ETC - Definizioni e Teoremi

Decidibilità

- **Teorema:** Il linguaggio ATM = {<M,w> | M è una MdT che accetta la parola w} non è decidibile. (Dimostrazioni Decidibilità 1)
- **Teorema:** Il linguaggio ATM = {<M,w> | M è una MdT che accetta la parola w} è Turing riconoscibile.(Dimostrazioni Decidibilità 2)
- **Definizione:** Diciamo che un linguaggio L è co-Turing riconoscibile se ¬L è Turing riconoscibile.
- **Teorema:** Se un linguaggio L è decidibile allora anche ¬L è decidibile
- **Teorema:** Un linguaggio L è decidibile se e solo se L è Turing riconoscibile e co-Turing riconoscibile. (Dimostrazioni Decidibilità 3)
- **Teorema:** ¬ATM non è Turing riconoscibile. (Dimostrazioni Decidibilità 4)

Riduzioni

- **Definizione:** Una funzione f: $\Sigma^* \to \Sigma^*$ è calcolabile se esiste una TM M tale che su ogni input w in Σ^*, M si arresta con f(w), e solo con f(w), sul suo nastro.

- **Definizione:** Un linguaggio $A \subseteq \Sigma^*$ è riducibile mediante funzione a un linguaggio $B \subseteq \Sigma^*$, e scriveremo $A \leq_m B$, se esiste una funzione calcolabile f: $\Sigma^* \to \Sigma^*$ tale che $\forall w \in \Sigma^*$, $w \in A \Leftrightarrow f(w) \in B$. La funzione f è chiamata una riduzione da A a B.

- **Teorema:** $A \leq_m B$ se e solo se $\neg A \leq_m \neg B$. (Dimostrazioni Riducibilità 1.)

- **Teorema:** Se $A \leq_m B$ e B è decidibile, allora A è decidibile. (Dimostrazioni Riducibilità 2.)

- **Teorema:** Se $A \leq_m B$ e B è Turing riconoscibile, allora A è Turing riconoscibile. (Dimostrazioni Riducibilità 3.)

- **Corollario:** Se $A \leq_m B$ e A è indecidibile, allora B è indecidibile.

- **Corollario:** Se $A \leq_m B$ e A non è Turing riconoscibile, allora B non è Turing riconoscibile.

Riduzioni, Indecidibilità e Riconoscibilità

- A_{TM} = { ⟨M,w⟩ | M è una MdT e M accetta w}
  $HALT_{TM}$ = { ⟨M,w⟩ | M è una MdT e M si arresta su w}

  Teorema: $A_{TM} \leq_m HALT_{TM}$ (Dimostrazioni Riducibilità 4.)

- **Teorema:** $HALT_{TM}$ è indecidibile.

- A_{TM} = {<M,w> | M è una TM e $w \in L(M)$}
  E_{TM} = {<M> | M è una TM e L(M) = Ø}

  Teorema: $A_{TM} \leq_m \neg E_{TM}$. (Dimostrazioni Riducibilità, Indecidibilità e Riconoscibilità 1)

- **Teorema:** $\neg E_{TM}$ è indecidibile.
  Infatti $A_{TM} \leq_m \neg E_{TM}$ e A_{TM} indecidibile ⇒ $\neg E_{TM}$ indecidibile.

- **Corollario:** E_{TM} è indecidibile..

- $REGULAR_{TM}$ = {<M> | M è una MdT e L(M) è regolare}
  Teorema: $A_{TM} \leq_m REGULAR_{TM}$. (Dimostrazioni Riducibilità, Indecidibilità e Riconoscibilità 2)

- E_{TM} = {<M> | M è una MdT e L(M) = Ø}
  EQ_{TM} = {<M1,M2> | M1,M2 sono MdT e L(M1) = L(M2)}
  Teorema: $E_{TM} \leq_m EQ_{TM}$.

- **Teorema:** $A_{TM} \leq_m EQ_{TM}$.

- **Teorema:**
  1) $A_{TM} \leq_m HALT_{TM}$
  2) $A_{TM} \leq_m \neg E_{TM}$
  3) $A_{TM} \leq_m REGULAR_{TM}$
  4) $E_{TM} \leq_m EQ_{TM}$
  5) $A_{TM} \leq_m EQ_{TM}$
  6) $A_{TM} \leq_m \neg EQ_{TM}$

- **Corollario:** A_{TM}, $HALT_{TM}$, E_{TM}, $\neg E_{TM}$, $REGULAR_{TM}$, EQ_{TM}, $\neg EQ_{TM}$, sono linguaggi indecidibili.

- **Teorema:** EQ_{TM} non è nè Turing riconoscibile nè co-Turing riconoscibile. (Dimostrazioni Riducibilità, Indecidibilità e Riconoscibilità 3)

- **Teorema di Rice:** Sia L = {<M> | M è una MdT che verifica la proprietà P} un linguaggio che soddisfa le seguenti due condizioni:

  1. P è una proprietà del linguaggio L(M), cioè: prese comunque due MdT M1, M2 tali che L(M1) = L(M2) risulta <M1>∈L ⇔ <M2>∈L

  2. P è una proprietà non banale, cioè: esistono due MdT M3, M4 tali che <M3>∈L; <M4>∉L.

  Allora L è indecidibile.

Teoria della Complessità classe P

- La **classe P** è l'insieme dei linguaggi L per i quali esiste una macchina di Turing deterministica con un solo nastro che decide L in tempo $O(n^k)$ per qualche k ≥ 0, cioè: $P = \cup_{k\geq 0} TIME\ (n^k)$

- **Teorema:**

  $PATH = \{ < G, s, t > \ | \ G \text{ è } un\ grafo\ orientato\ in\ cui\ c'\text{è } un\ cammino\ da\ s\ a\ t \}$

  E' un problema di raggiungibilità nei grafi. Una visita BFS o DFS da s ha tempo lineare nella codifica di una rappresentazione dell'istanza.

- **Teorema:** $RELPRIME \in P$

Teoria della Complessità classe NP

- **Definizione:** La classe EXPTIME = $\cup_{k\geq 1} TIME\ (2^{n^k})$
- **Osservazione:** P ⊆ EXPTIME
- **Definizione**: HAMPATH = { <G, s, t> | G è un grafo orientato, s e t vertici, G ha un cammino Hamiltoniano da s a t }

- **Definizione Algoritmo di verifica**

  Definizione

  Un **algoritmo di verifica (o verificatore)** *V per un linguaggio A è un algoritmo tale che*

  $$A = \{w \mid \exists c \text{ tale che } V \text{ accetta } \langle w, c\rangle\}$$

  La stringa c prende il nome di **certificato** *o* **prova**.

  A è il **linguaggio verificato** *da V*.

- **Definizione:** NP è la classe dei linguaggi verificabili in tempo polinomiale.

- **Teorema**: Un linguaggio L è in NP se e solo se esiste una macchina di Turing non deterministica che decide L in tempo polinomiale.

- **<u>Definizione classe NTIME</u>**

Definizione

Sia $t : \mathbb{N} \to \mathbb{R}^+$ una funzione. La classe di complessità in tempo non deterministico $NTIME(t(n))$ è

$$NTIME(t(n)) = \{L \mid \exists \text{ una macchina di Turing non deterministica } M \text{ che decide } L \text{ in tempo } O(t(n))\}$$

Corollario 7.22

$$NP = \bigcup_{k \geq 0} NTIME(n^k)$$

- **<u>Definizione:</u>** Una clique o cricca in un grafo non orientato G è un sottografo di G in cui ogni coppia di vertici è connessa da un arco. Una k-clique è una clique che contiene k vertici.
- **<u>Teorema</u>**:
  CLIQUE = {<G,k> | G è un grafo non orientato in cui esiste una k-clique}
  CLIQUE ∈ NP.

- **<u>Definizione</u>**: 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-SUM= { <S,t> | S = {x_1,...,x_n} ed esiste S' ⊆ S tale che $\sum_{s \in S'} s$ = t}
- **<u>Teorema</u>**: SUBSET-SUM ∈ NP

- **<u>Teorema</u>**: HAMPATH ∈ NP

- **<u>Proposizione</u>**: La classe P è chiusa rispetto al complemento.
- **<u>Teorema</u>**: P ⊆ NP

Complessità - Riduzioni in tempo polinomiale

- **Definizione:** Una funzione f : Σ* → Σ* è **calcolabile in tempo polinomiale** se esiste una macchina di Turing deterministica M di complessità di tempo polinomiale tale che su ogni input w, M si arresta con f(w) sul suo nastro.

- **Definizione:** Siano A e B linguaggi sull'alfabeto Σ. Una **riduzione** in tempo **polinomiale** f di A in B è una funzione f : Σ* → Σ* calcolabile in tempo polinomiale tale che ∀w ∈ Σ* abbiamo che w ∈ A ⇔ f (w) ∈ B

- **Definizione**: Un linguaggio A ⊆ Σ* è **riducibile** in tempo **polinomiale** a un linguaggio B ⊆ Σ*, e scriveremo A ≤p B, se esiste una riduzione di tempo polinomiale di A in B

- **Definizione:** Una formula booleana ϕ è soddisfacibile se esiste un insieme di valori 0 o 1 per le variabili che rendono ϕ soddisfatta (uguale a 1)

- **Definizione:** Una formula booleana ϕ è in forma CNF se è un AND di clausole (una clausola è un OR di letterali). Una formula booleana ϕ è in forma 3CNF se è un AND di clausole e ogni clausola ha 3 letterali.

- **Teorema**: 3SAT ≤p CLIQUE.

Teoria della Complessità - NP-completezza

- **Teorema:** Se A $\leq_p$ B e B∈P, allora A∈P (dim.1)
- **Teorema:** Se A $\leq_p$ B e B $\leq_p$ C allora A $\leq_p$ C. (dim.2)
- **Teorema:** P⊆NP. (dim.3)

- **Definizione:** Linguaggio NP-Completo

  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)

- **Teorema:** Se B è NP-completo e B è in P allora P = NP (dim.4).
- **Teorema (Cook-Levin):** SAT è NP-completo. (No Dim)
- **Teorema:** Se B è NP-completo e B $\leq_p$ C, con C∈NP, allora C è NP-completo. (dim.5)
- **Teorema:** SAT_{CNF} è NP-completo
- **Teorema:** 3SAT è NP-completo (dim.6)
- **Teorema:** CLIQUE è NP-completo (dim.7)

Linguaggi NP completi

- **Definizione: VERTEX-COVER =** {<G,k> | G è un grafo non orientato che ha un vertex cover di cardinalità k}
- **Teorema**: **VERTEX-COVER** ∈ NP
- **Teorema**: **VERTEX-COVER** è NP-completo

- **Definizione: SUBSET-SUM =** $\{< S, t > | \, S = \{x_1, \ldots\ldots, x_k\} \; ed \; esiste \; S' \subseteq S \; tale \; che \; \sum_{s \in S'} \boldsymbol{s} = t \,\}$
- **Teorema**: **SUBSET-SUM** ∈ NP
- **Teorema**: **SUBSET-SUM** è NP-completo

- **Teorema**: **HAMPATH** ∈ NP
- **Teorema**: **HAMPATH** ∈ NP-completo

- **Definizione: UHAMPATH** = { <G, s, t > | G è un grafo non-orientato, s e t vertici e ha un cammino hamiltoniano da s a t}
- **Teorema**: **UHAMPATH** ∈ NP
- **Teorema**: **UHAMPATH** ∈ NP-completo

Decidibilità

Linguaggio	**Problema**	**Linguaggio Associato**	**Decidibilità**	**Riduzioni**
A_{TM}	Accettazione	$\{<M, w> \mid M$ è una MdT che accetta $w\}$	**- Non decidibile - Riconoscibile - Non co-Turing Riconoscibile**	$A_{TM} <_m HALT_{TM}$ $A_{TM} <_m \neg E_{TM}$ $A_{TM} <_m REGULAR_{TM}$ $A_{TM} <_m EQ_{TM}$ $A_{TM} <_m \neg EQ_{TM}$
$HALT_{TM}$	Fermata	$\{<M, w> \mid M$ è una MdT che si arresta su $w\}$	**- Non decidibile - Non co-Turing riconoscibile**	$A_{TM} <_m HALT_{TM}$
E_{TM}	Vuoto	$\{<M> \mid L(M) = \emptyset\}$	**- Non decidibile - Non riconoscibile - co-Turing riconoscibile**	$A_{TM} <_m \neg E_{TM}$ $E_{TM} <_m EQ_{TM}$
$REGULAR_{TM}$	Regolare	$\{<M> \mid L(M)$ è regolare$\}$	**- Non decidibile - Non co-Turing riconoscibile**	$A_{TM} <_m REGULAR_{TM}$
EQ_{TM}	Uguaglianza	$\{<M_1, M_2> \mid L(M_1) = L(M_2)\}$	**- Non decidibile - Non riconoscibile - Non co-Turing riconoscibile**	$E_{TM} <_m EQ_{TM}$ $A_{TM} <_m EQ_{TM}$ $A_{TM} <_m \neg EQ_{TM}$

Complessità

Linguaggio	Problema	Linguaggio Associato	Classe	Riduzioni
PATH	Cammino	{<G, s, t> | G grafo orientato in cui c'è un cammino da s a t}	**P**	-
RELPRIME	Co-Primi	{<x, y> | x e y sono primi tra loro}	**P**	-
HAMPATH	Cammino Hamiltoniano	{<G, s, t> | G grafo orientato in cui esiste un cammino hamiltoniano da s a t}	**NP-COMPLETO**	3SAT $<_p$ HAMPATH HAMPATH $<_p$ UHAMPATH
CLIQUE	k-Clique	{<G, k> | G grafo non orientato in cui esiste una k-clique}	**NP-COMPLETO**	3SAT $<_p$ CLIQUE
SUBSET-SUM	Zaino semplificato	{<S, t> |$S \subseteq N$, $\exists S' \subseteq S : \sum_{x \in S'} x = t$}	**NP-COMPLETO**	3SAT $<_p$ SUBSET-SUM
SAT	Soddisfacibilità	{ <Φ> | Φ è soddisfacibile }	**NP-COMPLETO**	SAT $<_p$ 3SAT
SAT_{CNF}	Soddisfacibilità in CNF	{ <Φ> | Φ è soddisfacibile e in CNF}	**NP-COMPLETO**	SAT $<_p$ SAT_{CNF}
3SAT	Soddisfacibilità 3CNF	{ <Φ> | Φ è soddisfacibile e in 3CNF}	**NP-COMPLETO**	SAT $<_p$ 3SAT 3SAT $<_p$ CLIQUE 3SAT $<_p$ VERTEX-COVER 3SAT $<_p$ SUBSET-SUM 3SAT $<_p$ HAMPATH
VERTEX-COVER	Vertex cover	{ <G, k> | G grafo non orientato in cui esiste un vertex-cover di taglia k }	**NP-COMPLETO**	3SAT $<_p$ VERTEX-COVER
UHAMPATH	Cammino Hamiltoniano non orientato	{<G, s, t> | G grafo non orientato in cui esiste un cammino hamiltoniano da s a t}	**NP-COMPLETO**	HAMPATH $<_p$ UHAMPATH