Teoria da Complexidade

- Há muitos problemas decidíveis, mas na prática eles podem precisar de muito tempo para serem resolvidos

Ex: Coloração de grafos, Fatorização de inteiros

- Aqui teremos problemas tratáveis e intratáveis

Complexidade de Tempo

Def: Uma M.T. é dita polinomialmente limitada se existe uma função polinomial $p(n)$ t.q. para toda entrada $w \in \Sigma_0^*$, qualquer computação a partir do estado inicial sempre alcança um estado final em no máximo $p(|w|)$ passos.

Def: A classe P é a família das linguagens que podem ser decididas por uma M.T. determinística e polinomialmente limitada

- Em geral, dizemos que um problema é tratável se está em P.

Def: A classe NP é a família das linguagens que podem ser decididas por uma M.T. não-determinística e polinomialmente limitada

Ex: As codificações de números compostos

Teorema. $P \subseteq NP$.

Prova: Toda máquina determinística é não-determinística.

DEF: Uma M.T. é dita exponencialmente limitada se existe uma constante c e uma função polinomial $p(n)$ t.q. para toda entrada $w \in \Sigma_0^*$, qualquer computação a partir do estado inicial sempre alcança um estado final em no máximo $c^{P(|w|)}$ passos.

DEF: A classe EXPTIME é a família das linguagens que podem ser decididas por uma M.T. determinística e exponencialmente limitada

DEF: A classe NEXPTIME é a família das linguagens que podem ser decididas por uma M.T. não-determinística e exponencialmente limitada

Teorema: EXPTIME $\subseteq$ NEXPTIME

Teorema: NP $\subseteq$ EXPTIME

Prova: Basta notar que a máquina de Turing determinística que construímos para simular uma M.T. não-determinística usa um número exponencial de passos.

$$P \subseteq NP \subseteq EXPTIME \subseteq NEXPTIME$$

COMPLEXIDADE DE ESPAÇO

DEF: UMA M.T. É DITA ESPACIALMENTE POLINOMIALMENTE LIMITADA SE HÁ UM POLINÔMIO $P(n)$ T.Q. PARA TODA ENTRADA $w \in \Sigma_0^*$, QUALQUER COMPUTAÇÃO A PARTIR DO ESTADO INICIAL DA MÁQUINA SEMPRE ALCANÇA UM ESTADO DE PARADA TENDO VISITADO NO MÁXIMO $p(|w|)$ CASAS DISTINTAS DA FITA.

DEF: A CLASSE PSPACE É A FAMÍLIA DE LINGUAGENS QUE PODEM SER DECIDIDAS POR UMA M.T. DETERMINÍSTICA ESPACIALMENTE POLINOMIALMENTE LIMITADO

DEF: A CLASSE NPSPACE É A FAMÍLIA DE LINGUAGENS QUE PODEM SER DECIDIDAS POR UMA M.T NÃO-DETERMINÍSTICA ESPACIALMENTE POLINOMIALMENTE LIMITADO

OBS: NÃO SABEMOS SE $P = NP$ OU $EXPTIME = NEXPTIME$

TEOREMA DE SAVITCH: $PSPACE = NPSPACE$

TEOREMA: $P \subseteq PSPACE$

PROVA: EM CADA PASSO DA COMPUTAÇÃO A MÁQUINA PODE VISITAR NO MÁXIMO UMA NOVA CASA.

- COM O MESMO ARGUMENTO OBTEMOS $NP \subseteq NPSPACE$, O QUE NOS DÁ

TEOREMA: $NP \subseteq PSPACE$

TEOREMA: $PSPACE \subseteq EXPTIME$

PROVA: SE USAMOS $p(|w|)$ CASAS, ENTÃO HÁ NO MÁXIMO $|Q| \cdot f(|w|) \cdot m^{f(|w|)}$ CONFIGURAÇÕES POSSÍVEIS, QUE LIMITA O NÚMERO DE PASSOS DA COMPUTAÇÃO.

Co-classes

Def: Se C é uma classe de complexidade, definimos Co-C como a família de linguagens cujo complemento está em C

$$\text{Co-}C = \{ L : \bar{L} \in C \}$$

Temos

1) $P = \text{Co-}P$
2) $PSPACE = \text{Co-}PSPACE = NPSPACE = \text{Co-}NPSPACE$
3) $EXPTIME = \text{Co-}EXPTIME$

⇝ basta inverter os estados de aceitação e rejeição

Perguntas: $\text{Co-}NP = NP$? $\text{Co-}NEXPTIME = NEXPTIME$?

Lema: Se $C_1 \subseteq C_2$, então $\text{Co-}C_1 \subseteq \text{Co-}C_2$

Prova: Se $L \in \text{Co-}C_1$, então $\bar{L} \in C_1$, logo $\bar{L} \in C_2$, e portanto $L \in \text{Co-}C_2$.

⇝ Isso implica que $\text{Co-}P \subseteq \text{Co-}NP$

$$\begin{tikzcd}[column sep=small]
 & NP \arrow[dr, phantom, "\subseteq"] & & & NEXPTIME \\
P \arrow[ur, phantom, "\subseteq"] \arrow[dr, phantom, "\subseteq"] & & PSPACE \subseteq EXPTIME \arrow[urr, phantom, "\subseteq"] \arrow[drr, phantom, "\subseteq"] & & \\
 & \text{Co-}NP \arrow[ur, phantom, "\subseteq"] & & & \text{Co-}NEXPTIME
\end{tikzcd}$$

REDUÇÕES POLINOMIAIS

DEF: DIZEMOS QUE UMA FUNÇÃO É POLINOMIALMENTE COMPUTÁVEL SE EXISTE M.T. DETERMINÍSTICA POL. LIMITADA QUE A COMPUTA.

DEF: DADAS $L_1, L_2 \subseteq \Sigma_{10}^*$, DIZEMOS QUE EXISTE UMA REDUÇÃO POL. DE L_1 PARA L_2, DENOTADA POR $L_1 \leq_p L_2$, SE EXISTE UMA FUNÇÃO POL. COMPUTÁVEL $f: \Sigma_{10}^* \to \Sigma_{10}^*$ T.Q. $w \in L_1$ SE E SÓ SE $f(w) \in L_2$.

PROP: SEJA C UMA CLASSE, $L_2 \in C$, E SEJA L_1 UMA LINGUAGEM T.Q. $L_1 \leq_p L_2$. ENTÃO $L_1 \in C$.

PROVA: SEJA $M_{1,2}$ UMA MÁQUINA QUE COMPUTA A REDUÇÃO $L_1 \leq_p L_2$.
SEJA M A MÁQUINA QUE DECIDE L_2 NO TEMPO/ESPAÇO DA CLASSE C.
ENTÃO $M_{1,2} \to M$ DECIDE L_1 NO TEMPO/ESPAÇO DA CLASSE C.

DEF: SEJA C UMA CLASSE DE COMPLEXIDADE E SEJA L UMA LINGUAGEM. DIZEMOS QUE L É C-DIFÍCIL SE PARA TODA $L' \in C$, EXISTE $L' \leq_p L$.

DEF: SEJA C UMA CLASSE DE COMPLEXIDADE E SEJA L UMA LINGUAGEM. DIZEMOS QUE L É C-COMPLETA SE

1) $L \in C$; E
2) L É C-DIFÍCIL

LEMA: SE L É C-COMPLETA E $L \leq_p L'$, ENTÃO L' É C-COMPLETA

TEOREMA DE COOK-LEVIN: SAT É NP-COMPLETA