COL:750

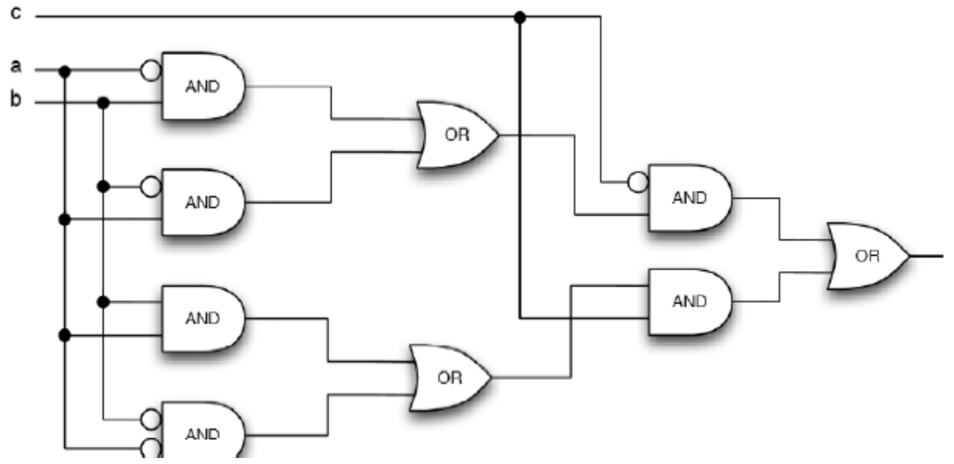
Foundations of Automatic Verification

Instructor: Priyanka Golia

Course Webpage

https://priyanka-golia.github.io/teaching/COL-750/index.html

```
PC1 (char[] SP, char[] UI) {
        for (int i=0; i<UI.length(); i++) {
                if(SP[i] != UI[i]) return No;
        }
        return Yes;
}
```



$\models$

System

Satisfies

Properties

S(I,O) $\models$ P(I,O)

Mathematical model of the system: specification of the property/problem:

- Boolean logic, First Order Logic (FOL), Linear Temporal Logic (LTL), Computational Tree Logic (CTL)
- Tools to check if the model satisfies the property.

Theorem: For any integers m and n, if m and n are odd, then m+n is even

Try to prove/disprove this theorem.

How do we formalize the definitions and reasoning we use in our proofs?

This week: Propositional Logic: reasoning about Boolean values
First Order Logic: reasoning about properties of multiple objects.

What is Logic?

A formal logic is defined by syntax and semantics.

Syntax:

- An alphabet of symbols.
- A finite sequence of these symbols is called expression
- A set of rules defines the well-formed expression.

Semantics:

- Gives meaning to well-formed expressions

Propositional Logic

$$TakeML \vee TakeFM$$

$$\neg FirstSucceed \rightarrow TryAgain$$

$$IsWinter \wedge IsSnow$$

Propositional Variables — TakeML, TryAgain, IsWinter,...

Each Proposition variables stands for a proposition, something that is either True or False

Propositional Connectives— ¬, ∨ , ∧
Links propositions into larger propositional

Propositional Logic: Syntax

(	Left parenthesis
)	Right parenthesis
$\neg$	Negation
$\wedge$	Or
$\vee$	And
$\rightarrow$	Condition
$\leftrightarrow$	Bi-Condition

Logical Symbols: The meaning of logical symbols is always the same.

P_1	Propositional variables
P_2	
P_n	

Non logical Symbols/Propositional Symbols: The meaning of nonlogical symbols depends on the context.

Propositional Logic: Syntax

Expression is a sequence of symbols.

$$(P_1 \wedge P_2), \quad ((\neg P_1) \vee P_2), \quad)) \leftrightarrow)P_1$$

We defined the set W of Well-Formed Fromulas (WFFs) as follows:

1. Every expression consists of a single proportional symbol is in W.
2. If α and β are in W, so are
   $(\neg \alpha), (\alpha \vee \beta), (\alpha \wedge \beta), (\alpha \rightarrow \beta), (\alpha \leftrightarrow \beta)$
3. No expression is in W unless forced by (1) and (2).

This definition is Inductive: the set being defined is used as part of definition.

Exercise-1: Propositional Logic

How would you use the definition of WFFs to prove that $)) \leftrightarrow)P$ is not a WFF?

Prove that any WFFs has the same number of left parentheses and right parentheses?

How do we parse the following:

$$\neg p \rightarrow q \vee r \rightarrow p \vee q \wedge z$$

Notational Conventions

- Larger variety of propositional symbols: $A, B, C, p_1, p_2, p, q, r, \alpha, \beta$
- Outermost parentheses can be omitted: $p \vee q$ instead of $(p \vee q)$
- Negation symbol binds stronger than binary connectives, and its scope is as small as possible:

$$\neg p \vee q \equiv ((\neg p) \vee q)$$

- $\{ \vee , \wedge \}$bind stronger than$\{ \rightarrow , \leftrightarrow \}$, for example:

$$p \wedge q \rightarrow \neg r \vee s \equiv ((p \wedge q) \rightarrow ((\neg r) \vee s))$$

- All operators are right-associative.

How do we parse the following:

$$\neg p \rightarrow q \vee r \rightarrow p \vee q \wedge z \equiv ((\neg p) \rightarrow ((q \vee r) \rightarrow (p \vee (q \wedge z))))$$

Propositional Logic: Semantics

Intuitively, given a WFF α and a value (either T or F) for each propositional symbol in α, we should be able to determine the value of α.

$F = ((p \vee q) \vee r)$

F is True

$p = 1, q = 0, r = 0$

F is called propositional Formula.

A mapping for assigning propositional variables to either 0 and 1, and evaluating F under that mapping.

Propositional Logic: Semantics

- τ is a function that maps proposition variables of a propositional formula to {0,1}.

$$F = ((p \vee q) \vee r)$$

$$\tau : \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$$

We call τ a truth assignment.

- How many such τ (truth assignments) can exists ? $2^{variables(F)}$

p	q	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

- τ satisfies formula F if and only if $F(\tau)$ is 1, such a τ is called satisfying assignment

$$F(\tau) : ((1 \vee 0) \vee 1) = 1$$

- We use $\tau \models F$ to represent.

Propositional Logic: Semantics

- If there exists a τ such that $\tau \vDash F$, we say that F is satisfiable.

$F = ((p \vee q) \vee r)$ $\quad \tau : \{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$ $\quad$ F is satisfiable

- If for all τ in $2^{variables(F)}$, $F(\tau)$ is 1, then F is valid.

Is $F = ((p \vee q) \vee r)$ is valid ? $\quad$ Is $F = (p \vee \neg p)$ is valid ?

- If there does not exists a τ in $2^{variables(F)}$ such that $F(\tau)$ is 1, then F is unsatisfiable.

Is $F = ((p \vee q) \vee r)$ is unsatisfiable? $\quad$ Is $F = (p \wedge \neg p)$ is unsatisfiable ?

Propositional Logic: Semantics

- Set of all satisfying assignment of F is called models. $models(F) = \{\tau \mid F(\tau) = 1\}$

$$Models(\neg F) = \{2^{variables}\} \setminus Models(F)$$

$$Models(F \vee G) = Models(F) \cup Models(G)$$

$$Models(F \wedge G) = Models(F) \cap Models(G)$$

- Equivalent formulas: Two formulas F and G are considered to be equivalent to

each other if and only if they both have same models, that is, if $Models(F) = Models(G), F \equiv G$.

Exercise-2: Propositional Logic

Determine whether the following formulas are satisfiable, unsatisfiable, or valid:

$(p \lor q) \land (\neg p \lor \neg q)$

$(p \lor q) \land (\neg p \lor \neg q) \land (p \leftrightarrow q)$

$\{p, p \to q\} \models q$

Given n proportional variables, how many Boolean functions $B(p_1, p_2, \ldots, p_n)$ can be generated?

Propositional Logic: Semantics

Suppose Σ is a set of WFFs, then $\Sigma \vDash \alpha$, if every truth assignment which satisfies each formula in Σ also satisfies α.

To check whether $\{\beta_1, \beta_2, \ldots, \beta_n\} \vDash \alpha$, check the satisfiability of $(\beta_1 \wedge \beta_2 \ldots \wedge \beta_n) \wedge (\neg \alpha)$.

If unsatisfiable, then $\{\beta_1, \beta_2, \ldots, \beta_n\} \vDash \alpha$.

Determining Satisfiability

To check whether α is satisfiable, form the truth table for α. If there is a row in which ***True*** appears as the value for α, then α is satisfiable. Otherwise, α is unsatisfiable.

What is the complexity of this algorithm?

2^n where n is the number of propositional symbols.

How to check the validity of a formula α?

If $\neg\alpha$ is unsatisfiable then α is valid.

If formula is SATisfiable, gives an satisfying assignment

Boolean /propositional formulas ——> SAT Solvers

UNSAT

Conjunction Normal Form (CNF)

- $F = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_3)$

Clauses

Literals : $x_1, \neg x_1, x_2, \neg x_2, x_3, \neg x_3$

CNF: $F = C_1 \wedge C_2 \wedge C_3 \ldots \wedge C_m$

where $C_i = (l_1 \vee l_2 \vee \ldots \vee l_k)$

where $l_j = p; l_j = \neg p$

Where p is propositional variable

SAT solvers takes

CNF formulas as input.

Can every formula F can be represented in CNF form, say F_{CNF}?

Can every formula F can be represented in CNF form, say F_{CNF}?

Yes, every F can be represented in F_{CNF}, such that $F \equiv F_{CNF}$

$F = ((x_1 \wedge \neg x_2) \vee (x_3 \wedge x_4))$ Can you convert F into F_{CNF}?

$F_{CNF} = (x_1 \vee x_3) \wedge (x_1 \vee x_4) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_2 \vee x_4)$

$F = ((x_1 \wedge \neg x_2) \vee (x_3 \wedge x_4)) \vee (x_5 \wedge x_6)$, Can you convert F into F_{CNF}?

$F = (x_1 \wedge y_1) \vee \ldots \vee (x_n \wedge y_n)$, size of equivalent F_{CNF}? 2^n

In the worst case, it may take exponential many steps. Can we do better?

Equisatisfiable Formulas

- $F = (p \lor \alpha) \land (\neg p \lor \beta)$ $\qquad$ $G = (\alpha \lor \beta)$

  F and G are Equisatisfiable. F is satisfiable if and only if G is satisfiable.

$F = ((x_1 \land \neg x_2) \lor (x_3 \land x_4))$ Can you convert F into F_{CNF}?

$= (t_1 \leftrightarrow (x_1 \land \neg x_2)) \land (t_2 \leftrightarrow (x_3 \lor x_4)) \land (t_1 \lor t_2)$

$= (\neg t_1 \lor (x_1 \land \neg x_2)) \land (\neg x_1 \lor x_2 \lor t_1) \land (\neg t_2 \lor (x_3 \land x_4)) \land (\neg x_3 \lor \neg x_4 \lor t_2) \land (t_1 \lor t_2)$

$= (\neg t_1 \lor x_1) \land (\neg t_1 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor t_1) \land (\neg t_2 \lor x_3) \land (\neg t_2 \lor x_4) \land (\neg x_3 \lor \neg x_4 \lor t_2) \land (t_1 \lor t_2)$

$= F_{CNF}$

$F = (x_1 \land y_1) \lor \ldots \lor (x_n \land y_n)$, size of equivalent F_{CNF}? $\quad 2n + n + 1$

Every formula F can be represented in CNF form, say F_{CNF} in polynomial time such that F is satisfiable if and only if F_{CNF} is satisfiable.

K-SAT

CNF: $F = C_1 \wedge C_2 \wedge C_3 \ldots \wedge C_m$

where $C_i = (l_1 \vee l_2 \vee \ldots \vee l_k)$

where $l_j = p; l_j = \neg p$

Where p is propositional variable

If $K = 2$, then $2 - SAT$. $F = (x_1 \vee \neg x_2) \wedge (x_3 \vee x_4)$

If $K = 3$, then $3 - SAT$. $F = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_3 \vee x_4)$

Exercise-3: Propositional Logic

Can you convert $4 - SAT$ formula into $3 - SAT$ formula?

Can you convert $3 - SAT$ formula into $2 - SAT$ formula?

Course Webpage

Thanks!