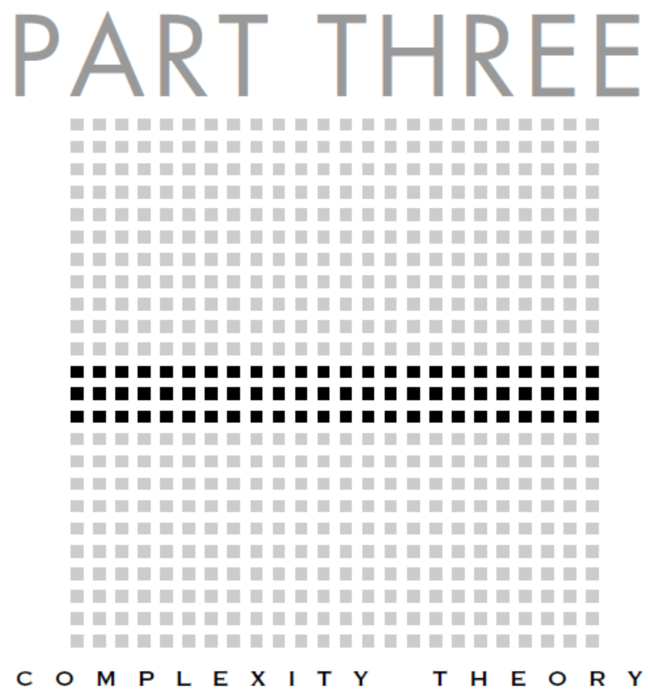


TEORIA DELLA COMPLESSITA’

Linguaggi NP-completi: *SUBSET-SUM e HAMPATH*

30 maggio 2023

Vogliamo definire quando un linguaggio B è uno dei linguaggi «più difficili» della classe NP.

Abbiamo visto un modo per definire quando B è «più difficile» di A, ovvero quando A è di difficoltà «minore o uguale» a B:

$$A \leq_p B$$

Quindi B è uno dei linguaggi «più difficili» della classe NP…..

Definizione

Un linguaggio B è *NP-completo* se soddisfa le seguenti due condizioni:

1. B appartiene a NP
2. Per ogni linguaggio A in NP, $A \leq_p B$ (ovvero B è NP-hard)

Provare la *NP* – completezza

Una possibile strategia per provare che un linguaggio C è NP-completo:

1. Mostrare che $C \in NP$
2. Scegliere un linguaggio B che sia NP-completo
3. Definire una riduzione di tempo polinomiale di B in C.

Proveremo che alcuni linguaggi sono NP-completi mostrando una riduzione di tempo polinomiale da 3SAT che utilizza la tecnica di ‘’riduzione mediante progettazione di componenti" o ‘’gadgets".

Occorre prima dimostrare che 3SAT è NP-completo.

Problemi *NP* – completi

- SAT (Cook-Levin)
- SAT_{CNF} (senza dimostrazione)
- 3SAT (cenni)
- CLIQUE (da 3SAT)
- CLIQUE (da 3SAT coi gadget)
- VERTEX-COVER (da 3SAT coi gadget)
- SUBSET-SUM (da 3SAT)
- HAMPATH (da 3SAT coi gadget)
- UHAMPATH (da HAMPATH)

SUBSET-SUM

SUBSET-SUM: Dato un insieme finito S di numeri interi e un numero intero t, esiste un sottoinsieme S' di S tale che la somma dei suoi numeri sia uguale a t?

$SUBSET\text{-}SUM = \{\langle S, t\rangle \mid S = \{x_1, \ldots, x_k\}$ ed esiste $S' \subseteq S$ tale che $\sum_{s \in S'} s = t\}$

Esempio: $\langle \{4, 11, 16, 21, 27\}, 25\rangle \in SUBSET\text{-}SUM$ perché $4 + 21 = 25$.

$3SAT \leq_p$ SUBSET-SUM

- Sia ϕ una formula $3CNF$ con variabili $x_1, \ldots x_\ell$ e clausole $c_1, \ldots, c_k$.
- Associamo a ϕ un insieme S di numeri e un numero t tali che ϕ è soddisfacibile se e solo se $\langle S, t \rangle \in SUBSET\text{-}SUM$. I numeri in S e il numero t sono espressi nella notazione decimale ordinaria.
- Inoltre $\langle S, t \rangle$ può essere costruita in tempo polinomiale nella lunghezza di $\langle \phi \rangle$.

3SAT $\leq_p$ SUBSET-SUM: Esempio

$$\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

Variabili: $x_1, x_2, x_3, \ell = 3$

Clausole: C_1, C_2, C_3, C_4

$k = 4$

			Variabili x_1	Variabili x_2	Variabili x_3	Clausole C_1	Clausole C_2	Clausole C_3	Clausole C_4	
		Numero	1	2	3	1	2	3	4	
Variabili	x_1	y_1	1	0	0	0	1	1	0	x_1
	$\sim x_1$	z_1	1	0	0	1	0	0	1	$\sim x_1$
	x_2	y_2	0	1	0	1	1	0	0	x_2
	$\sim x_2$	z_2	0	1	0	0	0	1	1	$\sim x_2$
	x_3	y_3	0	0	1	0	1	1	0	x_3
	$\sim x_3$	z_3	0	0	1	1	0	0	1	$\sim x_3$
Clausole	C_1	g_1	0	0	0	1	0	0	0	
		h_1	0	0	0	1	0	0	0	
	C_2	g_2	0	0	0	0	1	0	0	
		h_2	0	0	0	0	1	0	0	
	C_3	g_3	0	0	0	0	0	1	0	
		h_3	0	0	0	0	0	1	0	
	C_4	g_4	0	0	0	0	0	0	1	
		h_4	0	0	0	0	0	0	1	
		t	1	1	1	3	3	3	3	

$S = \{y_1, z_1, y_2, z_2, y_3, z_3, g_1, h_1, g_2, h_2, g_3, h_3, g_4, h_4\}$

$t = 1113333$

$3SAT \leq_p SUBSET\text{-}SUM$: Esempio

$$\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

Variabili: $x_1, x_2, x_3, \ell = 3$

Clausole: C_1, C_2, C_3, C_4

$k = 4$

		Variabili			Clausole				
		x_1	x_2	x_3	C_1	C_2	C_3	C_4	
	Numero	1	2	3	1	2	3	4	
Variabili: x_1	y_1	1	0	0	0	1	1	0	x_1
$\sim x_1$	z_1	1	0	0	1	0	0	1	$\sim x_1$
x_2	y_2	0	1	0	1	1	0	0	x_2
$\sim x_2$	z_2	0	1	0	0	0	1	1	$\sim x_2$
x_3	y_3	0	0	1	0	1	1	0	x_3
$\sim x_3$	z_3	0	0	1	1	0	0	1	$\sim x_3$
Clausole: C_1	g_1	0	0	0	1	0	0	0	
	h_1	0	0	0	1	0	0	0	
C_2	g_2	0	0	0	0	1	0	0	
	h_2	0	0	0	0	1	0	0	
C_3	g_3	0	0	0	0	0	1	0	
	h_3	0	0	0	0	0	1	0	
C_4	g_4	0	0	0	0	0	0	1	
	h_4	0	0	0	0	0	0	1	
	t	1	1	1	3	3	3	3	

$$S = \{y_1, z_1, y_2, z_2, y_3, z_3, g_1, h_1, g_2, h_2, g_3, h_3, g_4, h_4\}$$
$$t = 1113333$$

$3SAT \leq_p$ SUBSET-SUM: Esempio

$$\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

		Variabili			Clausole				
		x_1	x_2	x_3	C_1	C_2	C_3	C_4	
	Numero	1	2	3	1	2	3	4	
Variabili x_1	y_1	1	0	0	0	1	1	0	x_1
$\sim x_1$	z_1	1	0	0	1	0	0	1	$\sim x_1$
x_2	y_2	0	1	0	1	1	0	0	x_2
$\sim x_2$	z_2	0	1	0	0	0	1	1	$\sim x_2$
x_3	y_3	0	0	1	0	1	1	0	x_3
$\sim x_3$	z_3	0	0	1	1	0	0	1	$\sim x_3$
Clausole C_1	g_1	0	0	0	1	0	0	0	
	h_1	0	0	0	1	0	0	0	
C_2	g_2	0	0	0	0	1	0	0	
	h_2	0	0	0	0	1	0	0	
C_3	g_3	0	0	0	0	0	1	0	
	h_3	0	0	0	0	0	1	0	
C_4	g_4	0	0	0	0	0	0	1	
	h_4	0	0	0	0	0	0	1	
	t	1	1	1	3	3	3	3	

$S = \{y_1, z_1, y_2, z_2, y_3, z_3, g_1, h_1, g_2, h_2, g_3, h_3, g_4, h_4\}$

$t = 1113333$

$3SAT \leq_p$ SUBSET-SUM: Esempio

$$\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

Assegnamento di verità alle variabili:

$x_1 = 0$
$x_2 = 1$
$x_3 = 1$

$$\phi = \underbrace{(\overline{x_1} \vee x_2 \vee \overline{x_3})}_{C_1} \wedge \underbrace{(x_1 \vee x_2 \vee x_3)}_{C_2} \wedge \underbrace{(x_1 \vee \overline{x_2} \vee x_3)}_{C_3} \wedge \underbrace{(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})}_{C_4}$$

$x_1 = 0$ allora si seleziona z_1
$x_2 = 1$ allora si seleziona y_2
$x_3 = 1$ allora si seleziona y_3

$3SAT \leq_p SUBSET\text{-}SUM$: Esempio

$$\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

	x_1	x_2	x_3	C_1	C_2	C_3	C_4	
Numero	1	2	3	1	2	3	4	
y_1	1	0	0	0	1	1	0	x_1
z_1	1	0	0	1	0	0	1	$\sim x_1$
y_2	0	1	0	1	1	0	0	x_2
z_2	0	1	0	0	0	1	1	$\sim x_2$
y_3	0	0	1	0	1	1	0	x_3
z_3	0	0	1	1	0	0	1	$\sim x_3$
g_1	0	0	0	1	0	0	0	
h_1	0	0	0	1	0	0	0	
g_2	0	0	0	0	1	0	0	
h_2	0	0	0	0	1	0	0	
g_3	0	0	0	0	0	1	0	
h_3	0	0	0	0	0	1	0	
g_4	0	0	0	0	0	0	1	
h_4	0	0	0	0	0	0	1	
t	1	1	1	3	3	3	3	

Variabili

Clausole

$S = \{y_1, z_1, y_2, z_2, y_3, z_3, g_1, h_1, g_2, h_2, g_3, h_3, g_4, h_4\}$

$t = 1113333$

Assegnamento

$x_1 = 0$ seleziono z_1

$x_2 = 1$ seleziono y_2

$x_3 = 1$ seleziono y_3

$3SAT \leq_p$ SUBSET-SUM

- Sia ϕ soddisfacibile e sia τ un assegnamento che soddisfa ϕ. Consideriamo il sottoinsieme S' di S che contiene y_i se τ assegna a x_i valore 1, z_i altrimenti.
- Se sommiamo ciò che abbiamo scelto fino ad ora, otteniamo un 1 in ciascuna delle prime ℓ cifre perché abbiamo selezionato y_i o z_i per ciascun i.
- Inoltre, ciascuna delle ultime k cifre è un numero da 1 a 3 perché ciascuna clausola è soddisfatta e quindi contiene da 1 a 3 letterali veri.
- Quindi scegliamo un numero sufficiente di g_j, h_j da aggiungere a S' per portar ciascuna delle ultime k cifre fino a 3 e ottenere $\sum_{s \in S'} s = t$.

- Viceversa supponiamo che esista un sottoinsieme S' di S tale che $\sum_{s \in S'} s = t$.

  Due osservazioni:

- Tutte le cifre negli elementi di S sono 0 o 1.
- Ciascuna colonna nella tabella che descrive S contiene al più cinque 1.
- Quindi, sommando elementi di un sottoinsieme di S non si verifica mai un "riporto" nella colonna successiva.

$3SAT \leq_p$ SUBSET-SUM: Esempio

$$\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

		ℓ Variabili			k Clausole			
	Numero	1	2	3	1	2	3	4
Variabili	y_1	1	0	0	0	1	1	0
	z_1	1	0	0	1	0	0	1
	y_2	0	1	0	1	1	0	0
	z_2	0	1	0	0	0	1	1
	y_3	0	0	1	0	1	1	0
	z_3	0	0	1	1	0	0	1
Clausole	g_1	0	0	0	1	0	0	0
	h_1	0	0	0	1	0	0	0
	g_2	0	0	0	0	1	0	0
	h_2	0	0	0	0	1	0	0
	g_3	0	0	0	0	0	1	0
	h_3	0	0	0	0	0	1	0
	g_4	0	0	0	0	0	0	1
	h_4	0	0	0	0	0	0	1
	t	1	1	1	3	3	3	3

$3SAT \leq_p$ SUBSET-SUM

- Sia S' un sottoinsieme di S tale che $\sum_{s \in S'} s = t$.
- Per ogni i, $1 \leq i \leq \ell$, S' deve contenere y_i o z_i ma non entrambi.
- Sia τ l'assegnamento definito come segue: per ogni i, $1 \leq i \leq \ell$, assegniamo a x_i valore 1 se S' contiene y_i, valore 0 se S' contiene z_i.
- Questo assegnamento τ soddisfa ϕ.

- Infatti, poiché le ultime k cifre di t sono uguali a 3, in ciascuna delle k colonne finali, la somma è sempre 3.
- Per ogni j, con $1 \leq j \leq k$, almeno un 1 nella colonna c_j deve venire da qualche y_i o z_i nel sottoinsieme S' perché da g_j ed h_j può venire al più 2.

- Per ogni j nella colonna c_j vi deve essere una cifra uguale a 1 corrispondente a un y_i o z_i in S'.
- Se è y_i, allora x_i è presente in c_j e gli viene assegnato 1, quindi c_j è soddisfatta.
- Se è z_i, allora $\overline{x_i}$ è presente in c_j e a x_i viene assegnato 0, quindi c_j è soddisfatta.
- Pertanto ϕ è soddisfatta.
- Infine, la riduzione può essere effettuata in tempo polinomiale. $\square$

HAMPATH

Un cammino Hamiltoniano in un grafo orientato è un cammino (orientato) che passa per ogni vertice del grafo una e una sola volta.

Consideriamo il problema di stabilire se un grafo orientato contiene un cammino Hamiltoniano che collega due nodi specificati.

Questo si può formulare come un problema di decisione, a cui corrisponde un linguaggio associato, il linguaggio *HAMPATH*.

$$HAMPATH = \{\langle G, s, t\rangle \mid G \text{ è un grafo orientato}$$
$$\text{e ha un cammino Hamiltoniano da } s \text{ a } t\}$$

HAMPATH è NP-completo

Teorema
HAMPATH è NP-completo.

Dimostrazione.
Abbiamo già provato che *HAMPATH* è in *NP*.
Per concludere la prova, basta provare che $3SAT \leq_P HAMPATH$.

In realtà la dimostrazione è parecchio complicata.
Quest'anno:

non sarà in programma!

UHAMPATH

È possibile definire una "versione non orientata" del problema del cammino Hamiltoniano.

- Un cammino Hamiltoniano in un grafo non orientato è un cammino che passa per ogni vertice del grafo una e una sola volta.

$$UHAMPATH = \{\langle G, s, t\rangle \mid G \text{ è un grafo non orientato e ha un cammino Hamiltoniano da } s \text{ a } t\}$$

Per mostrare che *UHAMPATH* è *NP*-completo, definiamo una riduzione di tempo polinomiale da *HAMPATH* a *UHAMPATH*.

Teorema

$UHAMPATH \in NP$

Dimostrazione.

Un algoritmo N che verifica $UHAMPATH$ in tempo polinomiale:
$N =$ "Sull'input $\langle\langle G, s, t\rangle, c\rangle$, dove $G = (V, E)$ è un grafo non orientato:

1. Verifica se $c = (u_1, \ldots, u_{|V|})$ è una sequenza di $|V|$ vertici di G, altrimenti rifiuta.
2. Verifica se i nodi della sequenza sono distinti, $u_1 = s$, $u_{|V|} = t$ e, per ogni i con $2 \leq i \leq n$, se $(u_{i-1}, u_i) \in E$, accetta in caso affermativo; altrimenti rifiuta."

$\exists c : \langle\langle G, s, t\rangle, c\rangle \in L(N)$ se e solo se $\langle G, s, t\rangle \in UHAMPATH$. $\square$

UHAMPATH è *NP*-completo

Teorema

UHAMPATH è NP-completo.

Dimostrazione

Abbiamo provato che *UHAMPATH* è in *NP*.

Per concludere la prova, dimostriamo che $HAMPATH \leq_P UHAMPATH$.

HAMPATH si riduce in tempo polinomiale a *UHAMPATH*

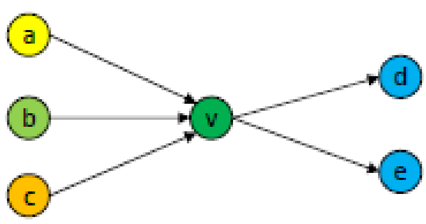
- La riduzione di tempo polinomiale associa a un grafo orientato $G = (V, E)$ con vertici s e t un grafo non orientato $G' = (V', E')$ con vertici s' e t'.
- Il grafo G ha un cammino Hamiltoniano da s a t se e solo se G' ha un cammino Hamiltoniano da s' a t'.
- Inoltre G' può essere costruito a partire da G in tempo polinomiale.

cammino Hamiltoniano in un grafo non orientato

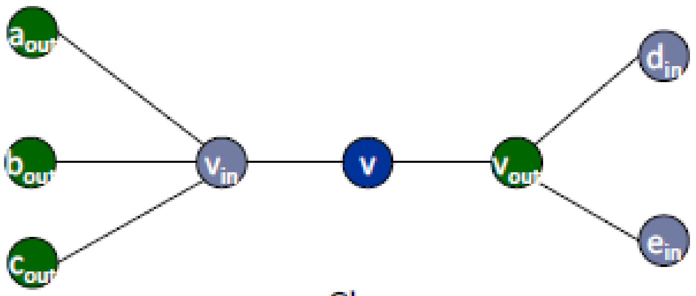
Dato grafo non orientato G' = (V', E') e due vertici s', t', esiste un cammino Hamiltoniano in G' da s' a t'?

Fatto. HAMPATH $\leq_P$ UHAMPATH.

Dim. Dato un grafo orientato G = (V, E) con n vertici, costruiamo un grafo non orientato G' con 3(n -2) + 2 vertici.



G



G'

(autore slide: Kevin Wayne)

$HAMPATH \leq_p UHAMPATH$

Costruzione di G':

- Ogni vertice u di G, diverso da s e t è rimpiazzato da tre vertici u^{in}, u^{mid} e u^{out} in G'.
- I vertici s e t sono sostituiti con i vertici s^{out} e t^{in} in G'.
- Per ogni $u \in V \setminus \{s, t\}$, (u^{in}, u^{mid}) e (u^{mid}, u^{out}) sono in E'.
- Se $(u, v) \in E$ allora $(u^{out}, v^{in}) \in E'$.

$HAMPATH \leq_p UHAMPATH$

Costruzione di G':

- Ogni vertice u di G, diverso da s e t è rimpiazzato da tre vertici u^{in}, u^{mid} e u^{out} in G'.
- I vertici s e t sono sostituiti con i vertici s^{out} e t^{in} in G'.
- Per ogni $u \in V \setminus \{s,t\}$, (u^{in}, u^{mid}) e (u^{mid}, u^{out}) sono in E'.
- Se $(u,v) \in E$ allora $(u^{out}, v^{in}) \in E'$.

Esempio: G = (V,E)
V = {s, 1, 2, t}
E = { (s,1), (1,2), (1,t), (2,1), (2,s), (2,t)}

$HAMPATH \leq_p UHAMPATH$

- Dimostriamo che G ha un cammino Hamiltoniano da s a t se e solo se G' ha un cammino Hamiltoniano da s^{out} a t^{in}.
- Se G ha un cammino Hamiltoniano P da s a t:

$$P = s, u_1, u_2, \ldots, u_k, t$$

allora P':

$$P' = s^{out}, u_1^{in}, u_1^{mid}, u_1^{out}, u_2^{in}, u_2^{mid}, u_2^{out}, \ldots, u_k^{in}, u_k^{mid}, u_k^{out}, t^{in}$$

è un cammino Hamiltoniano in G' da s^{out} a t^{in}.

HAMPATH si riduce in tempo polinomiale a *UHAMPATH*

- Viceversa se G' ha un cammino Hamiltoniano P' da s^{out} a t^{in}, è facile vedere che P' deve essere della forma

$$P' = s^{out}, u_1^{in}, u_1^{mid}, u_1^{out}, u_2^{in}, u_2^{mid}, u_2^{out}, \ldots, u_k^{in}, u_k^{mid}, u_k^{out}, t^{in}$$

- La prova è per induzione su k. Infatti P' ha come primo vertice s^{out} il quale è connesso solo a vertici della forma u_i^{in}. Quindi il secondo vertice è u_i^{in} per qualche i. I vertici successivi devono essere u_i^{mid}, u_i^{out} perché u_i^{mid} è connesso solo a u_i^{in} e u_i^{out}.
- Ma se P' ha la forma suddetta allora

$$P = s, u_1, u_2, \ldots, u_k, t$$

è un cammino Hamiltoniano da s a t.

$\square$

Linguaggi *NP* – completi

- SAT (Teorema di Cook-Levin: senza dimostrazione)
- SAT_{CNF} (senza dimostrazione)
- 3SAT (cenni)
- CLIQUE (da 3SAT coi gadget)
- VERTEX-COVER (da 3SAT coi gadget)
- SUBSET-SUM (da 3SAT coi gadget)
- HAMPATH (da 3SAT coi gadget: senza dimostrazione)
- UHAMPATH (da HAMPATH)

Teoria della complessità: argomenti trattati

- Definizione di complessità di tempo
- La complessità di tempo dipende dal modello di calcolo; useremo decisori e modelli polinomialmente equivalenti
- La complessità di tempo dipende dalla codifica utilizzata: useremo codifica in binario o polinomialmente correlata
- TIME (f(n)) = insieme dei linguaggi decisi in tempo O(f(n))
- La classe $P = \bigcup_{k \geq 0} TIME(n^k)$ e sua robustezza
- La classe EXPTIME
- Algoritmi di verifica e la classe NP
- Il concetto di riduzione polinomiale
- Il concetto di NP-completezza
- Linguaggi NP-completi

Classi di complessità

co-Turing riconoscibili

Turing riconoscibili

Decidibili

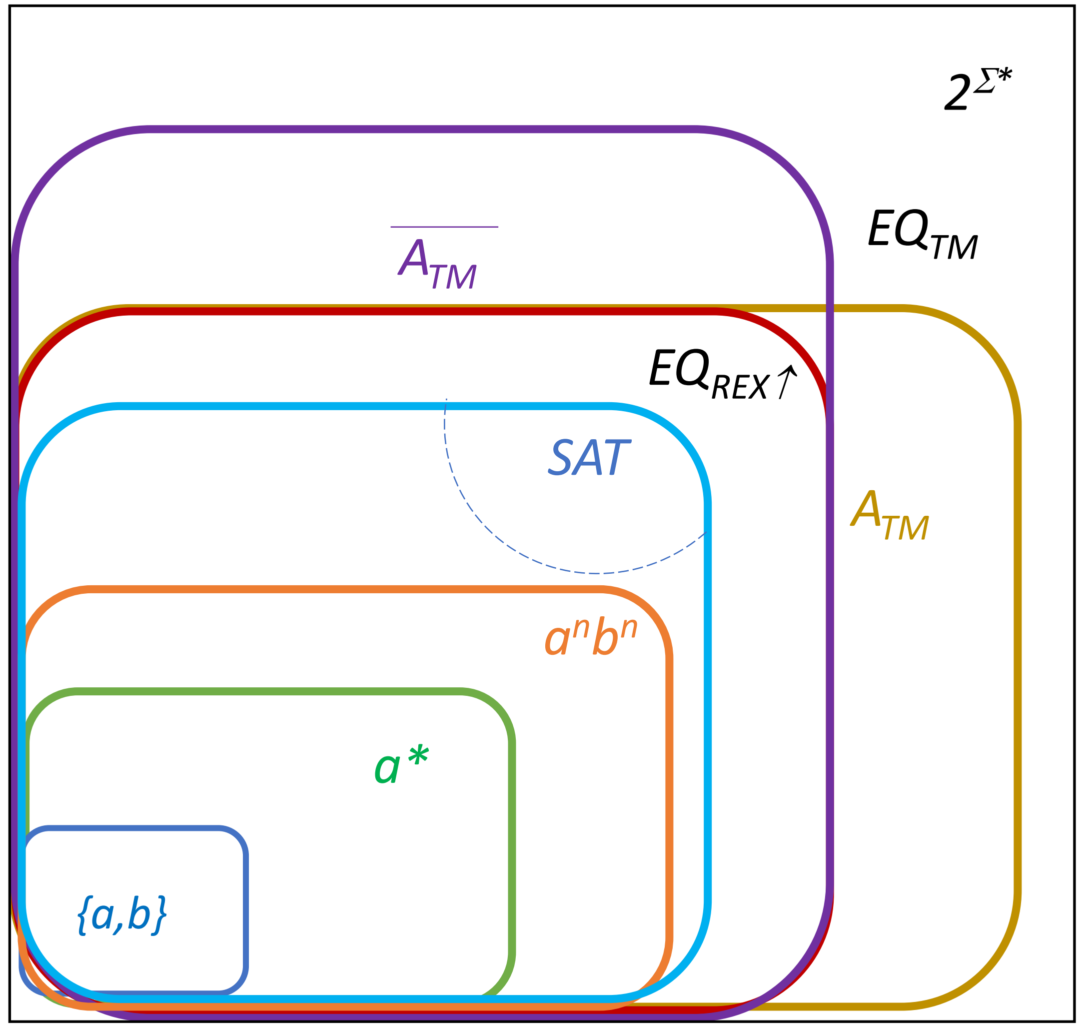
NP

NP-completi

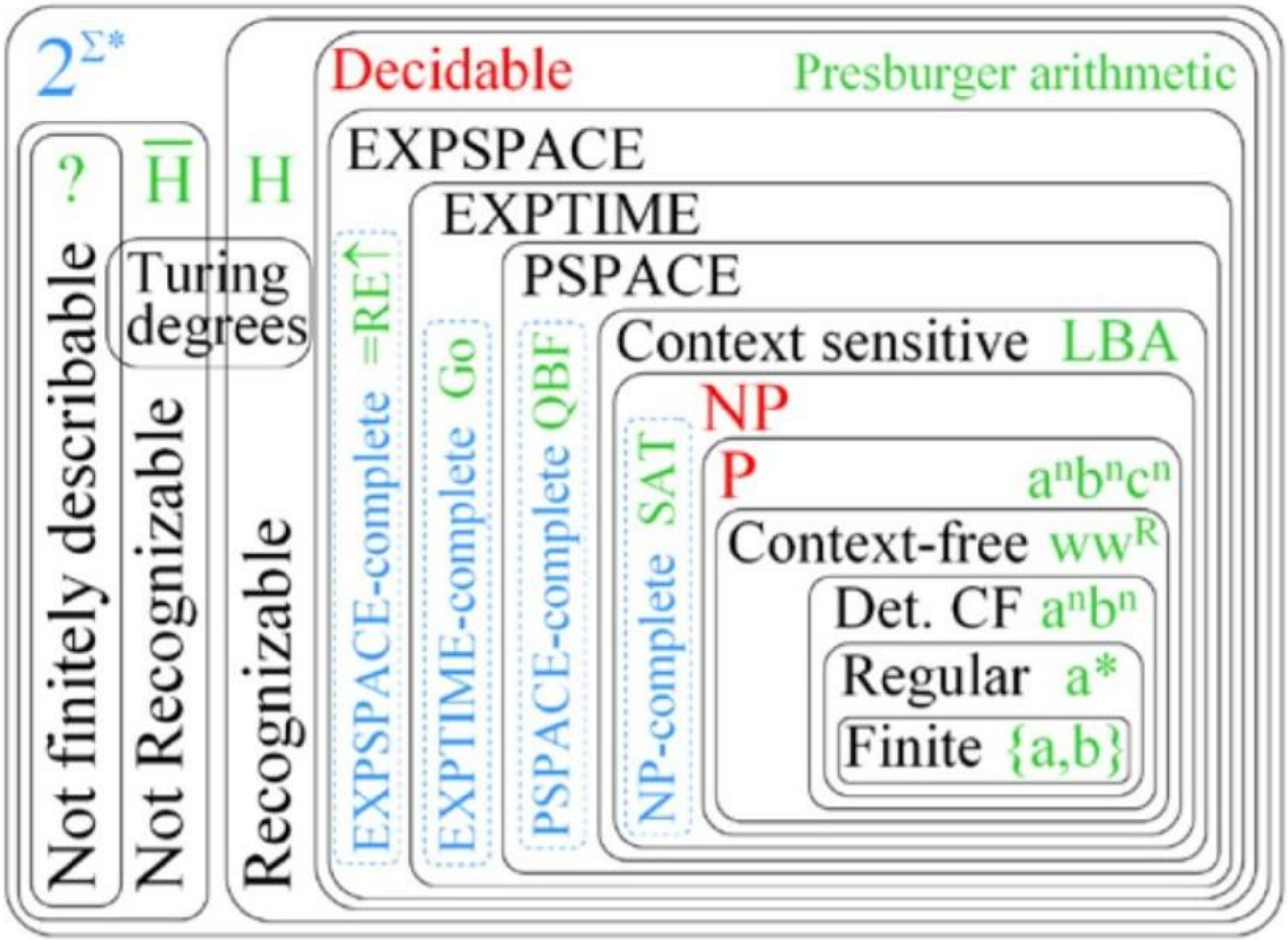
P

Regolari

Finiti



The Extended Chomsky Hierarchy



Contenuto del corso

- MODELLI DI COMPUTAZIONE:

  AUTOMI FINITI DETERMINISTICI E NON DETERMINISTICI.
  ESPRESSIONI REGOLARI. PROPRIETÀ DI CHIUSURA DEI LINGUAGGI REGOLARI. TEOREMA DI KLEENE. PUMPING LEMMA PER I LINGUAGGI REGOLARI.

  MACCHINA DI TURING DETERMINISTICA A NASTRO SINGOLO. IL LINGUAGGIO RICONOSCIUTO DA UNA MACCHINA DI TURING. VARIANTI DI MACCHINE DI TURING E LORO EQUIVALENZA.

- IL CONCETTO DI COMPUTABILITÀ: FUNZIONI CALCOLABILI, LINGUAGGI DECIDIBILI E LINGUAGGI TURING RICONOSCIBILI. LINGUAGGI DECIDIBILI E LINGUAGGI INDECIDIBILI. IL PROBLEMA DELLA FERMATA. RIDUZIONI. TEOREMA DI RICE.

- IL CONCETTO DI COMPLESSITÀ: MISURE DI COMPLESSITÀ: COMPLESSITÀ IN TEMPO DETERMINISTICO E NON DETERMINISTICO. RELAZIONI DI COMPLESSITÀ TRA VARIANTI DI MACCHINE DI TURING. LA CLASSE P. LA CLASSE NP. RIDUCIBILITÀ IN TEMPO POLINOMIALE. DEFINIZIONE DI NP-COMPLETEZZA. RIDUZIONI POLINOMIALI. ESEMPI DI LINGUAGGI NP-COMPLETI.

Fine

Esercizio (svolto)

La seguente affermazione è vera?

‘’Comunque prendo due linguaggi NP-completi A e B, si ha:

$$A \leq_p B \quad e \quad B \leq_p A\ .\text{’’}$$

Cioè, i linguaggi NP-completi hanno tutti «uguale difficoltà».

Vero o Falso? Perché? (1)

Per ognuna delle seguenti affermazioni, dire se è vera o falsa, giustificando (brevemente) la risposta citando i risultati utilizzati.
Siano A, B, C linguaggi su un alfabeto Sigma.

1. Se A ≤m B e B è decidibile, allora A è decidibile
2. Se A ≤m B e A è decidibile, allora B è decidibile
3. Se A ≤m B e B ≤m C allora C è indecidibile
4. 3SAT è indecidibile (svolto)
5. A_{TM} è riconoscibile, ma non decidibile

Vero o Falso? Perché? (2)

4. Siano A, B due linguaggi. Dire se le seguenti affermazioni sono vere o false, giustificando la risposta. Occorre fornire la definizione di A_{TM}. La valutazione dipende dal livello di precisione e rigore formale della risposta.

   (a) Se $A_{TM} \leq_m A$ e $A \leq_m B$ allora B è indecidibile.

   (b) Se $B \leq_m A$ e $A \leq_m A_{TM}$ allora B è indecidibile.