The 3-*SAT* Decision Problem
Exhaustive Search Implementations

Team Satisfaction

Christopher Wood, Eitan Romanoff, Ankur Bajoria

March 26, 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 that will cause it to evaluate 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 literal is 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

1: $C \leftarrow 0^n$ (vector of n 0, the initial configuration)
2: **for** $r = 0 \rightarrow 2^n - 1$ **do**
3: $SAT \leftarrow TRUE$
4: **for all** $clause \in \phi_n$ **do**
5: **if** $evaluate(clause, C) = FALSE$ **then**
6: $SAT \leftarrow FALSE$
7: **end if**
8: **end for**
9: **if** $SAT = TRUE$ **then return** YES
10: $C \leftarrow nextConfig(C)$
11: **end for**
12: **return** NO

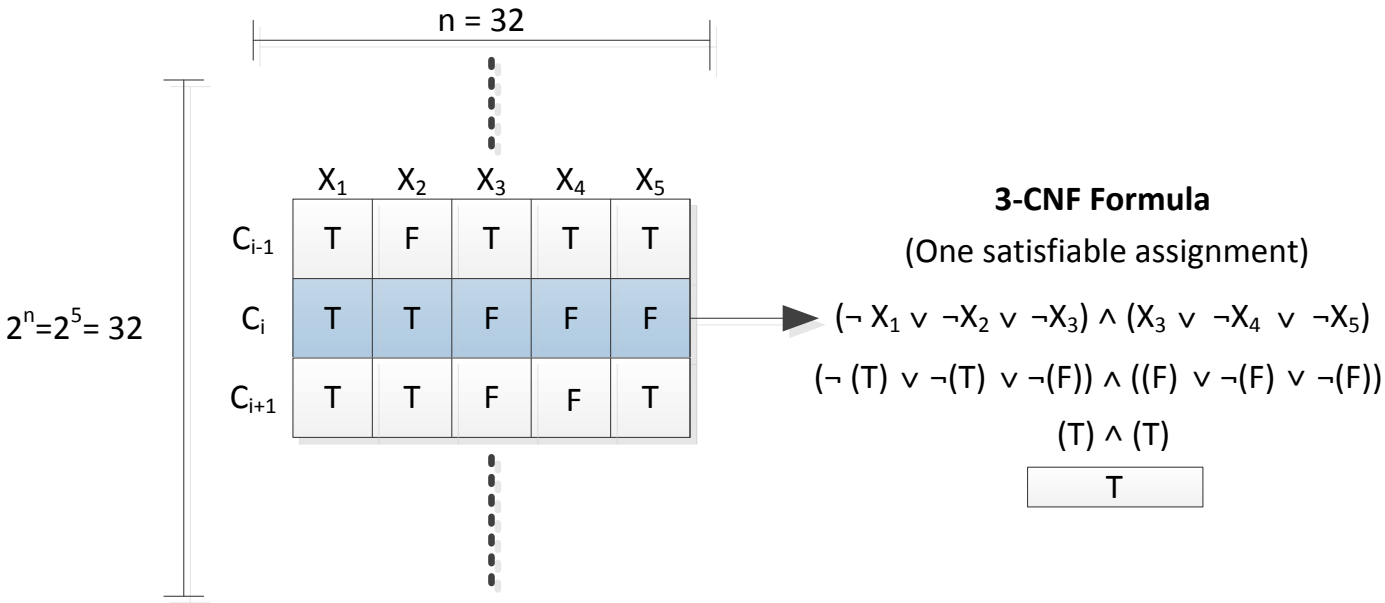
Algorithm 1: Exhaustive search for 3-*SAT*.

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!

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