Université Paris 5 – René Descartes
LIPADE

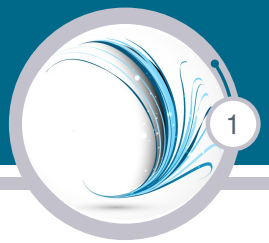
Algorithmic Complexity

Project: SAT Solver

Jean-Guy Mailly (jean-guy.mailly@parisdescartes.fr)

2018

Outline



SAT Problem

General definition:

- Given φ a propositional formula, is φ satisfiable?
  i.e. has φ at least one model?

Usual definition:

- Given φ a CNF formula, is φ satisfiable?

Interpretations of a Formula

Interpretation

- An interpretation is a mapping from Boolean variables to values in $\{0, 1\}$

Interpretations of a Formula

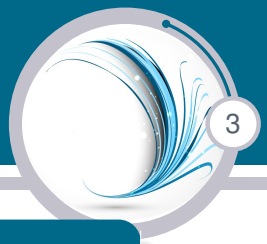
Interpretation

- An interpretation is a mapping from Boolean variables to values in $\{0, 1\}$

Sets of Atoms

- An interpretation can be written as the set of atoms which are assigned 1
- Example: $V = \{x_1, x_2, x_3\}$, $\omega(x_1) = 0, \omega(x_2) = \omega(x_3) = 1$ is equivalent to $\omega = \{x_2, x_3\}$

Interpretations of a Formula



Interpretation

- An interpretation is a mapping from Boolean variables to values in $\{0, 1\}$

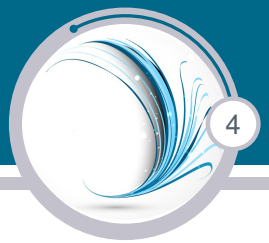
Sets of Atoms

- An interpretation can be written as the set of atoms which are assigned 1
- Example: $V = \{x_1, x_2, x_3\}, \omega(x_1) = 0, \omega(x_2) = \omega(x_3) = 1$ is equivalent to $\omega = \{x_2, x_3\}$

Vectors of Bits

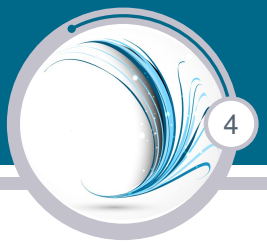
- An interpretation can be written as a vector of 0 and 1; the order has to be fixed (usually lexicographical order)
- Example: $V = \{x_1, x_2, x_3\}, \omega(x_1) = 0, \omega(x_2) = \omega(x_3) = 1$ is equivalent to $\omega = 011$

Conjunctive Normal Form



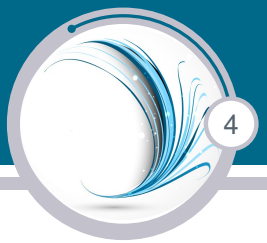
- A formula is a **literal** iff it is an atom or the negation of an atom
  x_1 and $\neg x_1$ are literals; $x_1 \vee x_2$ is not a literal

Conjunctive Normal Form



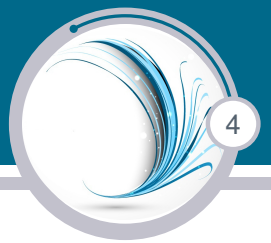
- A formula is a **literal** iff it is an atom or the negation of an atom
  x_1 and $\neg x_1$ are literals; $x_1 \vee x_2$ is not a literal
- A formula is a **clause** iff it is a disjunction of literals
  $x_1 \vee x_2 \vee \neg x_3$ is a clause ; $x_1 \vee (x_2 \wedge x_3)$ is not a clause

Conjunctive Normal Form



- A formula is a **literal** iff it is an atom or the negation of an atom
  x_1 and $\neg x_1$ are literals; $x_1 \vee x_2$ is not a literal
- A formula is a **clause** iff it is a disjunction of literals
  $x_1 \vee x_2 \vee \neg x_3$ is a clause ; $x_1 \vee (x_2 \wedge x_3)$ is not a clause

Conjunctive Normal Form

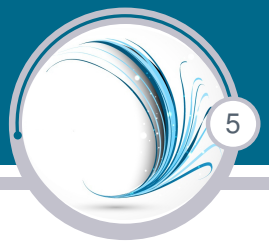


- A formula is a **literal** iff it is an atom or the negation of an atom
  x_1 and $\neg x_1$ are literals; $x_1 \vee x_2$ is not a literal
- A formula is a **clause** iff it is a disjunction of literals
  $x_1 \vee x_2 \vee \neg x_3$ is a clause ; $x_1 \vee (x_2 \wedge x_3)$ is not a clause

Conjunctive Normal Form

A formula is in Conjunctive Normal Form (CNF) iff it is a conjunction of clauses

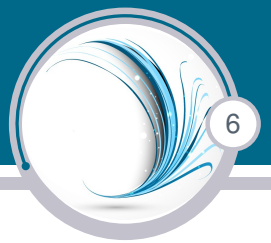
Conjunctive Normal Form



Examples of CNF formulas

- $(x_1 \vee x_2) \wedge (x_3 \vee x_4)$
- $(x_1 \vee \neg x_2 \vee x_5) \wedge (x_3 \vee x_5)$
- $(x_1 \vee x_3) \wedge (\neg x_2 \vee x_5) \wedge x_4$

Semantics of CNF Formulas



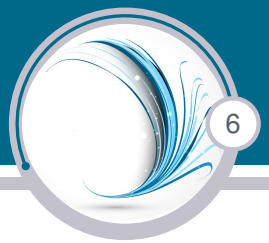
- A positive literal (atom x) is satisfied by ω if $\omega(x) = 1$
- A negative literal ($\neg x$) is satisfied by ω if $\omega(x) = 0$
- A clause is satisfied by ω if at least one of its literals is satisfied by ω
- A CNF formula is satisfied by ω if all its clauses are satisfied by ω

Example

With $\omega(x_1) = \omega(x_2) = 1$ and $\omega(x_3) = \omega(x_4) = 0$

- $(x_1 \vee x_2) \wedge (x_3 \vee x_4)$ is not satisfied

Semantics of CNF Formulas



- A positive literal (atom x) is satisfied by ω if $\omega(x) = 1$
- A negative literal ($\neg x$) is satisfied by ω if $\omega(x) = 0$
- A clause is satisfied by ω if at least one of its literals is satisfied by ω
- A CNF formula is satisfied by ω if all its clauses are satisfied by ω

Example

With $\omega(x_1) = \omega(x_2) = 1$ and $\omega(x_3) = \omega(x_4) = 0$

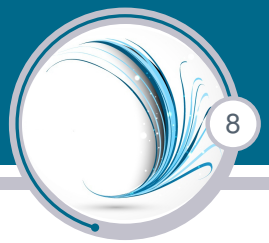
- $(x_1 \vee x_2) \wedge (x_3 \vee x_4)$ is not satisfied

With $\omega(x_1) = \omega(x_3) = 1$ and $\omega(x_2) = \omega(x_4) = 0$

- $(x_1 \vee x_2) \wedge (x_3 \vee x_4)$ is satisfied

Outline

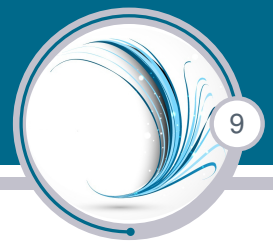
SAT Solver



A SAT Solver is a program that

- reads a CNF formula from a file in Dimacs format (extension `.cnf`), a special format used to represent CNF formulas
- determines if the formula is satisfiable
  1. if it is satisfiable, the solver prints the message `SATISFIABLE` and a model of the formula
  2. otherwise, the solver prints the message `UNSATISFIABLE`

Dimacs Format



Format definition:

```
p cnf nbVar nbClauses
first clause 0
second clause 0
etc 0
```

- The Boolean variables are represented by integers > 0
- An atom x is represented by $i > 0$, and the negative literal $\neg x$ is represented by $-i$
- Example:

```
p cnf 5 3
1 -5 4 0
-1 5 3 4 0
-3 -4 0
```

Outline

General Instructions

- The detailled explanations will be given on Moodle, on Friday October 5^{th}
- The solver must be implemented in Java, C or C++
- The solver must be compilable and executable on Ubuntu (18.04 Bionic Beaver) or MacOS (10.14 Mojave)
- The solver must be implemented by a team of 2 students
- Test cases will be provided sonn (on Moodle)
- Deadline: Sunday, Decembre 2^{nd}, 23:59 (Paris time), on Moodle

What must be delivered?

- The whole source code
- A shell script `build.sh` that does not have parameters, and that compile your program into an executable file or an executable Jar file
- A short report (`.pdf` file) that describes:
  - The features of the solver (especially, if you find that some test cases do not work)
  - The compilation and execution environment
  - Any dependency (external library or software)
  - The (short) description of your code (the role of the main classes and functions)
  - If you decide to use any external techniques, describe them and cite their references

How to deliver?

- Moodle
- The whole project must be located in a directory named from the students (e.g. `JohnDoe_JaneDoe`):
  - The `.pdf` report
  - The source code directory
  - The `build.sh` file
  - Any test case that you decide to use (in a separate directory `test`)
- Archive format for delivery: `.zip` or `.tar.gz`
- **There will be penalties for any violation of the instructions** (late delivery, other programming language, no citation of external sources,. . .)