2.2 DPLL

sabato 26 marzo 2022 11:54

DPLL

DPLL è un algoritmo di ricerca esaustiva, utilizzato per decidere la soddisfacibilità booleana di formule di logica proposizionale in forma normale congiuntiva (CNF), problema noto come CNF-SAT.

CNF

Una formula è in CNF quando è una congiunzione (and: ∧) di clausole

Letterale: P oppure ¬P (P=proposizione)
Clausola: P v Q v ¬R (disgiunzione or di più letterali)

Una formula CNF ha la seguente forma
$(A \vee B) \wedge (C \vee D) \wedge \ldots \wedge (P \vee Q)$

Qualunque formula può essere scritta in CNF

$CNF(p)$	=	p if $p \in PROP$
$CNF(\neg p)$	=	$\neg p$ if $p \in PROP$
$CNF(\varphi \supset \psi)$	=	$CNF(\neg\varphi) \otimes CNF(\psi)$
$CNF(\varphi \wedge \psi)$	=	$CNF(\varphi) \wedge CNF(\psi)$
$CNF(\varphi \vee \psi)$	=	$CNF(\varphi) \otimes CNF(\psi)$
$CNF(\varphi \equiv \psi)$	=	$CNF(\varphi \supset \psi) \wedge CNF(\psi \supset \varphi)$
$CNF(\neg\neg\varphi)$	=	$CNF(\varphi)$
$CNF(\neg(\varphi \supset \psi))$	=	$CNF(\varphi) \wedge CNF(\neg\psi)$
$CNF(\neg(\varphi \wedge \psi))$	=	$CNF(\neg\varphi) \otimes CNF(\neg\psi)$
$CNF(\neg(\varphi \vee \psi))$	=	$CNF(\neg\varphi) \wedge CNF(\neg\psi)$
$CNF(\neg(\varphi \equiv \psi))$	=	$CNF(\varphi \wedge \neg\psi) \otimes CNF(\psi \wedge \neg\varphi)$

Esempio esercizio

$CNF((a \wedge b) \vee \neg(c \supset d))$ =
$CNF(a \wedge b) \otimes CNF(\neg c(\supset d))$ =
$(CNF(a) \wedge CNF(b)) \otimes (CNF(c) \wedge CNF(\neg d))$ =
$(a \wedge b) \otimes (c \wedge \neg d)$ =
$(a \vee c) \wedge (a \vee \neg d) \wedge (b \vee c) \wedge (b \vee \neg d)$

Verificare che una formula in CNF sia soddisfacibile/valida è molto più semplice che usando una formula normale
-validità/soddisfacibilità di una formula CNF può essere provata in tempo lineare

DNF

Una formula è in DNF quando è una disgiunzione (or: ∨) di clausole

SODDISFACIBILITA' DI UN CNF

Una formula CNF è **soddisfacibile quando, semplificandola, raggiungiamo l'insieme vuoto**

Per semplificare una formula CNF:

- assumo che un valore P o ¬P sia vero aggiungendo |P o |¬P a fine formula e sostituendolo con T
- sostituisco i valori Q con ⊥ se per la mia assunzione risultano falsi (ad esempio dico che P è T quindi ¬P è ⊥)
- semplifico il risultato rimuovendo:
  -tutte le clausole contenenti una disgiunzione (or) con T
  -i letterali ¬T = ⊥ in tutte le clausole rimanenti

Inoltre, devo seguire queste regole:

-una clausola è VERA se almeno uno dei letterali è VERO
-una clausola è FALSA se tutti i letterali sono falsi
-in tutti gli altri casi, una clausola C è UNDEFINED (non risolta)
-una clausola è lasciata UNDEFINED quando il valore di verità di tutti i letterali è irrilevante alla computazione dell'interpretazione corrente