ESERCIZIO 6

Illustrare i concetti di riduzione polinomiale e di self-reducibility.

RIDUZIONE POLINOMIALE

Un problema X si riduce in tempo polinomiale al problema Y se arbitrarie istanze di X possono essere risolte usando:

- un numero polinomiale di passi di computazione per trasformare un'istanza di X in una di Y.
- un numero polinomiale di chiamate all'oracolo che risolve Y.

SELF-REDUCIBILITY

Problema di ricerca $\leq_p$ Problema di decisione.

Un problema di ricerca X si riduce alla versione decisionale dello stesso problema X.

Si considerino il problema SAT ed il suo corrispondente problema di ricerca RICSAT. Mostrare che RICSAT $\leq_p$ SAT.

RICSAT: sia data un'espressione booleana ϕ. Se ϕ è soddisfacibile, determinare l'assegnazione di verità alle variabili che la rende vera; altrimenti, segnalare che ϕ non è soddisfacibile.

Descrivere un algoritmo di ricerca per SAT partendo da un algoritmo di decisione per SAT.

INPUT: formula ϕ con n variabili $x_1, x_2, \_, x_n$

ALGORITMO:

a) Fissa $x_1 = 0$ in ϕ ed ottieni una nuova formula ϕ'. Controlla se ϕ' sia soddisfacibile usando l'algoritmo di decisione per SAT.

Se ϕ' è soddisfacibile, cerca in modo ricorsivo un assegnamento di

SATISFYING-ASSIGNMENT ($\phi(x_1, \_\_, x_n)$)

1. $\psi(x_1, \_\_, x_n) \leftarrow \phi(x_1, \_\_, x_n)$
2. for $i \leftarrow 1$ to n do
3. $\quad \psi' \leftarrow \psi(x_2, \_\_, x_{i-1}, 0, x_{i+1}, \_\_, x_n)$
4. $\quad \psi'' \leftarrow \psi(x_1, \_\_, x_{i-1}, 1, x_{i+1}, \_\_, x_n)$
5. $\quad$ if (SATYSFIABLE(ψ')) then → sub routine che risolve correttamente un'istanza di SAT fino a n variabili
6. $\quad\quad a_i \leftarrow 0$
7. $\quad\quad \psi \leftarrow \psi(x_1, \_\_, x_{i-1}, a_i, x_{i+1}, \_\_, x_n)$
8. $\quad$ elseif (SATYSFIABLE(ψ'')) then
9. $\quad\quad a_i \leftarrow 1$
10. $\quad\quad \psi \leftarrow \psi(x_1, \_\_, x_{i-1}, a_i, x_{i+1}, \_\_, x_n)$
11. $\quad$ else
12. $\quad\quad$ return IMPOSSIBLE
13. $\quad$ if $\phi(a_1, \_\_, a_n) = 1$ then Questo if verifica che la sub-routine non sia errata
14. $\quad\quad$ return $(a_1, \_\_, a_n)$
15. $\quad$ else
16. $\quad\quad$ return IMPOSSIBLE

valori alle variabili $x_2, x_3, \_\_, x_n$ che soddisfa ϕ' e produci in output $x_1 = 0$ insieme all'assegnamento effettuato per $x_2, \_\_, x_n$.

b) Se ϕ' non è soddisfacibile, allora fissa $x_1 = 1$ in ϕ ed ottieni la nuova formula ϕ''.

Se ϕ'' è soddisfacibile, cerca ricorsivamente un assegnamento a $x_2, x_3, \_\_, x_n$ che soddisfi ϕ'' e produci in output $x_1 = 1$ insieme all'assegnamento effettuato per $x_2, x_3, \_\_, x_n$.

c) se ϕ' e ϕ'' sono entrambe non soddisfacibili, allora ϕ non è soddisfacibile.

Questo algoritmo ha tempo polinomiale se l'algoritmo di decisione per SAT ha tempo polinomiale. L'algoritmo di decisione viene chiamato al più $2n$ volte.