

Efficient Parity Decision Trees and Their Connections to Logical Proofs and Total Search Problems in NP

Facoltà di Ingegneria dell'informazione, informatica e statistica
Corso di Laurea in Informatica



SAPIENZA
UNIVERSITÀ DI ROMA

Candidato: Simone Bianco

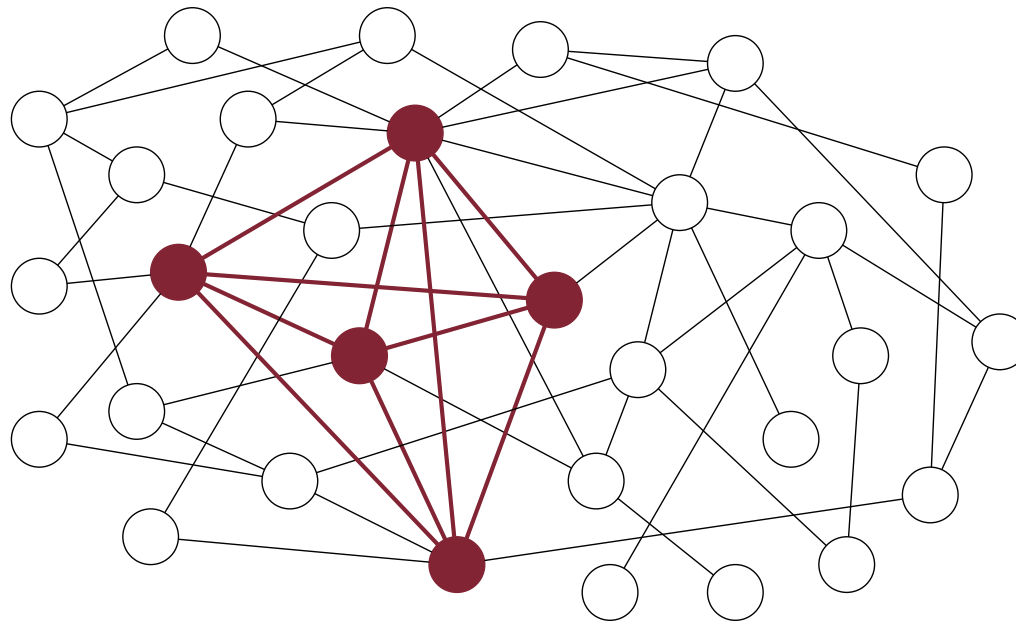
Relatore: Nicola Galesi
Co-relatore: Massimo Lauria

Anno Accademico: 2023/2024

Problema di Ricerca

Un problema di ricerca è un qualsiasi problema il cui obiettivo è trovare una soluzione, proprietà o sotto-struttura all'interno di una specifica istanza del problema.

Esempio: Trovare una 5-clique all'interno di un grafo



Problema di Ricerca

FP = Problema di ricerca **risolvibili** in tempo polinomiale

FNP = Problema di ricerca **verificabili** in tempo polinomiale

TFNP = Problema di ricerca **totali** in FNP

└─→ Ogni istanza ha almeno una soluzione

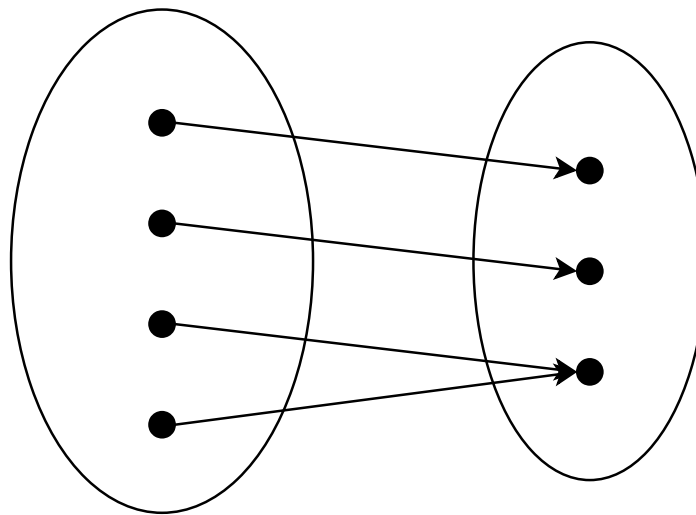
Trovare una 5-clique all'interno di un grafo ─→ Non Totale!

Trovare una fattorizzazione di un numero naturale ─→ Totale!

Polynomial Pigeonhole Principle (PPP)

Data una funzione $f : [n] \rightarrow [n-1]$, trovare una collisione

Vincolo: la descrizione della funzione è esponenziale rispetto all'input del problema



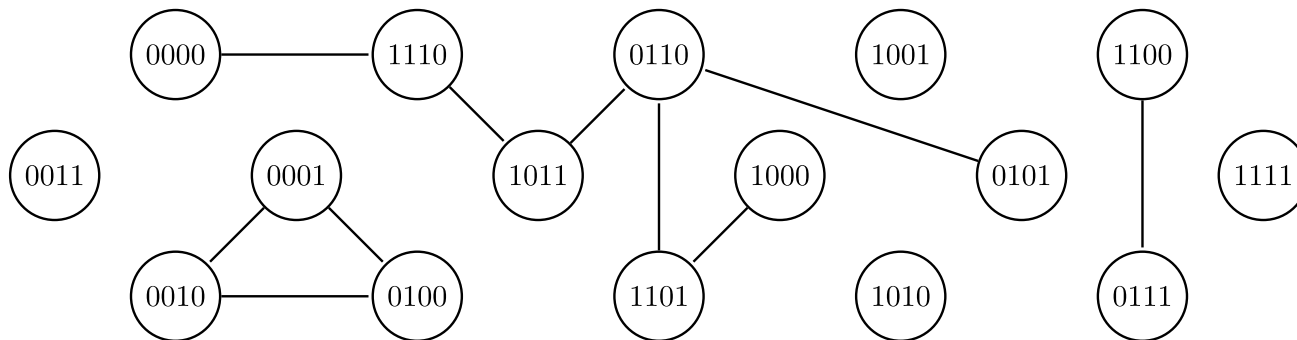
Totalità garantita dal **Principio della Piccionaia**:
Non può esistere una funzione $f : [n] \rightarrow [n-1]$ che sia anche iniettiva



Polynomial Parity Argument (PPA)

Dato un nodo con grado dispari etichettato da n bit, trovare un altro nodo di grado dispari

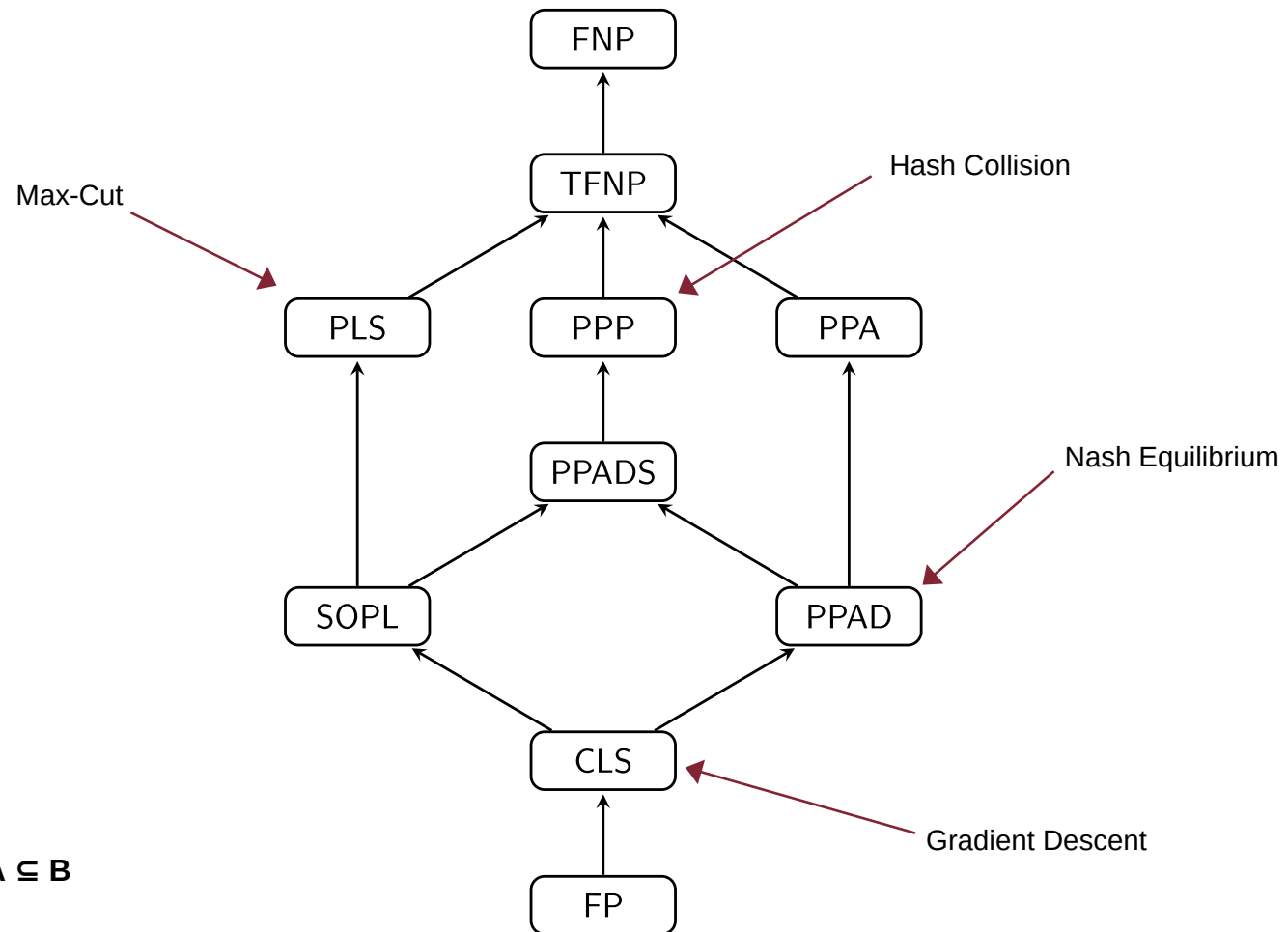
Vincolo: la descrizione del grafo è esponenziale rispetto all'input del problema



Totalità garantita dall'**Handshaking Lemma**:
Ogni grafo possiede un numero pari di nodi con grado dispari



La Gerarchia TFNP

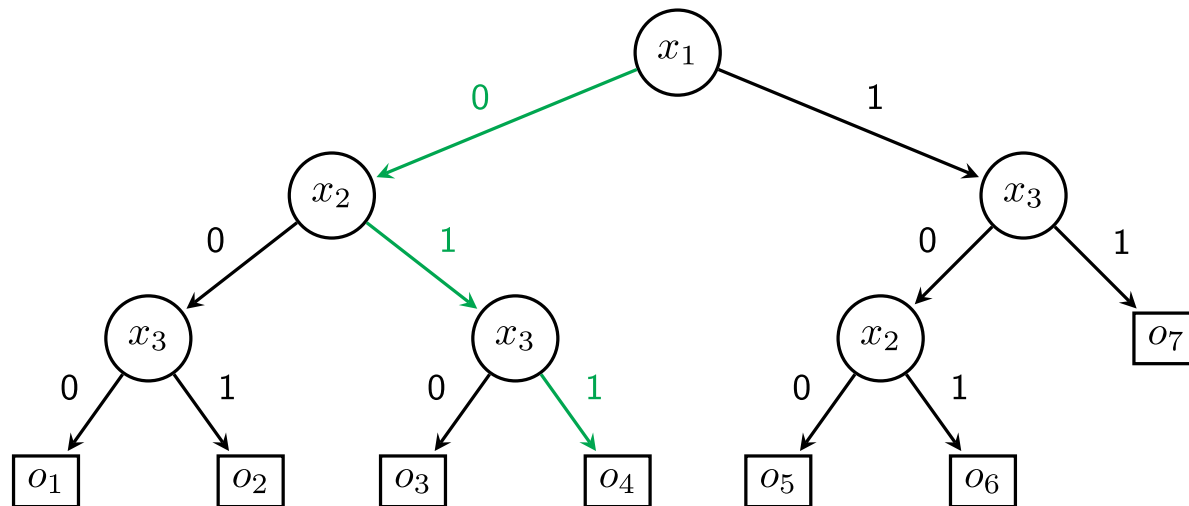


Legenda:

$A \rightarrow B \equiv A \subseteq B$

Il Modello Black-box

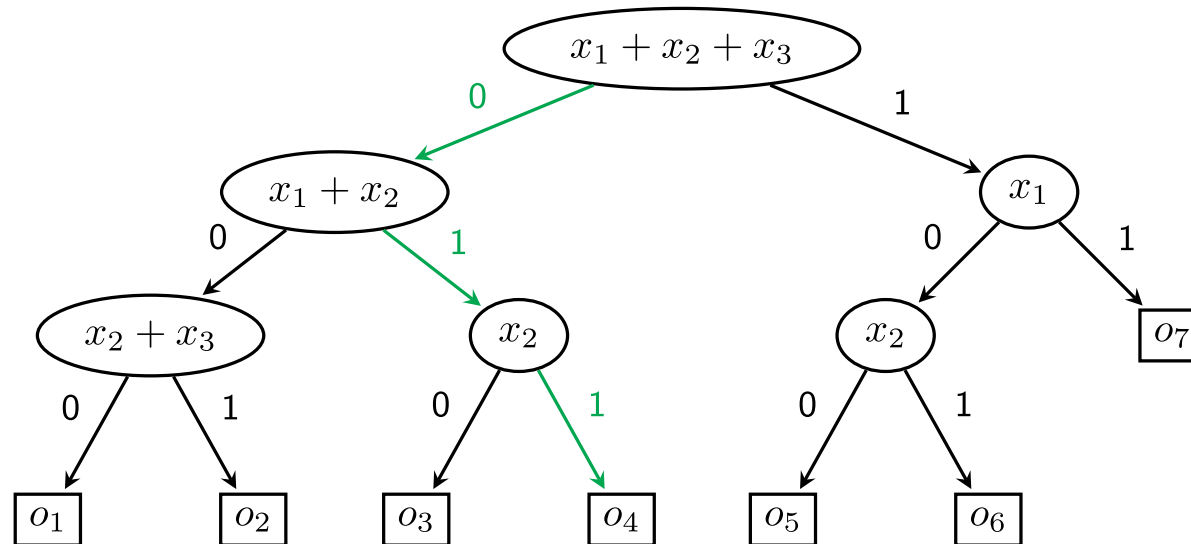
Ogni computazione svolta da una macchina dotata di oracolo può essere vista come un Decision Tree



Restringiamo il nostro interesse alla Query Complexity

La Classe FP_{pdt}

Un Parity Decision Tree effettua query di parità su k variabili

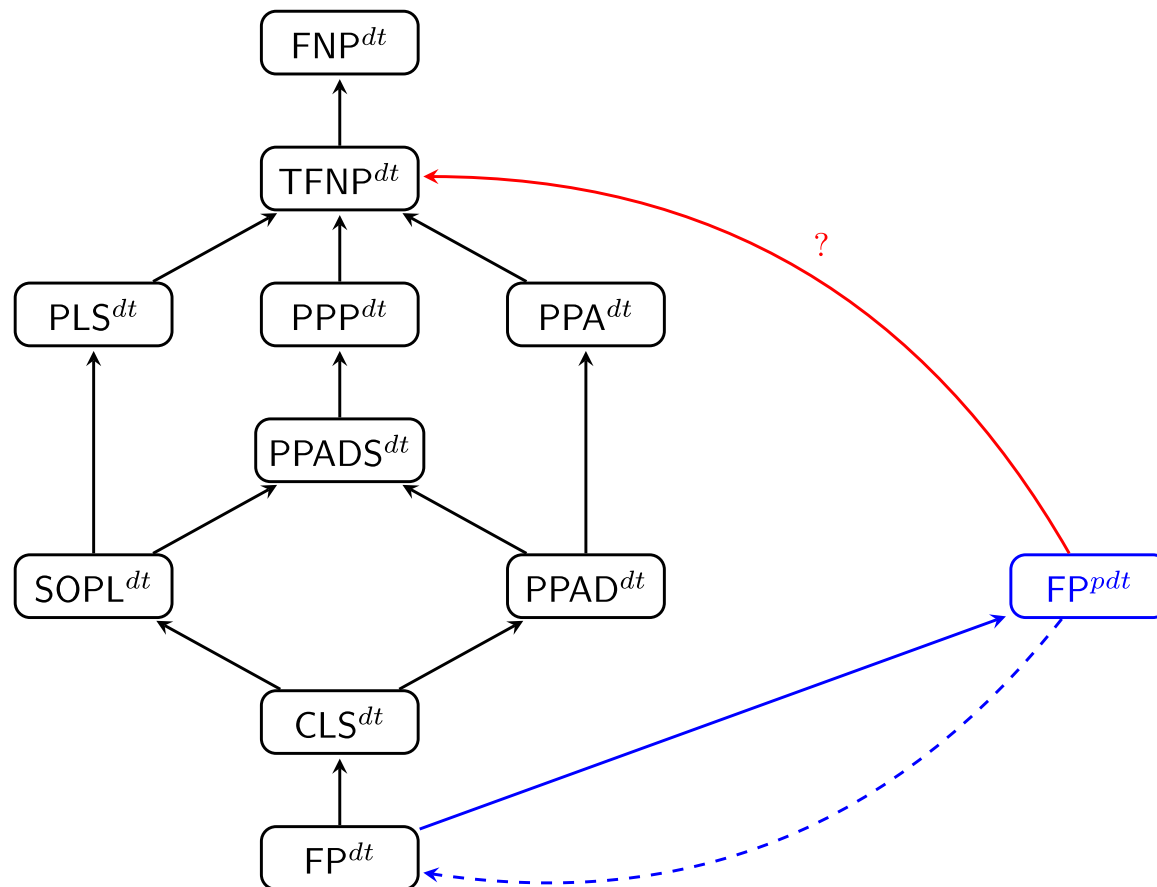


I PDT hanno un potere computazionale maggiore rispetto ai DT

Esempio: $XOR(x_1, \dots, x_n)$ richiede un DT di altezza $\Omega(n)$ e un PDT di altezza $O(1)$

Estensione del Modello Black-box

Dov'è collocata la nuova classe?



Legenda:

$A \rightarrow B \equiv A \subseteq B$

$A \dashrightarrow B \equiv A \not\subseteq B$

Relazioni con la Proof Complexity

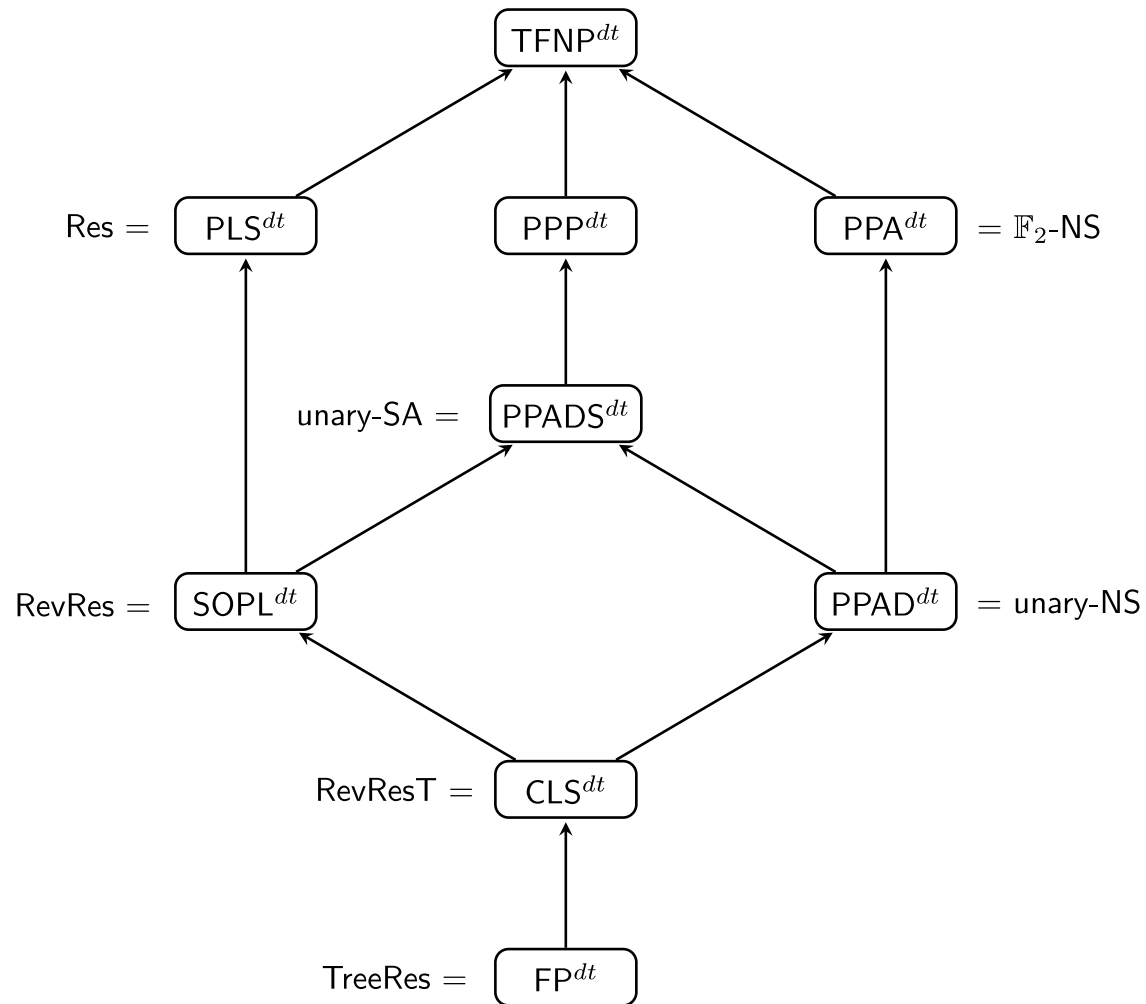
Ogni DT può essere codificato come una CNF insoddisfacibile che genera un problema di ricerca equivalente

$$\varphi_T = \bigwedge_{p : \text{path in } T} \neg \varphi_p \longrightarrow \text{Search}(\varphi_T) : \text{Dato un assegnamento per } x_1, \dots, x_n, \text{ trovare la clausola falsificata}$$

Otteniamo che $R \equiv \text{Search}(\varphi_{T_R})$

TFNP^{dt} equivale allo studio del Problema della Clausola Falsificata

TFNP^{dt} e Sistemi Logici di Dimostrazione



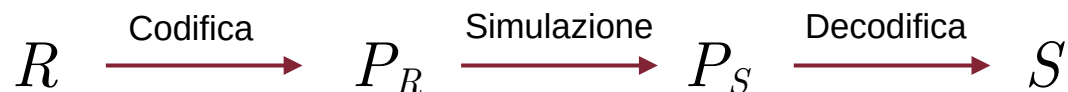
Legenda:

$A \rightarrow B \equiv A \subseteq B$

Riduzioni tramite Proof Systems

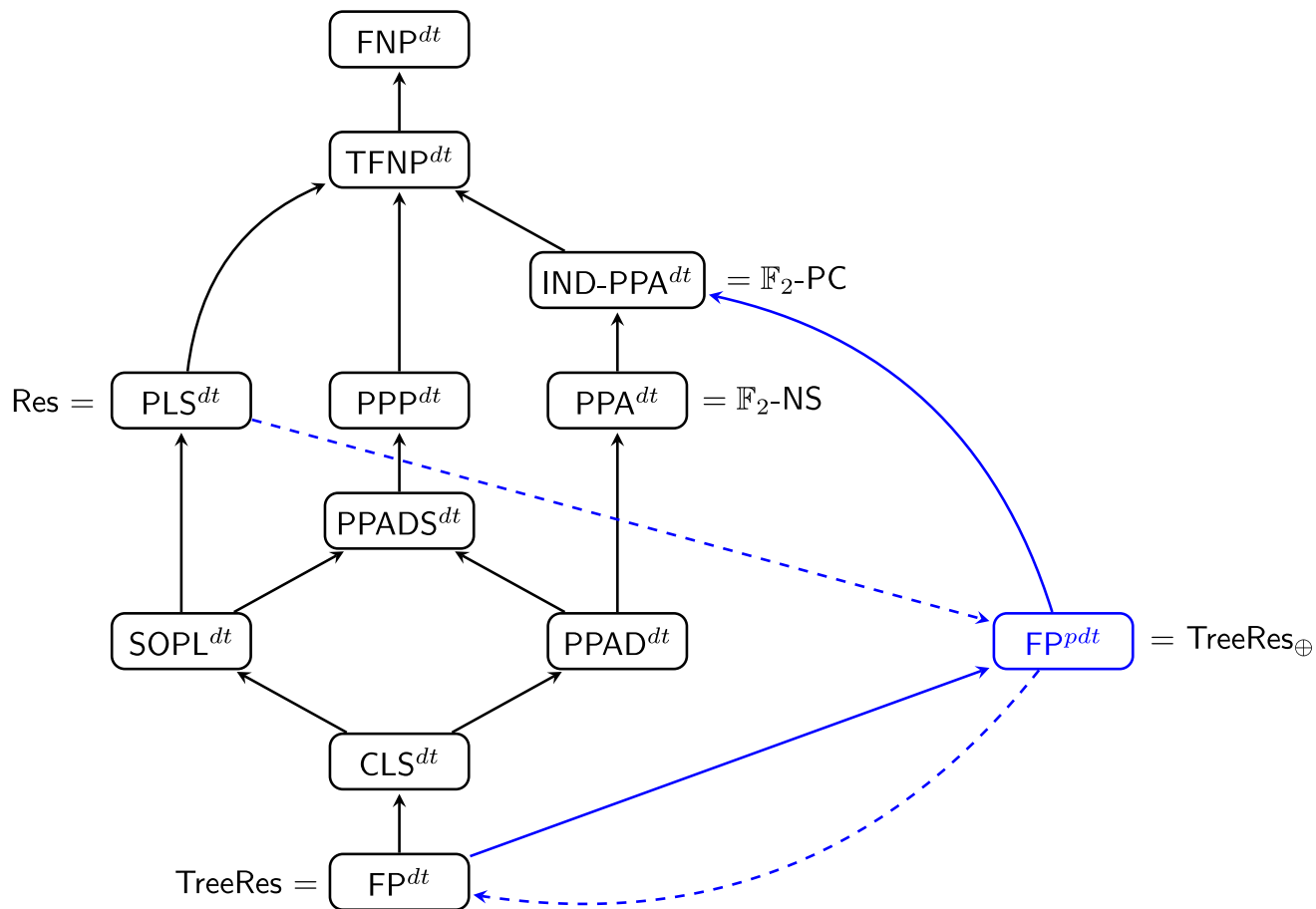
(Thm.) Un problema $R^{dt} \in \text{TFNP}^{dt}$ è nella classe $\mathcal{S}^{dt} \subseteq \text{TFNP}^{dt}$ se e solo se φ_R è dimostrabile efficientemente in P_S

Idea: Passare tramite i proof system per ottenere inclusioni o separazioni tra le classi che essi caratterizzano



Estensione del Modello Black-box

(Thm.) TreeRes_{\oplus} caratterizza FP^{pdt}



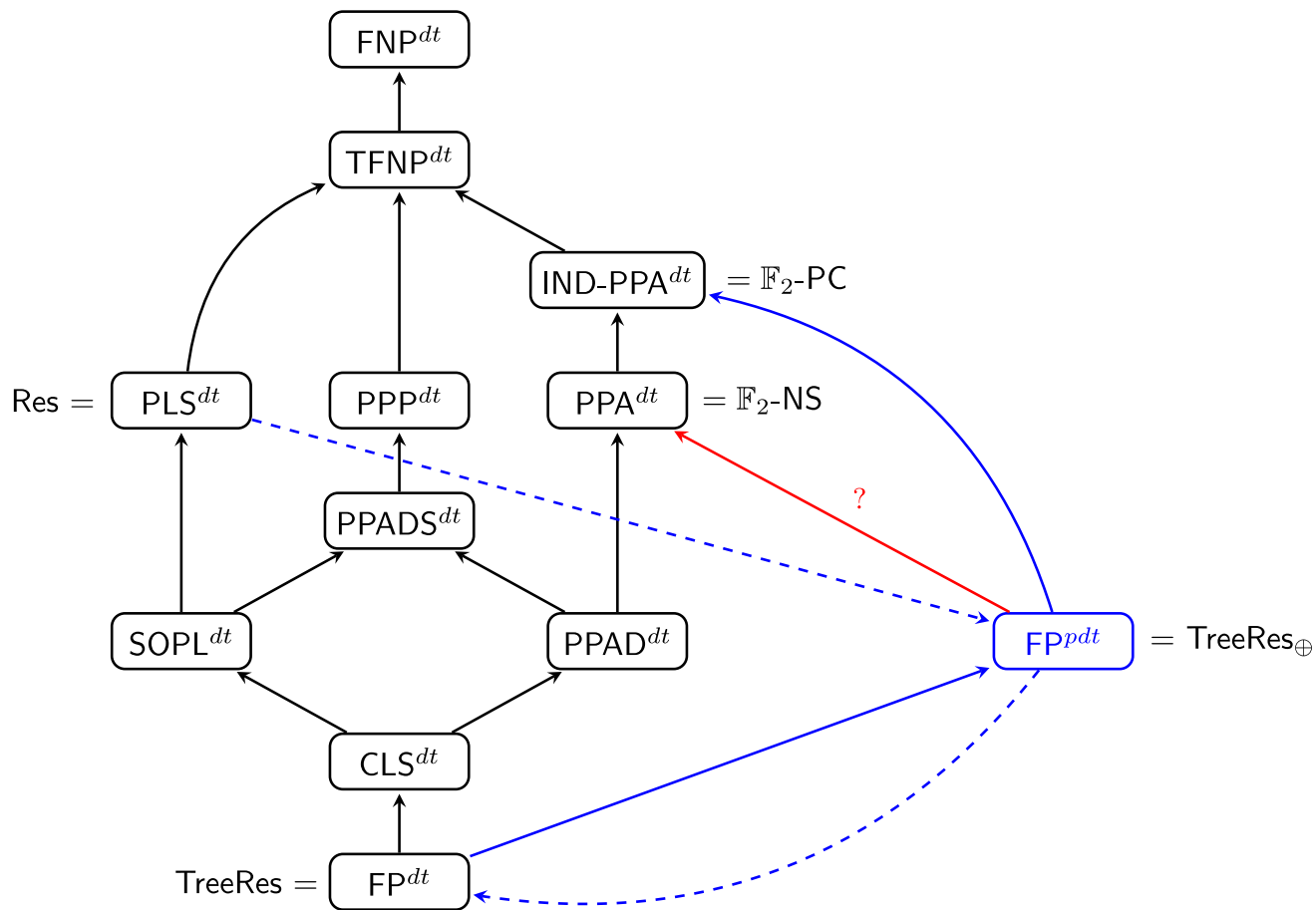
Legenda:

$A \rightarrow B \equiv A \subseteq B$

$A \dashrightarrow B \equiv A \not\subseteq B$

Estensione del Modello Black-box

Ipotesi: l'inclusione può valere anche per PPA^{dt}



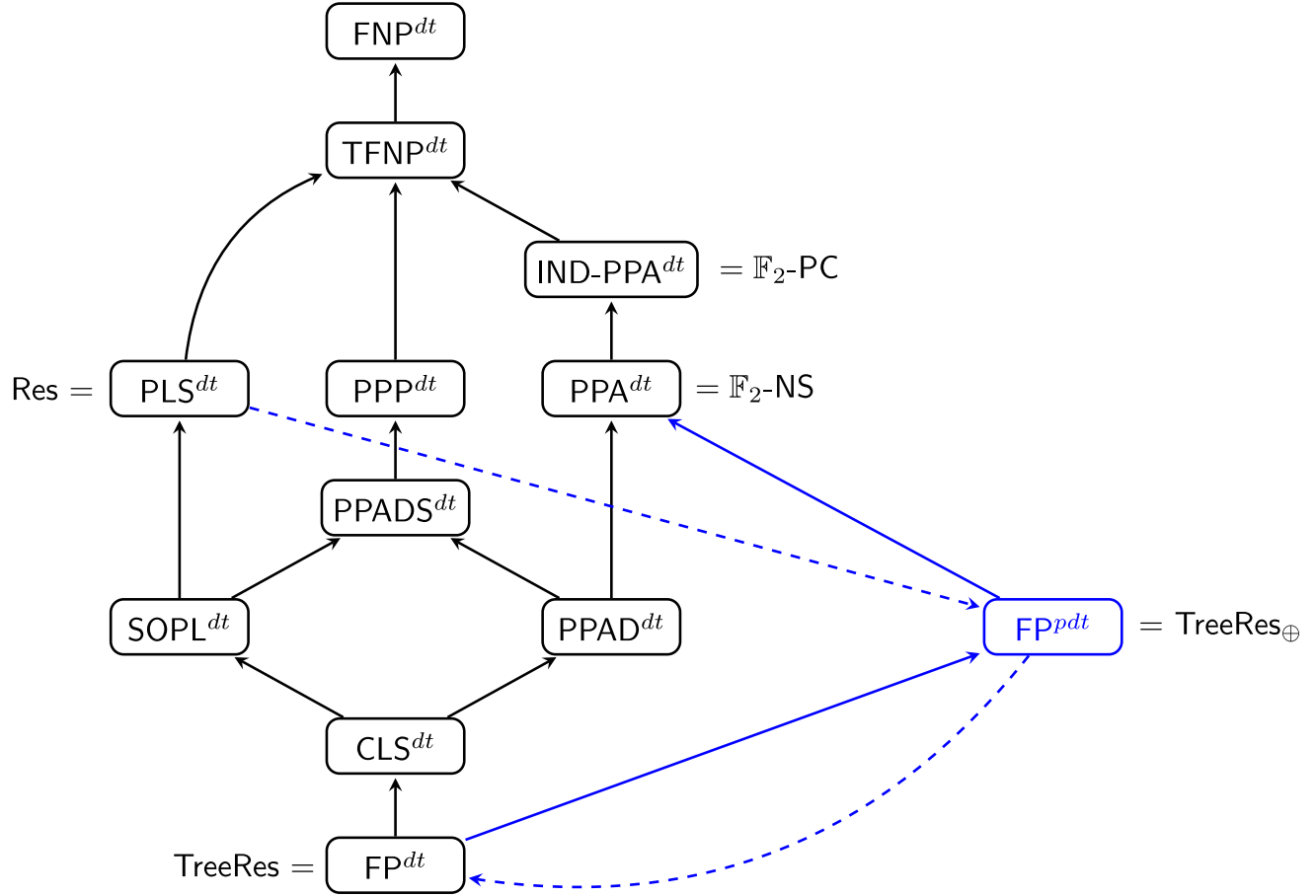
Legenda:

$A \rightarrow B \equiv A \subseteq B$

$A \dashrightarrow B \equiv A \not\subseteq B$

Estensione del Modello Black-box

(Thm.) \mathbb{F}_2 -Nullstellensatz simula efficientemente TreeRes_\oplus



Lavori futuri

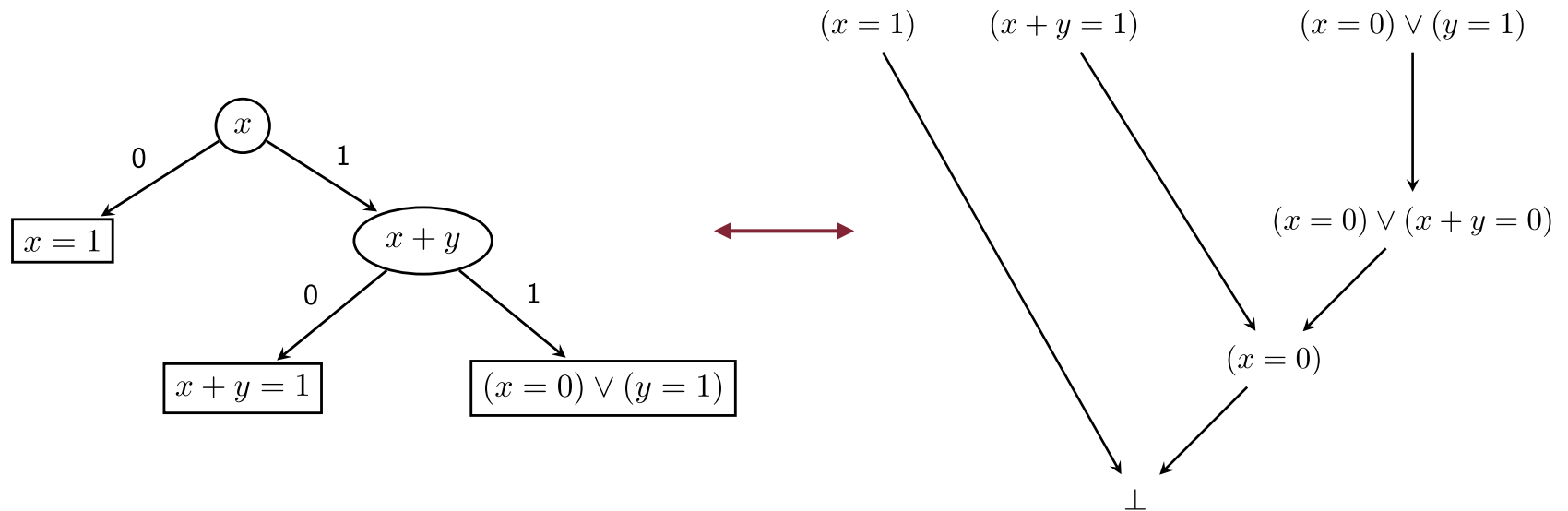
- Dimostrare inclusioni più forti per FP^{pdt}
- Delineare ogni separazione tra FP^{pdt} e le altre classi
- Generalizzare i risultati ottenuti per \mathbb{F}_q invece di \mathbb{F}_2
- Confrontare la gerarchia $TFNP^{dt}$ con la gerarchia $TFNP^{pdt}$
- Riduzioni tramite PDT invece di DT

Grazie per l'attenzione

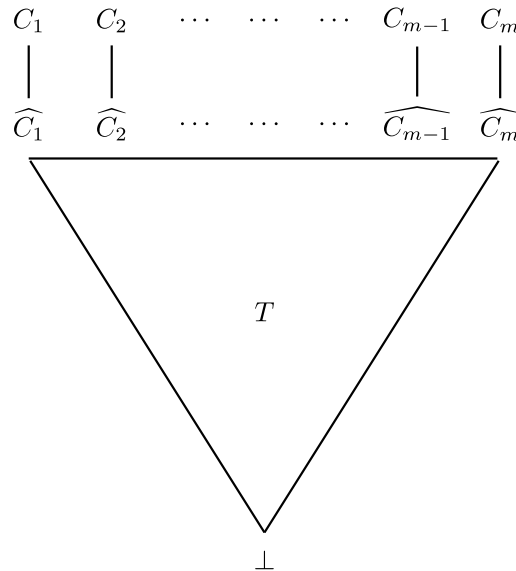


Tree-like Linear Resolution su \mathbb{F}_2

(Thm.) $FP^{pdt}(\text{Search}(\varphi)) = \Theta(\text{TreeRes}_{\oplus}(\varphi))$



Simulare TreeRes_⊕ in \mathbb{F}_2 -NS



$$\sum_{i=1}^m p_{\widehat{C}_i} + \sum_{i=1}^n x_i(1 - x_i) = 1$$



$$\sum_{i=1}^m p_{C_i} + \sum_{i=1}^n x_i(1 - x_i) = 1$$