The 3-*SAT* Decision Problem Exhaustive Search Implementations

Team Satisfaction

Christopher Wood, Eitan Romanoff, Ankur Bajoria

April 1, 2013

Agenda

1 Problem Statement

2 Exhaustive Search Algorithm

3 Sequential Program Demo

Boolean Satisfiability

Boolean satisfiability is an *NP*-complete decision problem defined as:

$$SAT : \phi \rightarrow \{YES, NO\}$$

Input: Boolean formula ϕ on n variables.

Output: *YES* if there exists a variable truth assignment to the variables in ϕ such that it evaluates to true, *NO* otherwise.

$$\phi \text{ is satisfiable} \Leftrightarrow SAT(\phi) = YES$$

3-*SAT* $\in$ *NP*

- A special case of *SAT* that fixes the format of ϕ.
- Each input formula is in 3-*CNF* form:
  - The conjunction (Boolean AND) of arbitrarily many clauses, where each clause is the disjunction (Boolean OR) of exactly three literals (a Boolean variable or its negation).

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$

- *SAT* reduces to 3-*SAT*, so 3-*SAT* $\in$ *NP*.

Exhaustive Search for 3-*SAT*

Input: 3-*CNF* formula ϕ_n on n variables, **Output:** YES or NO

$C \leftarrow FALSE^n$ (vector of n False values, the initial configuration)
for $i = 0 \rightarrow 2^n - 1$ **do**
    $SAT \leftarrow TRUE$
    for all $clause \in \phi_n$ **do**
        if $evaluate(clause, C) = FALSE$ **then**
            $SAT \leftarrow FALSE$
        end if
    end for
    if $SAT = TRUE$ **then return** YES
    $C \leftarrow nextConfig(C)$
end for
return NO

Evaluating a Clause

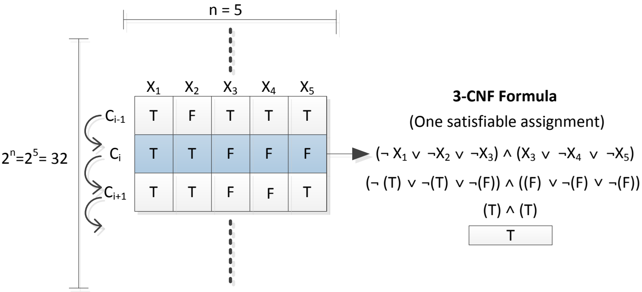
- Evaluating a clause depends on how ϕ_n and the variable truth assignments are stored.
  - `boolean[]` for variable assignments and `Literal[][3]` for ϕ_n.
- A literal is true (meaning the clause is true), if one of the following hold:
  - The literal is negated and its assignment is false.
  - The literal is not negated and its assignment is true.
- Two (fast) lookups and Boolean operations.

Exhaustive Search for 3-*SAT* - A *Very Satisfiable* Example!

Input: $\phi_5 = (\neg X_1 \vee \neg X_2 \vee \neg X_3) \wedge (X_3 \vee \neg X_4 \vee \neg X_5)$

Output: Yes

Note: no early termination once a satisfiable solution is found



The Sequential Solver

Demo time!

$$\phi_5 = (\neg X_1 \vee \neg X_2 \vee \neg X_3) \wedge (X_3 \vee \neg X_4 \vee \neg X_5)$$