(2) Para cada tempo k, o cabeçote está em pelo menos uma posição e também o cabeçote está no máximo em uma posição.

Cláusulas	Condições
$\bigvee_{j=0}^{p(n)} P_{j,k}$	$0 \leq k \leq p(n)$
$\neg P_{j,k} \vee \neg P_{j',k}$	$0 \leq j < j' \leq p(n)$ $0 \leq k \leq p(n)$

SAT está em $\mathcal{NP}$-difícil

(3) Para cada tempo k e posição j, a posição j contém pelo menos um símbolo e também no máximo um símbolo.

Cláusulas	Condições
$\bigvee_{r=0}^{t} S_{j,r,k}$	$0 \leq j \leq p(n)$ $0 \leq k \leq p(n)$
$\neg S_{j,r,k} \vee \neg S_{j,r',k}$	$0 \leq j \leq p(n)$ $0 \leq r < r' \leq t$ $0 \leq k \leq p(n)$

SAT está em $\mathcal{NP}$-difícil

- (1) assegura que a máquina está em um único estado em cada tempo;
- (1)-(3) define uma configuração de M para cada tempo entre 0 e $p(n)$;
- (1)-(2) asseguram que a máquina está lendo uma única posição em um único estado em cada tempo;
- (3) assegura que a fita está bem definida, isto é, a fita contém exatamente um símbolo em cada posição que pode ser referenciada na computação.

SAT está em $\mathcal{NP}$-difícil

(4) Condições iniciais para a entrada $u = a_{r_1} a_{r_2} \ldots a_{r_n}$. A computação começa lendo o branco mais à esquerda e a string u é a entrada. Também representamos a posição no tempo 0, e o fato de que o restante da fita é branco no tempo 0.

Cláusulas	Condições
$Q_{0,0}$	
$P_{0,0}$	
$S_{0,0,0}$	
$S_{1,r_1,0}$	
$S_{2,r_2,0}$	

SAT está em $\mathcal{NP}$-difícil

(5) A condição de aceitação: o estado de parada é q_m.

Cláusulas	Condições
$Q_{m,p(n)}$	

SAT está em $\mathcal{NP}$-difícil

- Uma computação não consiste de configurações não relacionadas.
- Cada configuração deve ser obtida a partir da aplicação de uma transição.
- Assuma que M está no estado q_i, lendo a_r na posição j no tempo k.
- As fórmulas a seguir geram a configuração permitida no tempo $k+1$ baseada nas variáveis que definem a configuração no tempo k.

SAT está em $\mathcal{NP}$-difícil

(6) Consistência da fita. Símbolos que não estão na posição do cabeçote permanecem inalterados.

Cláusulas	Condições
$\neg S_{j,r,k} \vee P_{j,k} \vee S_{j,r,k+1}$	$0 \leq j \leq p(n)$
	$0 \leq r \leq t$
	$0 \leq k \leq p(n)$

- Esta cláusula não é satisfeita se uma mudança ocorre na fita em posição diferente daquela que está sendo lida pelo cabeçote.
- Basta notar que a fórmula é equivalente a
- $\neg P_{j,k} \rightarrow (S_{j,r,k} \rightarrow S_{j,r,k+1})$

SAT está em $\mathcal{NP}$-difícil

- Assuma que, para um dado tempo k, M está em q_i lendo a_r na posição j.
- Isto é modelado atribuindo-se 1 para $Q_{i,k}$, $P_{j,k}$ e $S_{j,r,k}$.

SAT está em $\mathcal{NP}$-difícil

- Então:

$$\begin{aligned}
&\text{(a)} \quad \neg Q_{i,k} \vee \neg P_{j,k} \vee \neg S_{j,r,k} \vee Q_{i',k+1} \\
&\text{(b)} \quad \neg Q_{i,k} \vee \neg P_{j,k} \vee \neg S_{j,r,k} \vee S_{j,r',k+1} \\
&\text{(c)} \quad \neg Q_{i,k} \vee \neg P_{j,k} \vee \neg S_{j,r,k} \vee P_{j+n(d),k+1}
\end{aligned}$$

- onde $n(L) = -1$ e $n(R) = 1$
- A conjunção de (a)-(c) é satisfeita apenas se a configuração em $k+1$ for obtida a partir da configuração no tempo k por uma aplicação da transição $[q_i', a_{r'}, d] \in \delta(q_i, a_r)$

SAT está em $\mathcal{NP}$-difícil

- A representação clausal das transições é usada para construir a fórmula cuja satisfação garanta que as variáveis que definem a configuração no tempo $k+1$ sejam obtidas das variáveis que definem a configuração no tempo k por uma aplicação de uma transição de M.

SAT está em $\mathcal{NP}$-difícil

- A menos dos estados q_m e q_{m-1}, as restrições em M asseguram que pelo menos uma transição é definida para cada par de símbolo e estado.
- Construímos então uma FBF em FNC para cada tempo, estado não terminal, posição do cabeçote e símbolo da fita:
  - $(\neg Q_{i,j} \vee \neg P_{j,k} \vee \neg S_{j,r,k} \vee Q_{i',k+1})$ (novo estado) $\wedge$
  - $(\neg Q_{i,j} \vee \neg P_{j,k} \vee \neg S_{j,r,k} \vee P_{j+n(d),k+1})$ (nova posição do cabeçote) $\wedge$
  - $(\neg Q_{i,j} \vee \neg P_{j,k} \vee \neg S_{j,r,k} \vee S_{j,r',k+1})$ (novo símbolo na posição r).
- onde $[q_{i'}, a_{r'}, d] \in \delta(q_i, a_r)$, menos se a posição for 0 e a direção especificada for L.

SAT está em $\mathcal{NP}$-difícil

Para evitar este caso, definimos uma transição para um estado de rejeição:

- $(\neg Q_{i,kj} \vee \neg P_{0,k} \vee \neg S_{0,r,k} \vee Q_{m-1,k+1})$ (entrou no estado de rejeição) $\wedge$
- $(\neg Q_{i,j} \vee \neg P_{0,k} \vee \neg S_{0,r,k} \vee P_{0,k+1})$ (mesma posição do cabeçote) $\wedge$
- $(\neg Q_{i,j} \vee \neg P_{0,k} \vee \neg S_{0,r,k} \vee S_{0,r,k+1})$ (mesmo símbolo na posição r).

Isso para todas as transições $[q_{i'}, a_{r'}, L] \in \delta(q_i, a_r)$.

SAT está em $\mathcal{NP}$-difícil

- Como M é não determinística, podem existir algumas transições que podem ser aplicadas para uma configuração.
- Qualquer uma destas transições é permitida na computação.

SAT está em $\mathcal{NP}$-difícil

- Seja $trans(i, j, r, k)$ a disjunção da FNC das fórmulas que representam as transições alternativas para uma configuração de M que está
  - no tempo k
  - no estado q_i
  - com cabeçote na posição j
  - lendo o símbolo r.
- A fórmula acima é satisfeita apenas se os valores das variáveis que codificam a configuração no tempo $k + 1$ representam um sucessor legítimo das variáveis que codificam a configuração no tempo k.

SAT está em $\mathcal{NP}$-difícil

(7) Geração das configurações sucessoras.

Cláusulas
$trans(i, j, r, k)$

SAT está em $\mathcal{NP}$-difícil

- Falta apenas especificar as fórmulas que representam as transições quando M está em q_m e q_{m-1}.
- Que são os estados de aceitação e rejeição de M.

SAT está em $\mathcal{NP}$-difícil

(8) Parada da computação.

Cláusulas	Interpretação
$\neg Q_{i,k} \vee \neg P_{j,k} \vee \neg S_{j,r,k} \vee Q_{i,k+1}$	mesmo estado
$\neg Q_{i,k} \vee \neg P_{j,k} \vee \neg S_{j,r,k} \vee P_{j,k+1}$	mesma posição do cabeçote
$\neg Q_{i,k} \vee \neg P_{j,k} \vee \neg S_{j,r,k} \vee S_{j,r,k+1}$	mesmo símbolo na posição r

- Estas cláusulas são construídas para todos os j, r, k, nas faixas apropriadas, e $i = q_{m-1}, q_m$.

SAT está em $\mathcal{NP}$-difícil

- Seja $f'(u)$ a conjunção das fórmulas construídas nos passos de (1) a (8).
- Quando $f'(u)$ é satisfeita por uma atribuição de valores para V, as variáveis definem a configuração da computação de M que aceita u.
- As cláusulas (4) especificam que a configuração no tempo 0 é o estado inicial da computação de M com entrada u.
- Cada configuração subsequente é obtida a partir do seu sucessor pelo resultado da aplicação de uma transição.
- u é aceito por M pois a condição (5) indica que a configuração final contém q_m.

SAT está em $\mathcal{NP}$-difícil

- As únicas fórmulas que não estão em FNC são as $trans(i, j, r, k)$.
- Mas sabemos que elas podem ser transformadas para FNC.
- Só resta mostrar que tudo isso pode ser feito em tempo polinomial.

SAT está em $\mathcal{NP}$-difícil

- A transformação de u para $f(u)$ consiste da construção das cláusulas e a conversão de *trans* para FNC.
- O número de cláusulas é uma função de:
  - O número de estados m e o número de símbolos t na fita;
  - O tamanho n da entrada u;
  - O limite $p(n)$ no tamanho da computação de M.

SAT está em $\mathcal{NP}$-difícil

- Os valores m e t obtidos de M são independentes da entrada;
- Para a faixa de subescritos, vemos que o número de cláusulas é polinomial em $p(n)$;
- A obtenção de $f(u)$ é completada com a conversão de *trans* em FNC.
- Mas isso é garantido por resultados da lógica proposicional.

SAT está em $\mathcal{NP}$-difícil

- Basta representar uma fórmula que serve como entrada para M resolver SAT;
- Por exemplo, a técnica usada para mostrar que SAT está em $\mathcal{NP}$ serve.
- Converter a representação em alto nível para a representação da máquina pode ser feito em tempo polinomial.

Licença

- Slides feitos em LaTeX usando beamer e tikz, editados com vim.
- Licença