Logic for Computer Science. Knowledge Representation and Reasoning.

Lecture Notes

for

Computer Science Students

Faculty EAIiIB-IEiT AGH

Antoni Ligęza

Other support material:

http://home.agh.edu.pl/~ligeza

https://ai.ia.agh.edu.pl/pl:dydaktyka:logic:start#logic_for_computer_science2020

Satifiability: In Search for Models. Decision Trees, OBDD, SAT

- An Example Problem,
- The SAT Problem – what is behind the purely logical statement,
- Approaches to SAT - search for models,
- Decision Tress,
- Reduced Decision Trees,
- Inference: applying the Resolution Rule,
- The Shannon Expansion Rule
- Ordered Binary Decision Diagrams (OBDD),
- The Unicorn and SAT again,
- The DPLL Algorithm,
- From CNF to DIMACS format and MiniSat, Picosat, etc.
- SAT as Constraint Programming: an application of Prolog + clp(fd) library

A Problem to Start: Tracking the Murderer

Some knowledge specification — in natural language:

- If Sarah was drunk then either James is the murderer or Sarah lies,
- Either James is the murderer or Sarah was not drunk and the crime took place after midnight,
- If the crime took place after midnight then either James is the murderer or Sarah lies,
- Sarah does not lie when sober.

Introduction symbols and transformation to formal specification:

- A = James is the murderer,
- B = Sarah is drunk,
- C = Sarah lies,
- D = The murder took place after midnight.

$$B \Longrightarrow A \vee C$$

$$A \vee (\neg B \wedge D)$$

$$D \Longrightarrow (A \vee C)$$

$$C \Longrightarrow B$$

Questions:

Who is the murderer? Which facts are true/false? Is the system consistent? How many models does it have (if consistent)? What are the exact models?

In fact – the set of logical formulas can be considered as constraints and we are looking for models satisfying these constraints.

Logic for KRR – Tasks and Tools

- Theorem Proving – Verification of Logical Consequence:

$$\Delta \models H;$$

- Automated Inference – Derivation:

$$\Delta \vdash H;$$

- SAT (checking for models) – verification of satisfiability:

$$\models_I H;$$

  In fact, we search for solution(s) for set of constraints (logical constraints are analog of mathematical equations).

- un-SAT verification – unsatisfiability:

$$\not\models_I H \quad \text{for any interpretation I;}$$

- Tautology verification (completeness):

$$\models H$$

- valid inference rules – checking:

$$(\Delta \vdash H) \quad \longrightarrow \quad (\Delta \models H)$$

- complete inference rules – checking:

$$(\Delta \models H) \quad \longrightarrow \quad (\Delta \vdash H)$$

Unicorn - Logical Model

Definition of propositional variables:

- M: The unicorn is mythical
- I: The unicorn is immortal
- L: The unicorn is mammal
- H: The unicorn is horned
- G: The unicorn is magical

Building a **Logical Model** for the puzzle:

- If the unicorn is mythical, then it is immortal:

$$M \longrightarrow I$$

- If the unicorn is not mythical, then it is a mortal mammal:

$$\neg M \longrightarrow (\neg I \wedge L)$$

- If the unicorn is either immortal or a mammal, then it is horned:

$$(I \vee L) \longrightarrow H$$

- The unicorn is magical if it is horned:

$$H \longrightarrow G$$

Resulting Boolean formula (the Knowledge Base):

$$(M \longrightarrow I) \wedge (\neg M \longrightarrow (\neg I \wedge L)) \wedge ((I \vee L) \longrightarrow H) \wedge (H \longrightarrow G)$$

A Solution: Formal Derivation of Logical Consequences

1. $(M \longrightarrow I) \equiv (\neg M \vee I)$
2. $(\neg M \longrightarrow (\neg I \wedge L)) \equiv (M \vee (\neg I \wedge L))$
3. $(M \vee (\neg I \wedge L)) \equiv ((M \vee \neg I) \wedge (M \vee L))$
4. $\neg M \vee I, M \vee L$
5. $I \vee L$
6. $I \vee L, (I \vee L) \longrightarrow H$
7. H
8. $H, H \longrightarrow G$
9. G

So we have:

$$\mathrm{KB} \vdash H \wedge G$$

Questions:

- What about M (mythical), I (immortal) and L (mammal)?
- What are the exact models? What combinations are admissible?
- How many models do we have?
- What is the CNF of the original formula?
- What is the DNF of the original formula?
- Resolution, Dual Resolution, Semantic Tableau, Fitch System,...
  Try each one; which one you prefer?

Inference example

A – signal from process,

P – signal added to a queue,

B – signal blocked by process,

D – signal received by process,

S – state of the process saved,

M – signal mask read,

H – signal management procedure activated,

N – procedure executed in normal mode,

R – process restart from context,

I – process must re-create context.

Rules — axiomatization:

$A \longrightarrow P$,

$P \wedge \neg B \longrightarrow D$,

$D \longrightarrow S \wedge M \wedge H$,

$H \wedge N \longrightarrow R$,

$H \wedge \neg R \longrightarrow I$,

Facts:

A, $\neg B$, $\neg R$.

Conclusions

P, D, S, M, H, I, $\neg N$.

Try to draw an *AND/OR/NOT Graph*

How to represent:

- facts?
- implication?
- disjunctive conditions?
- conjunctive conditions?
- negation?
- constraints?

Examine Forward Chaining vs Backward Chaining!

Problem Solving – Satisfiability Verification – SAT

Definition 1 **Satisfiability** *Formula Ψ is satisfiable, iff there exists an interpretation I, such that Ψ is satsified with it:*

$$\models_I \Psi$$

Fundamental questions:

- SAT — is a given formula satisfiable?
- how many models — how many interpretations satisfy a formula?
- find a single/first model — a constructive task.
- find all models — much costly task.
- an alternative approach — prove unsatisfiability;
- in case of unsatisfiability: find maximal satisfied subsets.

Two alternative approchaes:

- analysis of possible interpretations — the zero-one methods; problem — combinatorial explosion;
- logical inference — derivation — with use of valid inference rules (e.g. the Resolution Rule) – try to reduce the problem.

Formula Evaluation - the 0/1 Approach

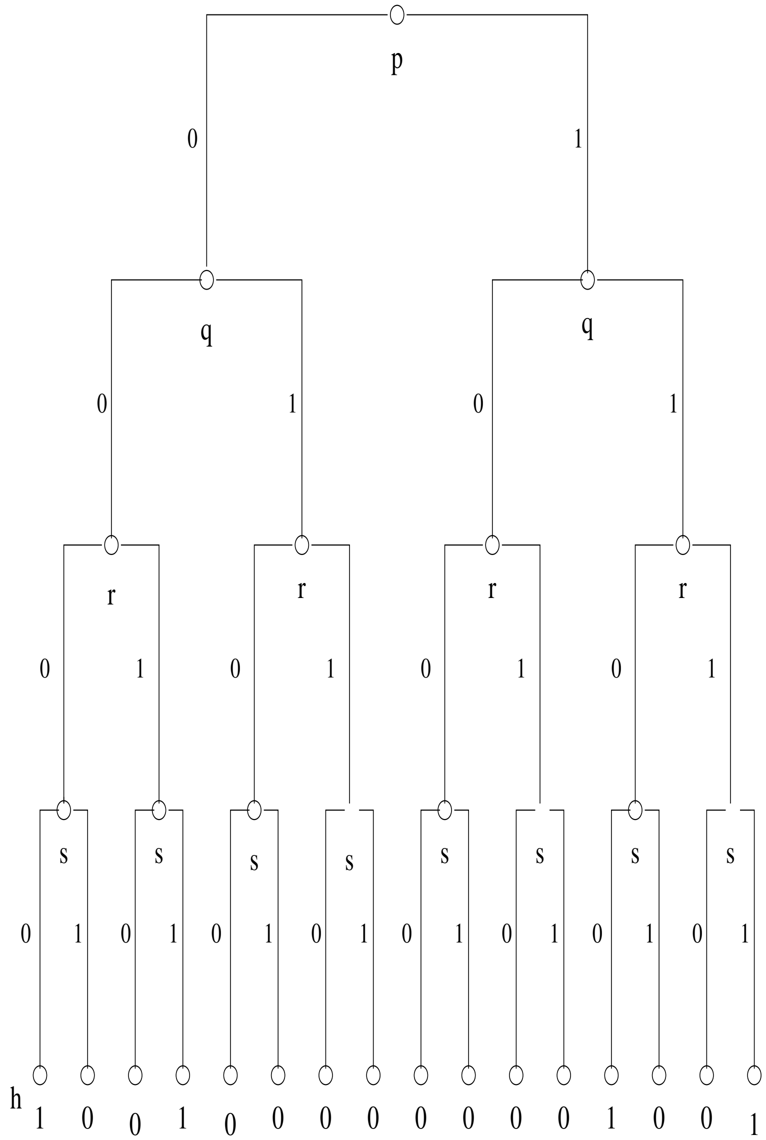
We check the satisfiability of an example formula:

$$h \equiv (p \Leftrightarrow q) \wedge (r \Leftrightarrow s)$$

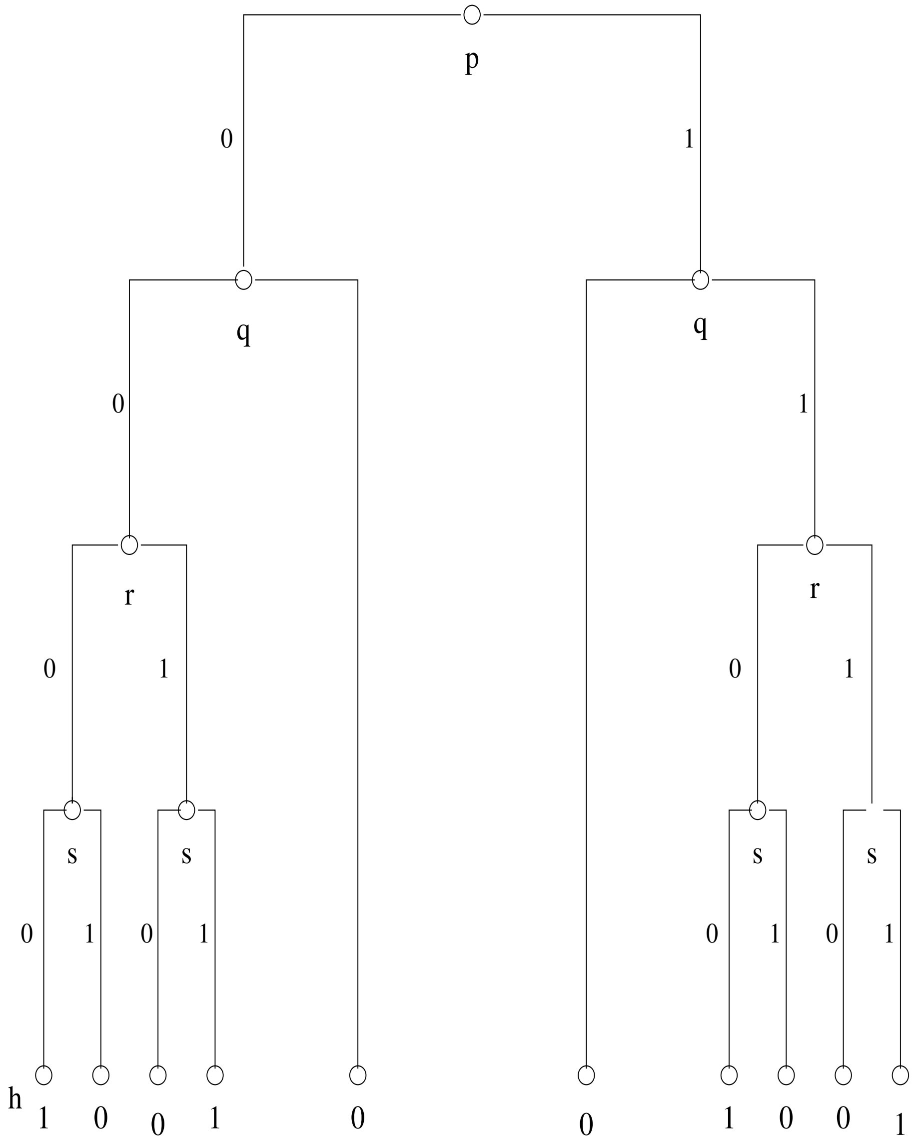
$RuleNo$	p	q	r	s	h
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

(1)

A Binary Tree - A more concise approach



Reduced Tree: a still better approach



SAT: Backtracking Search and Reduction

Example – in CNF:

$$\{p \vee q, p \vee \neg q, \neg p \vee q, \neg p \vee \neg q \vee \neg r, \neg p \vee r\}$$

The analysis can be performed with decision tree and backtracking search (DFS).

Example after reduction for $p = 1$:

$$\{q, \neg q \vee \neg r, r\}$$

Example after reduction for $p = 0$:

$$\{q, \neg q\}$$

Unit Propagation Rule: If q is a single literal in S, then one can remove q from S and apply reduction to all elements of S by replacing all occurences of q with 1 (for positive occurrence) and by 0 (for negative occurrence).

Ordered Binary Decision Diagrams (OBDD)

Key Notation:

$$p \longrightarrow h_0, h_1$$

and its meaning:

$$\textit{if } p \textit{ then } h_0 \textit{ else } h_1.$$

Definition 2 *The Shannon's Expansion Rule*

$$\phi \equiv p \longrightarrow \phi\{p/1\}, \phi\{p/0\},$$

Example:

$$p \wedge q \equiv p \longrightarrow q, 0,$$

$$p \vee q \equiv p \longrightarrow 1, q$$

$$\neg p \equiv p \longrightarrow 0, 1.$$

Formula Reduction

$$\phi = (p \Leftrightarrow q) \wedge (r \Leftrightarrow s).$$

$$\phi \equiv p \longrightarrow \phi_1, \phi_0 \quad (2)$$

$$\phi_1 \equiv q \longrightarrow \phi_{11}, 0 \quad (3)$$

$$\phi_0 \equiv q \longrightarrow 0, \phi_{00} \quad (4)$$

$$\phi_{11} \equiv r \longrightarrow \phi_{111}, \phi_{110} \quad (5)$$

$$\phi_{00} \equiv r \longrightarrow \phi_{001}, \phi_{000} \quad (6)$$

$$\phi_{111} \equiv s \longrightarrow 1, 0 \quad (7)$$

$$\phi_{110} \equiv s \longrightarrow 0, 1 \quad (8)$$

$$\phi_{001} \equiv s \longrightarrow 1, 0 \quad (9)$$

$$\phi_{000} \equiv s \longrightarrow 0, 1 \quad (10)$$

(11)

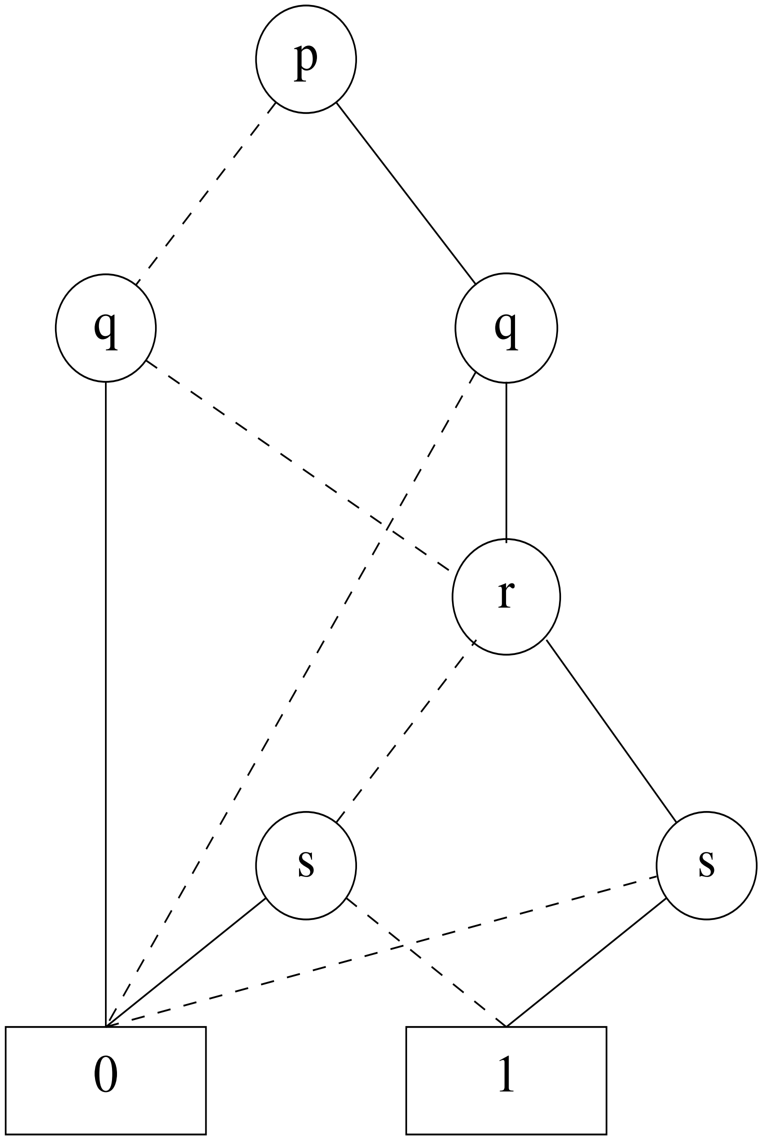
Reduction after detecting repeated subgraphs:

$$\phi \equiv p \longrightarrow \phi_1, \phi_0 \tag{12}$$
$$\phi_1 \equiv q \longrightarrow \phi_{11}, 0 \tag{13}$$
$$\phi_0 \equiv q \longrightarrow 0, \phi_{00} \tag{14}$$
$$\phi_{11} \equiv r \longrightarrow \phi_{111}, \phi_{110} \tag{15}$$
$$\phi_{00} \equiv r \longrightarrow \phi_{001}, \phi_{000} \tag{16}$$
$$\phi_{111} \equiv s \longrightarrow 1, 0 \tag{17}$$
$$\phi_{110} \equiv s \longrightarrow 0, 1 \tag{18}$$
$$\phi_{001} \equiv s \longrightarrow 1, 0 \tag{19}$$
$$\phi_{000} \equiv s \longrightarrow 0, 1 \tag{20}$$
$$\tag{21}$$

The final form:

$$\phi \equiv p \longrightarrow \phi_1, \phi_0 \tag{22}$$
$$\phi_1 \equiv q \longrightarrow \phi_{11}, 0 \tag{23}$$
$$\phi_0 \equiv q \longrightarrow 0, \phi_{11} \tag{24}$$
$$\phi_{11} \equiv r \longrightarrow \phi_{111}, \phi_{110} \tag{25}$$
$$\phi_{111} \equiv s \longrightarrow 1, 0 \tag{26}$$
$$\phi_{110} \equiv s \longrightarrow 0, 1 \tag{27}$$
$$\tag{28}$$

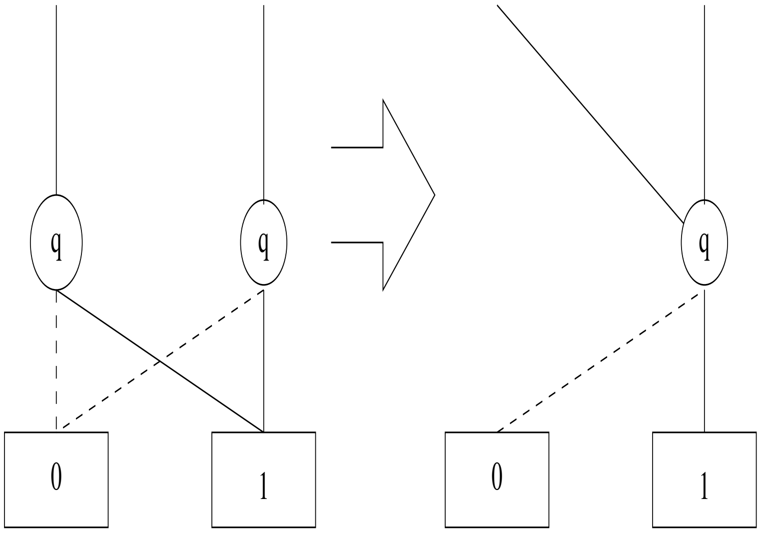
The Reduced OBDD (Ordered Binary Decision Diagram)



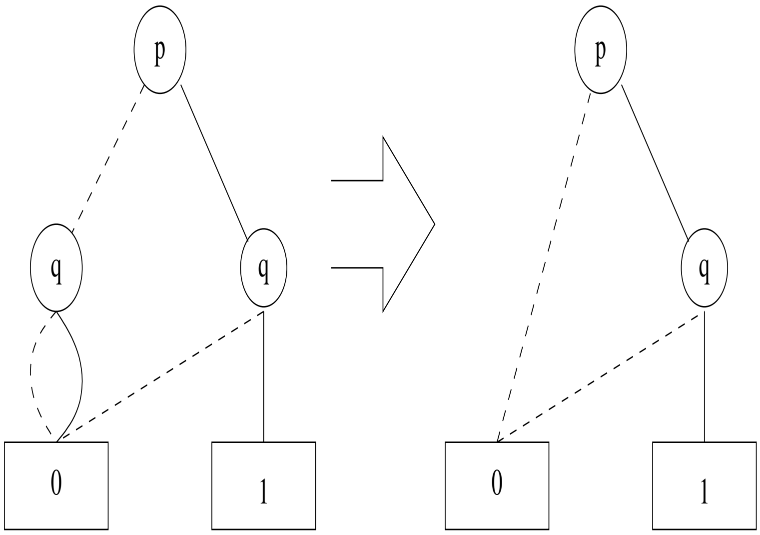
Applications of OBDD and its Analysis???

Reduction Methods

Reduction by Gluing:



Reduction by Elimination



SAT by Example: Unicorn

Given the following Knowledge Base (KB):

- If the unicorn is mythical, then it is immortal
- If the unicorn is not mythical, then it is a mortal mammal
- If the unicorn is either immortal or a mammal, then it is horned
- The unicorn is magical if it is horned

answer the following questions:

- Is the unicorn mythical? (M)
- Is it magical? (G)
- Is it horned? (H)

In terms of logic:

$$\text{KB} \models G, H, M$$

$$\text{KB} \vdash G, H, M$$

SAT by Example: Unicorn

Given the following Knowledge Base (KB):

- If the unicorn is mythical, then it is immortal
- If the unicorn is not mythical, then it is a mortal mammal
- If the unicorn is either immortal or a mammal, then it is horned
- The unicorn is magical if it is horned

answer the following questions:

- Is the unicorn mythical? (M)
- Is it magical? (G)
- Is it horned? (H)

In terms of logic:

$$\text{KB} \models G, H, M$$

$$\text{KB} \vdash G, H, M$$

Unicorn - Logical Model

Definition of propositional variables:

- M: The unicorn is mythical
- I: The unicorn is immortal
- L: The unicorn is mammal
- H: The unicorn is horned
- G: The unicorn is magical

Building a **Logical Model** for the puzzle:

- If the unicorn is mythical, then it is immortal:

$$M \longrightarrow I$$

- If the unicorn is not mythical, then it is a mortal mammal:

$$\neg M \longrightarrow (\neg I \wedge L)$$

- If the unicorn is either immortal or a mammal, then it is horned:

$$(I \vee L) \longrightarrow H$$

- The unicorn is magical if it is horned:

$$H \longrightarrow G$$

Resulting Boolean formula (the Knowledge Base):

$$(M \longrightarrow I) \wedge (\neg M \longrightarrow (\neg I \wedge L)) \wedge ((I \vee L) \longrightarrow H) \wedge (H \longrightarrow G)$$

Solving Unicorn by Hand

1. Apply the Resolution Rule,
2. Find which facts are necessarily true; (here: H and G).
3. Apply the pure/single literal strategy propagation for formula reduction.
4. Try to draw a decision tree covering all potential models,
5. prune (stop at) any branch where the model is unsatisfiable (at least one minterm of the CNF),
6. the remaining leafs specify the models.

Then read about the SAT problem: `https://en.wikipedia.org/wiki/Boolean_satisfiability_problem`.

Unicorn Example: Resulting CNF:

$$\{\neg M \vee I, M \vee \neg I, M \vee L, \neg I \vee H, \neg L \vee H, \neg H \vee G\}$$

But $H = 1$ and $G = 1$ (the so-called pure literals), so we have:

$$\{\neg M \vee I, M \vee \neg I, M \vee L\}$$

L=1 L=0

M=1 M=0 M=1

I=1 I=0 I=1

An outline of the DPLL Algorithm

```
  Algorithm DPLL
    Input: A set of clauses S.
    Output: A Truth Value.

function DPLL(S)
    if S is a consistent set of literals then
        return true;
    if S contains an empty clause (a false one) then
        return false;
    for every unit clause {l} in S do:
       S <-- unit-propagate(S);
    for every literal l that occurs pure in S do:
       S <-- pure-literal-assign(l, S);
    l <-- choose-literal(S);
    return DPLL(S & {l}) or DPLL(S & {not(l)});
```

For details see: `https://en.wikipedia.org/wiki/DPLL_algorithm`

CNF and Encoded File

Resulting CNF:

$$\{\neg M \vee I, M \vee \neg I, M \vee L, \neg I \vee H, \neg L \vee H, \neg H \vee G\}$$

We enumerate all 5 propositional symbols (how?). Each negative literal is denoted with the '-' sign preceding it. Each minterm is in one line. See below:

```
 M   I    L    H     G
 -----------------
-1   2
 1  -2
 1        3
    -2         4
          -3   4
              -4     5
```

This leads to a standard representation: the DIMACS format.

Input file in the DIMACS format:

```
p cnf 5 6
-1 2 0
1 -2 0
1 3 0
-2 4 0
-3 4 0
-4 5 0
```

Using Minisat Try the Minisat:

Page: `http://minisat.se/`

Online: `http://www.msoos.org/2013/09/minisat-in-your-browser/`

Manual: Page: `http://fmv.jku.at/picosat/`

How to get ALL solutions?

How to use Prolog for finding models? The SWI-Prolog + the clp(fd) library.

Extra problem – try to find a DIMACS representation...

Assumptions:

A1. There are 3 houses in a row

A2. The houses are numbered 1, 2 and 3, from left to right

A3. Each house has one of the colors Blue, Green or White

A4. Each house is inhabited by one person with one of the nationalities: Dutch, German and Italian

A5. Each person drinks (exactly one) of the following beverages: Coffee, Tea and Water

Conditions (constraints):

C1 The third house is green

C2 There is one house between the house of the person drinking coffee and the blue house

C3 The person drinking water lives in the blue house

C4 The Italian lives to the left of the coffee drinking person

C5 The German lives in house two

Query:

Who lives in the 1st house? What does the Dutch drink?