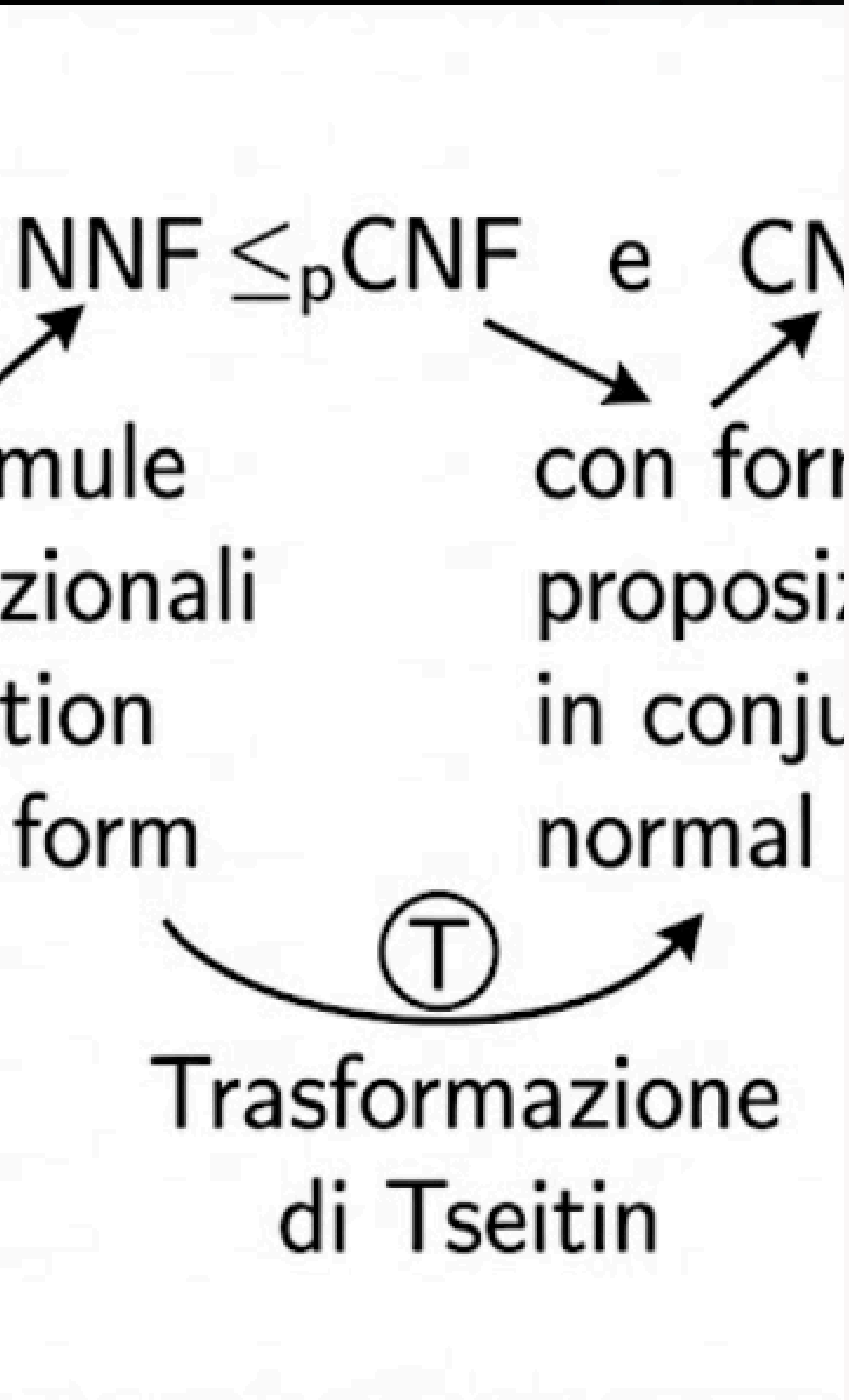


Catena di riduzioni

$$SAT \leq_p NNF \leq_p CNF \leq_p 3CNF$$

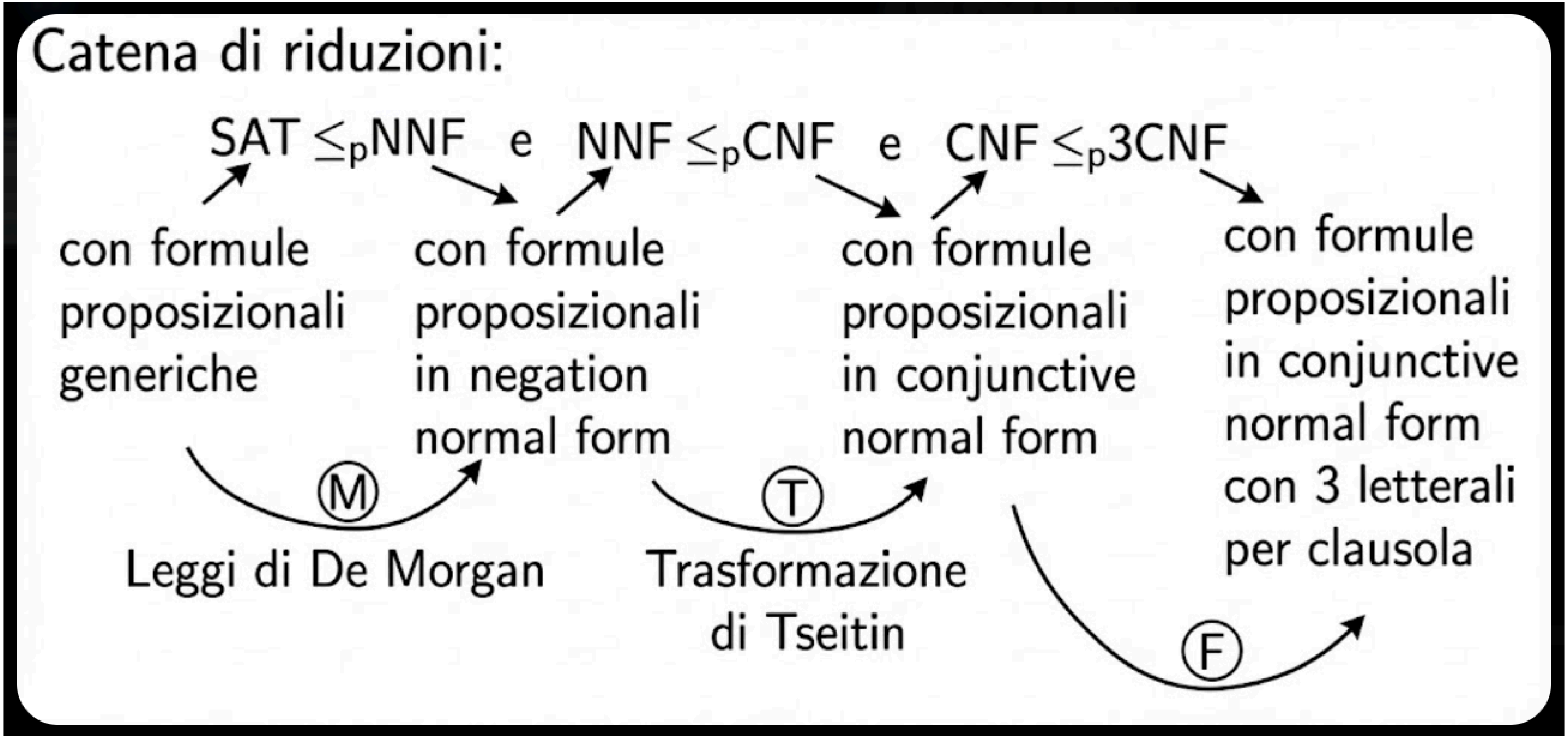
Aspetti fondamentali relativi alla catena di riduzioni

Gabriele Brizio

Domanda 2.3

Algoritmi e Complessità

Panoramica


Catena di riduzioni:
SAT $\leq_p$ NNF e NNF $\leq_p$ CNF e CNF $\leq_p$ 3CNF
con formule proposizionali generiche
con formule proposizionali in negation normal form
con formule proposizionali in conjunctive normal form
con formule proposizionali in conjunctive normal form con 3 letterali per clausola
M
Leggi di De Morgan
T
Trasformazione di Tseitin
F

Normalizzazioni sintattiche delle formule

NNF — Negation Normal Form

Una formula è in **NNF** se:

- le negazioni compaiono solo davanti a variabili,
- non compaiono negazioni su formule composte,

Esempio

Formula arbitraria:

$F = \neg(p \wedge (q \vee \neg r))$

Forma NNF:

$NNF(F) = (\neg p \vee (\neg q \wedge r))$

CNF — Conjunctive Normal Form

Una formula è in **CNF** se è una:

- congiunzione di clausole,
- ognuna composta da disgiunzioni di letterali.

Struttura:

$$(l_{1,1} \vee l_{1,2} \vee \ldots) \ \wedge \ (l_{2,1} \vee l_{2,2} \vee \ldots) \ \wedge \ldots$$

Esempio

$(p \vee \neg q \vee r) \ \wedge \ (\neg p \vee s)$

3CNF — 3-Conjunctive Normal Form

Una formula è in 3CNF se:

- è una CNF,
- ogni clausola contiene esattamente 3 letterali.

Esempio

$$(p \lor q \lor r) \ \land \ (\neg p \lor s \lor t)$$

I problemi di soddisfacibilità

SAT è il seguente problema decisionale:

Dato una formula proposizionale F, esiste un'assegnazione φ che rende F vera?

Dati i rispettivi insiemi sintattici (NNF, CNF, 3CNF), si definiscono i problemi:

Dato $F \in NNF/CNF/3CNF$, esiste un'assegnazione φ che la soddisfa?

Questi problemi sono equivalenti dal punto di vista della soddisfacibilità, ma diversi dal punto di vista strutturale, il che permette di costruire la catena di riduzioni:

$$SAT \leq_p NNF \leq_p CNF \leq_p 3CNF$$

Prima riduzione: $SAT \leq_p NNF$

Obiettivo

Trasformare qualsiasi formula proposizionale in **Negation Normal Form**, dove:

- le negazioni compaiono solo davanti alle variabili,
- non sono ammesse negazioni di formule composte.

Trasformazione

Ricorsivamente si applicano:

- doppia negazione: $\neg\neg G \to G$
- **leggi di De Morgan**:
  $\neg(G \land H) = (\neg G \lor \neg H)$
  $\neg(G \lor H) = (\neg G \land \neg H)$

Correttezza

La trasformazione è:

- polinomiale,
- strutturale,
- semanticamente corretta:

$$F \in SAT \iff NNF(F) \in SAT$$

Seconda riduzione: $NNF \leq_p CNF$

Obiettivo

Convertire una formula in NNF in **Conjunctive Normal Form** (CNF).

Trasformazione di Tseitin

- Si interpreta la formula come un circuito.
- A ogni sottoformula si associa una nuova variabile a_G.
- Per ogni operatore si aggiungono vincoli CNF che esprimono:

$$a_G \leftrightarrow (g \circ h)$$

Il risultato è un insieme di clausole che "certifica" il comportamento dell'intera formula.

$$F \in NNF \iff T(F) \in CNF$$

Dimostrazione

Teorema: Per ogni formula $f \in NNF$ vale:

$$f \in SAT \iff T(f) \in CNF\text{-}SAT$$

Per definizione della trasformazione di Tseitin: $T(f) = [a], C[a, f]$

Assumiamo $f \in NNF$ e sia φ un'assegnazione tale che $\exists\varphi\ \phi_{\varphi}(f) = \mathrm{true}$

Osserviamo che esiste un'estensione φ' di φ alle variabili introdotte da T tale che:

$$\varphi'(C[a_f, f]) = \mathrm{true} \ \wedge\ \varphi'(a_f) = \mathrm{true}.$$

Per il **Lemma T**, vale l'equivalenza strutturale:

$$\exists\varphi\ \phi_{\varphi}(f) = \mathrm{true} \iff \left(\exists\varphi'\ \phi_{\varphi'}(C[a, f]) = \mathrm{true}\ \land\ \varphi'(a) = \mathrm{true}\right).$$

Dalla direzione sinistra verso destra otteniamo dunque che

$$\left(\exists\varphi'\ \phi_{\varphi'}(C[a, f]) = \mathrm{true}\ \land\ \varphi'(a) = \mathrm{true}\right)$$

è equivalente a

$$\left(\exists\varphi'\ \phi_{\varphi'}(T(f)) = \mathrm{true}\right)$$

cioè:

$$T(f) \in CNF.$$

Lemma T: schema di dimostrazione

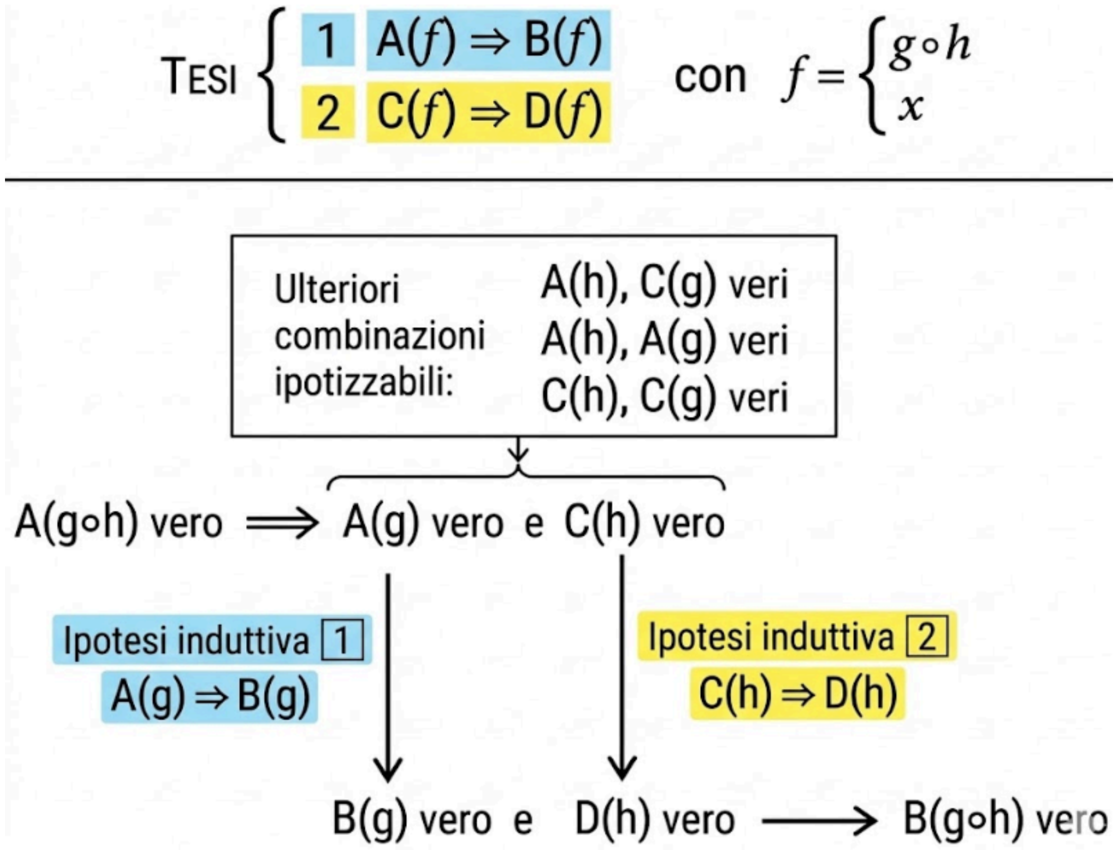
Partendo da una tesi in due punti:

1. $A(f) \Rightarrow B(f)$
2. $C(f) \Rightarrow D(f)$

ed una funzione f composta $f = g \circ h$

Il passo induttivo dipenderà da quanto è difficile f (nel nostro caso $g \circ h$)

Conoscendo g ed h se si riesce a dimostrare che A è vera su g e C è vera su h allora posso usare il modus ponens



Usare la struttura di f per risalire le proprietà

Dal fatto che $A(f)$ sia vero, possiamo ricavare condizioni su $A(g)$ e $C(h)$.

Questo step dipende dal tipo di operatore ($\wedge$, $\vee$, $\neg$), ma lo schema è sempre:

$$A(f) \text{ vero} \quad \Rightarrow \quad A(g) \text{ vero e } C(h) \text{ vero}$$

$A(g \circ h)$ vero $\Rightarrow$ $A(g)$ vero e $C(h)$ vero

Applicazione delle ipotesi induttive

Da qui il diagramma si biforca:

- **Da $A(g)$** la prima ipotesi induttiva dà
$$A(g) \Rightarrow B(g)$$
- **Da $C(h)$** la seconda ipotesi induttiva dà
$$C(h) \Rightarrow D(h)$$

Ricombinazione per concludere su f

A questo punto abbiamo:

- $B(g)$ vero
- $D(h)$ vero

Ora, per la struttura dell'operatore $\circ$, questo implica:

$B(g)$ vero e $D(h)$ vero $\Rightarrow B(g \circ h)$ vero

Conclusione dello schema

L'intero Lemma T dice, in sostanza:

Se si riesce a dimostrare la tesi per i casi base (variabili)
e ogni passo induttivo per i costruttori (negazione e operatori binari) segue lo schema "scomponi → applica ipotesi → ricompone",

allora le due implicazioni $A(f) \Rightarrow B(f)$ e $C(f) \Rightarrow D(f)$ valgono per tutte le formule.

Terza riduzione: $CNF \leq_p 3CNF$

Obiettivo

Convertire ogni clausola in una formula in cui ogni clausola ha **esattamente 3 letterali**.

Strategia

Una clausola lunga come:

$$(l_1 \vee l_2 \vee l_3 \vee l_4 \vee l_5)$$

viene riscritta introducendo nuove variabili ausiliarie:

$$(l_1 \vee l_2 \vee a_1) \wedge (\neg a_1 \vee l_3 \vee a_2) \wedge (\neg a_2 \vee l_4 \vee l_5)$$

Le nuove variabili mantengono la soddisfacibilità senza crescita esponenziale.

Relazione fra le riduzioni

La catena:

$$SAT \leq_p NNF \leq_p CNF \leq_p 3CNF$$

ha le seguent proprietà fondamentali:

- ogni trasformazione è polinomiale,
- ogni trasformazione preserva la soddisfacibilità,
- la struttura sintattica viene progressivamente normalizzata.

Risultato:
F è soddisfacibile ⇔ la sua forma $3CNF(F)$ lo è.

Importanza della catena

Questa sequenza di riduzioni:

- è il cuore della dimostrazione che **3SAT è NP-completo**,
- mostra come sia possibile raggiungere forme sintattiche altamente vincolate mantenendo la semantica,
- introduce tecniche fondamentali usate in tutte le riduzioni NP-complete (nuove variabili, Tseitin, splitting).

Rappresenta uno degli esempi più eleganti di trasferimento controllato di complessità tramite normalizzazione sintattica.

Sintesi finale

La catena $SAT \to NNF \to CNF \to 3CNF$:

- usa trasformazioni polinomiali,
- mantiene la soddisfacibilità,
- produce formule sempre più strutturate,
- permette di capire perché 3SAT è un problema “universale” per NP,
- è supportata dal Lemma T che fornisce lo schema di correttezza induttiva.