Python Programming

SAT Solver, Propositional Logic

Rathindra Nath Dutta

Senior Research Fellow
Advanced Computing & Microelectronics Unit
Indian Statistical Institute, Kolkata

November 08, 2023

SAT Solver

- An ‘algorithm’ for establishing satisfiability of a Boolean formula

SAT Solver

- An 'algorithm' for establishing satisfiability of a Boolean formula
- Input: A Boolean logic formula (typically in CNF)
  Output: True/False

SAT Solver

- An ‘algorithm’ for establishing satisfiability of a Boolean formula
- Input: A Boolean logic formula (typically in CNF)
  Output: True/False
- For a satisfiable formula it also provides a *satisfiable assignment* of the Boolean variables

SAT Solver

- An ‘algorithm’ for establishing satisfiability of a Boolean formula
- Input: A Boolean logic formula (typically in CNF)
  Output: True/False
- For a satisfiable formula it also provides a *satisfiable assignment* of the Boolean variables
- There are many libraries available, we will focus on **PySAT**[1]

[1] https://pysathq.github.io/

SAT Solver

- An 'algorithm' for establishing satisfiability of a Boolean formula
- Input: A Boolean logic formula (typically in CNF)
  Output: True/False
- For a satisfiable formula it also provides a *satisfiable assignment* of the Boolean variables
- There are many libraries available, we will focus on **PySAT**[1]
- Installation:
  `pip install python-sat`

[1] https://pysathq.github.io/

Using PySAT

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

Using PySAT

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

Syntax: A literal in PySAT is assumed to be an integer
Example: 2 would represent literal x_2, while -1 represents $\neg x_1$

Using PySAT

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

Syntax: A literal in PySAT is assumed to be an integer
Example: 2 would represent literal x_2, while -1 represents $\neg x_1$

```
from pysat.solvers import Solver

formula = Solver()
formula.add_clause([1, 2])
formula.add_clause([-2, 3])

if formula.solve() == True:
    print(formula.get_model())
```

Using PySAT

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

Syntax: A literal in PySAT is assumed to be an integer
Example: 2 would represent literal x_2, while -1 represents $\neg x_1$

```
from pysat.solvers import Solver

formula = Solver()
formula.add_clause([1, 2])
formula.add_clause([-2, 3])

if formula.solve() == True:
    print(formula.get_model())
```

Output: `[1, -2, -3]` $\implies x_1 = True, x_2 = False, x_3 = False$

Extracting All Solutions

- After getting a solution, simply 'ban' it

Extracting All Solutions

- After getting a solution, simply ‘ban’ it
- How to ban a solution?

Extracting All Solutions

- After getting a solution, simply ‘ban’ it
- How to ban a solution? – add a reverse clause

  To ban the solution $x_1 = True$, $x_2 = False$, $x_3 = False$
  We need the constraint $\neg(x_1 \wedge \neg x_2 \wedge \neg x_3)$
  Thus we would add the clause $(\neg x_1 \vee x_2 \vee x_3)$

```
...
while formula.solve():
    solution = formula.get_model()
    print(solution)
    ban_clause = [-literal for literal in solution]
    formula.add_clause(ban_clause)
```

Some Useful Tips

- $P \to Q$ holds iff $\neg P \vee Q$ is a tautology

  i.e., $P \wedge \neg Q$ is unsatisfiable

Some Useful Tips

- $P \to Q$ holds iff $\neg P \vee Q$ is a tautology
  i.e., $P \wedge \neg Q$ is unsatisfiable

- Solvers requires the input formula in CNF
  Use De Morgan's laws wherever needed

Some Useful Tips

- $P \to Q$ holds iff $\neg P \vee Q$ is a tautology
  i.e., $P \wedge \neg Q$ is unsatisfiable

- Solvers requires the input formula in CNF
  Use De Morgan's laws wherever needed

- Be creative with the variables

Example: Unicorn Puzzle

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Example: Unicorn Puzzle

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Encoding propositions:

Example: Unicorn Puzzle

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Encoding propositions:

- If the unicorn is mythical, then it is immortal

Example: Unicorn Puzzle

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Encoding propositions:

- If the unicorn is mythical, then it is immortal
$$M \to I$$

Example: Unicorn Puzzle

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Encoding propositions:

- If the unicorn is mythical, then it is immortal
$$M \to I$$
- If the unicorn is not mythical, then it is a mortal mammal

Example: Unicorn Puzzle

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Encoding propositions:

- If the unicorn is mythical, then it is immortal
$$M \to I$$
- If the unicorn is not mythical, then it is a mortal mammal
$$\neg M \to (\neg I \land A)$$

Example: Unicorn Puzzle

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Encoding propositions:

- If the unicorn is mythical, then it is immortal
$$M \to I$$
- If the unicorn is not mythical, then it is a mortal mammal
$$\neg M \to (\neg I \land A)$$
- If the unicorn is either immortal or a mammal, then it is horned

Example: Unicorn Puzzle

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Encoding propositions:

- If the unicorn is mythical, then it is immortal
$$M \to I$$
- If the unicorn is not mythical, then it is a mortal mammal
$$\neg M \to (\neg I \land A)$$
- If the unicorn is either immortal or a mammal, then it is horned
$$(I \lor A) \to H$$

Example: Unicorn Puzzle

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Encoding propositions:

- If the unicorn is mythical, then it is immortal
$$M \to I$$
- If the unicorn is not mythical, then it is a mortal mammal
$$\neg M \to (\neg I \land A)$$
- If the unicorn is either immortal or a mammal, then it is horned
$$(I \lor A) \to H$$
- The unicorn is magical if it is horned

Example: Unicorn Puzzle

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Encoding propositions:

- If the unicorn is mythical, then it is immortal
$$M \to I$$
- If the unicorn is not mythical, then it is a mortal mammal
$$\neg M \to (\neg I \land A)$$
- If the unicorn is either immortal or a mammal, then it is horned
$$(I \lor A) \to H$$
- The unicorn is magical if it is horned
$$H \to G$$

Example: Unicorn Puzzle (contd.)

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Let $\Gamma = \{M \to I, \ \ \neg M \to (\neg I \wedge A), \ \ (I \vee A) \to H, \ \ H \to G\}$

Example: Unicorn Puzzle (contd.)

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Let $\Gamma = \{M \to I, \;\; \neg M \to (\neg I \wedge A), \;\; (I \vee A) \to H, \;\; H \to G\}$

Given Γ, answer the followings:

Example: Unicorn Puzzle (contd.)

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Let $\Gamma = \{M \to I, \;\; \neg M \to (\neg I \wedge A), \;\; (I \vee A) \to H, \;\; H \to G\}$

Given Γ, answer the followings:

- Is the unicorn magical?

Example: Unicorn Puzzle (contd.)

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Let $\Gamma = \{M \to I,\ \ \neg M \to (\neg I \wedge A),\ \ (I \vee A) \to H,\ \ H \to G\}$

Given Γ, answer the followings:

- Is the unicorn magical? $\equiv\ \ \Gamma \to G$?

Example: Unicorn Puzzle (contd.)

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Let $\Gamma = \{M \to I, \;\; \neg M \to (\neg I \wedge A), \;\; (I \vee A) \to H, \;\; H \to G\}$

Given Γ, answer the followings:

- Is the unicorn magical? $\equiv \;\; \Gamma \to G$? Test if $\Gamma \wedge \neg G$ is unsatisfiable

Example: Unicorn Puzzle (contd.)

Consider the following variables:

- M: The unicorn is Mythical
- I: The unicorn is Immortal
- A: The unicorn is mAmmal
- H: The unicorn is Horned
- G: The unicorn is maGical

Let $\Gamma = \{M \rightarrow I, \;\; \neg M \rightarrow (\neg I \wedge A), \;\; (I \vee A) \rightarrow H, \;\; H \rightarrow G\}$

Given Γ, answer the followings:

- Is the unicorn magical? $\equiv$ $\Gamma \rightarrow G$? Test if $\Gamma \wedge \neg G$ is unsatisfiable
- Is the unicorn horned? $\equiv$ $\Gamma \rightarrow H$? Test if $\Gamma \wedge \neg H$ is unsatisfiable
- Is it mythical? $\equiv$ $\Gamma \rightarrow M$? Test if $\Gamma \wedge \neg M$ is unsatisfiable

Example: Scheduling AI Class using AI

Find all weekdays for scheduling the AI lab class, given the followings:

- Rathin can not take class on Friday
- Ansuman sir insists for classes on Tuesday, Wednesday or Friday
- Students do not want classes on Monday and Thursday

Example: Scheduling AI Class using AI

Find all weekdays for scheduling the AI lab class, given the followings:

- Rathin can not take class on Friday
- Ansuman sir insists for classes on Tuesday, Wednesday or Friday
- Students do not want classes on Monday and Thursday

Solution:

- Take five variables for the weekdays: Mon, Tue, Wed, Thu, Fri

Example: Scheduling AI Class using AI

Find all weekdays for scheduling the AI lab class, given the followings:

- Rathin can not take class on Friday
- Ansuman sir insists for classes on Tuesday, Wednesday or Friday
- Students do not want classes on Monday and Thursday

Solution:

- Take five variables for the weekdays: Mon, Tue, Wed, Thu, Fri
- Write clauses for each of the three facts:

$$(\neg Fri) \quad \wedge \quad (Tue \vee Wed \vee Fri) \quad \wedge \quad (\neg Mon \wedge \neg Thu)$$

Example: Scheduling AI Class using AI

Find all weekdays for scheduling the AI lab class, given the followings:

- Rathin can not take class on Friday
- Ansuman sir insists for classes on Tuesday, Wednesday or Friday
- Students do not want classes on Monday and Thursday

Solution:

- Take five variables for the weekdays: Mon, Tue, Wed, Thu, Fri
- Write clauses for each of the three facts:
  $$(\neg Fri) \quad \wedge \quad (Tue \vee Wed \vee Fri) \quad \wedge \quad (\neg Mon \wedge \neg Thu)$$
- The last clause is not a disjunctive one, split it into two clauses
  $$(\neg Fri) \quad \wedge \quad (Tue \vee Wed \vee Fri) \quad \wedge \quad (\neg Mon) \wedge (\neg Thu)$$

Example: Scheduling AI Class using AI

Find all weekdays for scheduling the AI lab class, given the followings:

- Rathin can not take class on Friday
- Ansuman sir insists for classes on Tuesday, Wednesday or Friday
- Students do not want classes on Monday and Thursday

Solution:

- Take five variables for the weekdays: Mon, Tue, Wed, Thu, Fri
- Write clauses for each of the three facts:
  $$(\neg Fri) \quad \wedge \quad (Tue \vee Wed \vee Fri) \quad \wedge \quad (\neg Mon \wedge \neg Thu)$$
- The last clause is not a disjunctive one, split it into two clauses
  $$(\neg Fri) \quad \wedge \quad (Tue \vee Wed \vee Fri) \quad \wedge \quad (\neg Mon) \wedge (\neg Thu)$$
- Find all truth assignments for this formula

Example: Graph Coloring

Given a graph $G = (V, E)$, can its vertices be colored with only K many colors so that no two adjacent vertices get the same color – decision problem

See https://haslab.github.io/MFES/2122/PL+SAT-handout.pdf for more

Example: Graph Coloring

Given a graph $G = (V, E)$, can its vertices be colored with only K many colors so that no two adjacent vertices get the same color – decision problem

- Create $|V| \times K$ many Boolean variables

$$X_{v,c} = \begin{cases} 1 & \text{if vertex } v \in V \text{ is assigned the color } c \in \mathbb{N}_K \\ 0 & \text{otherwise} \end{cases}$$

See https://haslab.github.io/MFES/2122/PL+SAT-handout.pdf for more

Example: Graph Coloring

Given a graph $G = (V, E)$, can its vertices be colored with only K many colors so that no two adjacent vertices get the same color – decision problem

- Create $|V| \times K$ many Boolean variables

$$X_{v,c} = \begin{cases} 1 & \text{if vertex } v \in V \text{ is assigned the color } c \in \mathbb{N}_K \\ 0 & \text{otherwise} \end{cases}$$

- Each vertex must get some color
  $\forall v \in V$ add the clause $(X_{v,1} \vee X_{v,2} \vee \cdots \vee X_{v,K})$

See https://haslab.github.io/MFES/2122/PL+SAT-handout.pdf for more

Example: Graph Coloring

Given a graph $G = (V, E)$, can its vertices be colored with only K many colors so that no two adjacent vertices get the same color – decision problem

- Create $|V| \times K$ many Boolean variables

$$X_{v,c} = \begin{cases} 1 & \text{if vertex } v \in V \text{ is assigned the color } c \in \mathbb{N}_K \\ 0 & \text{otherwise} \end{cases}$$

- Each vertex must get some color
  $\forall v \in V$ add the clause $(X_{v,1} \vee X_{v,2} \vee \cdots \vee X_{v,K})$
- Adjacent vertices must not get the same color
  $\forall (u, v) \in E$ and $\forall c \in \mathbb{N}_K$ add the clause $(X_{u,c} \rightarrow \neg X_{v,c})$

See https://haslab.github.io/MFES/2122/PL+SAT-handout.pdf for more

Example: Graph Coloring

Given a graph $G = (V, E)$, can its vertices be colored with only K many colors so that no two adjacent vertices get the same color – decision problem

- Create $|V| \times K$ many Boolean variables

$$X_{v,c} = \begin{cases} 1 & \text{if vertex } v \in V \text{ is assigned the color } c \in \mathbb{N}_K \\ 0 & \text{otherwise} \end{cases}$$

- Each vertex must get some color
  $\forall v \in V$ add the clause $(X_{v,1} \vee X_{v,2} \vee \cdots \vee X_{v,K})$
- Adjacent vertices must not get the same color
  $\forall (u, v) \in E$ and $\forall c \in \mathbb{N}_K$ add the clause $(X_{u,c} \rightarrow \neg X_{v,c})$
- Each vertex must not get more that once color (optional)
  $\forall v \in V$ and $\forall c \in \mathbb{N}_K$ add the clause $(X_{u,c} \rightarrow \bigwedge_{c' \in \mathbb{N}, c' \neq c} \neg X_{v,c'})$

See https://haslab.github.io/MFES/2122/PL+SAT-handout.pdf for more