Propositional Logic – II

Madhavan Mukund

Chennai Mathematical Institute
http://www.cmi.ac.in/~madhavan

SAT-SMT School, TIFR
4 December 2016

Logical consequence

- Recall, logical consequence $X \models F$ — any assignment $\mathcal{A}$ that satisfies all of X also satisfies F
- Special case is validity, $\models F$
- If X is finite, we can check logical consequence using a truth table
- What if X is infinite?

Axiomatizations and proofs

- Set up a formal system to derive judgements about logical consequences
- $X \vdash F$ will dnote that *"F can be derived from X"*
- Inference rules reflect the semantics

  If $X \vdash F$ and $X \vdash G$ then $X \vdash F \wedge G$ ($\wedge$ introduction)

- Rules are uniquely identified by a label, here "$\wedge$ introduction"
- Typically written "vertically" as

  $$\frac{X \vdash F, X \vdash G}{X \vdash F \wedge G} \ (\wedge \text{ introduction})$$

- Above the line is the premise, below is the conclusion

Rules

$$\frac{G \in X}{X \vdash G} \text{ (Axiom)} \qquad \frac{X \vdash G, X \subseteq X'}{X' \vdash G} \text{ (Monotonicity)}$$

$$\frac{X \vdash G}{X \vdash \neg\neg G} \text{ (Double negation)} \qquad \frac{X \vdash F, X \vdash G}{X \vdash F \wedge G} \text{ ($\wedge$ introduction)}$$

$$\frac{X \vdash F \wedge G}{X \vdash F} \text{ ($\wedge$ elimination)} \qquad \frac{X \vdash F \wedge G}{X \vdash G \wedge F} \text{ ($\wedge$ symmetry)}$$

$$\frac{X \vdash F \vee G}{X \vdash G \vee F} \text{ ($\vee$ symmetry)} \qquad \frac{X \vdash F}{X \vdash F \vee G} \text{ ($\vee$ introduction)}$$

$$\frac{X \vdash F \vee G, X \cup \{F\} \vdash H, X \cup \{G\} \vdash H}{X \vdash H} \text{ ($\vee$ elimination)}$$

$$\frac{X \cup \{F\} \vdash G}{X \vdash F \rightarrow G} \text{ ($\rightarrow$ introduction)} \qquad \frac{X \vdash F \rightarrow G, \; X \vdash F}{X \vdash G} \text{ ($\rightarrow$ elimination)}$$

Formal proofs

- Some more rules to rewrite $\rightarrow$, $\leftrightarrow$ in terms of $\neg, \vee$ ...
- A proof is a sequence of statements $X \vdash F$ where each line follows from a previous one by one of the rules

Example

- Anything can be derived from a contradiction
- Assume $F \wedge \neg F \in X$

1.	$X \vdash F \wedge \neg F$	(Axiom)
2.	$X \vdash \neg F \wedge F$	($\wedge$ symmetry, 1)
3.	$X \vdash \neg F$	($\wedge$ elimination, 2)
4.	$X \vdash \neg F \vee G$	($\vee$ introduction, 3)
5.	$X \vdash F \rightarrow G$	($\rightarrow$ rewrite, 5)
6.	$X \vdash F$	($\wedge$ elimination, 1)
7.	$X \vdash G$	($\rightarrow$ elimination, 5 and 6)

Soundness

- If $X \vdash F$ then $X \models F$
- Derivations only reveal "true" logical consequences
- By induction on the length of the proof
- The Axiom is sound
- Every rule preserves soundness

Completeness

- If $\boldsymbol{X} \models \boldsymbol{F}$ then $\boldsymbol{X} \vdash \boldsymbol{F}$
- Every logical consequence can be derived in the system
- This is more difficult to prove
- Introduce a new rule, resolution

Resolution

- Assume F is in CNF
- Recall that a clause can be seen as a set of literals
- Let C_1, C_2 be clauses and $A \in \mathcal{P}$ such that $A \in C_1$, $\neg A \in C_2$
- We can resolve C_1 and C_2 to get $R = (C_1 \setminus \{A\}) \cup (C_2 \setminus \{\neg A\})$

Example

- $C_1 = \{A_1, \neg A_2, A_3\}$, $C_2 = \{A_2, \neg A_3, A_4\}$
- Resolve (on A_3) to get $\{A_1, A_2, \neg A_2, A_4\}$
- Resolvent is not unique—resolve on A_2 to get $\{A_1, \neg A_3, A_3, A_4\}$

Soundness of Resolution

Soundness

Let R be a resolvent of C_1 and C_2. Then $\{C_1, C_2\} \vdash R$

Let $X = \{C_1, C_2\}$, $C_1 = A \vee F$, $C_2 = \neg A \vee G$

1.	$X \vdash A \vee F$	(Axiom)
2.	$X \cup \{\neg A\} \vdash A \vee F$	(Monotonicity, 1)
3.	$X \cup \{\neg A\} \vdash \neg A$	(Axiom)
4.	$X \cup \{\neg A\} \vdash F$	($\rightarrow$ elimination, 2 and 3)
5.	$X \cup \{\neg A\} \vdash F \vee G$	($\vee$ introduction, 4)
6.	$X \vdash \neg A \vee G$	(Axiom)
7.	$X \cup \{\neg\neg A\} \vdash \neg A \vee g$	(Monotonicity, 6)
8.	$X \cup \{\neg\neg A\} \vdash \neg\neg A$	(Axiom)
9.	$X \cup \{\neg\neg A\} \vdash G$	($\rightarrow$ elimination, 7 and 8)
10.	$X \cup \{\neg\neg A\} \vdash G \vee F$	($\vee$ introduction, 9)
11.	$X \cup \{\neg\neg A\} \vdash F \vee G$	($\vee$ symmetry, 10)
12.	$X \vdash F \vee G$	(Proof by cases, 5 and 11)

Soundness of Resolution

- Hence we can add Resolution as a rule to our formal proof system
- In fact, we need only Resolution to prove completness!
- Resolution preserve satisfiability
  - If C_1, C_2 are satisfiable, their resolvent R is satisfiable
  - If R is not satisfiable, C_1, C_2 are not satisfiable
  - Empty clause (empty disjunction) is not satisfiable
  - If resolution produces an empty clause, we have derived a contradiction

Completeness

- Let $\text{Res}^0(F) = \{C \mid C \text{ is a clause in } F\}$
- For $n > 0$, $\text{Res}^n(F) = \text{Res}^{n-1}(F) \cup$
  $\{R \mid R \text{ is a resolvent of two clauses in } \text{Res}^{n-1}(F)\}$
- Since F is finite, we can only apply resolution a finite number of times
- For some m, $\text{Res}^m(F) = \text{Res}^{m+1}(F) = \text{Res}^*(F)$

If $\emptyset \in \text{Res}^*(F)$, then F is unsatisfiable

- $\emptyset$ can only arise as resolvent of $\{A\}, \{\neg A\}$

Completeness ...

If F is unsatisfiable, then $\emptyset \in \mathrm{Res}^*(F)$

- Assume F is in CNF
- Discard all tautological clauses
- Proof is by induction on number of atomic propositions in F
- Base case, one atomic proposition
  - Possible clauses are $\{A\}$, $\{\neg A\}$, $\{A, \neg A\}$
  - Last is a tautology, discard
  - $F = \{\{A\}\}$ or $F = \{\{\neg A\}\}$, F is satisfiable
  - $F = \{\{A\}, \{\neg A\}\}$, F is unsatisfiable, $\emptyset \in \mathrm{Res}^*(F)$
- Induction step ...

Completeness ...

Let $F, G \in \mathcal{F}$. Let H be CNF form of $F \wedge \neg G$.
The following are equivalent.

1. $F \models G$
2. $\{F\} \vdash G$
3. $\emptyset \in \mathrm{Res}^*(H)$

Compactness

Consider the following infinite set of sentences

- *The universe has finitely many objects*
- *The universe has at least one object*
- *The universe has at least two objects*

  . . .
- *The universe has at least* n *objects*

  . . .

- The entire set of sentences is contradictory
- However, every finite subset of sentences is satisfiable
- Compactness says that such a situation is impossible

Compactness . . .

Compactness

A set of formulas X is unsatisfiable iff some finite subset of X is unsatisfiable

To prove this, we need König's Lemma.

König's Lemma

Let T be a finitely branching tree with infinitely many nodes. Then T has an infinite path

- Call a node in T good if the subtree below the node is infinite
- Clearly the root of T is good
- Every good node has at least one good child (finite branching!)
- Build an infinite path starting from the root, extending it to include one good child at each step

Compactness . . .

- Enumerate $\mathcal{P} = \{A_0, A_1, A_2, \ldots\}$
- A k-assignment is a function $f : \{A_0, A_1, \ldots, A_k\} \to \{0, 1\}$
- Build a tree $T_{\mathcal{A}}$ of k-assignments where
    - Root is empty assignment
    - Nodes at level j are j-assignments
    - Children of a node at level j correspond to extensions setting $A_{j+1} \mapsto 0$ and $A_{j+1} \mapsto 1$
    - Infinite binary tree, each infinite path is an assignment $\mathcal{A}$

Compactness . . .

- Suppose every finite set of X is satisfiable, but X is not satisfiable overall
- Call a k-valuation in $T_{\mathcal{A}}$ bad if it does not satisfy some formula in X
- Prune each path below the first bad node on that path
- If the resulting tree is infinite, it has an infinite path π in which no nodes are bad
- This path π defines a valuation that satisfies X
  - Pick any $F \in X$.
  - Let A_j be the largest proposition in F
  - The j-valuation at depth j is not bad, so it satisfies F

Compactness . . .

- Suppose every finite set of X is satisfiable, but X is not satisfiable overall
- Call a k-valuation in $T_{\mathcal{A}}$ bad if it makes some formula in X false
- Prune each path below the first bad node on that path
- The resulting tree must be finite, otherwise we have a valuation that satisfies X
- This finite tree has a finite frontier $\{v_1, v_2, \ldots, v_m\}$
- Each frontier node v_i is bad, so it fails to satisfy some formula $F_i \in X$
- $\{F_1, F_2, \ldots, F_k\} \subseteq X$ is not satisfiable
- Contradiction! Every finite subset of X is satisfiable

Compactness . . .

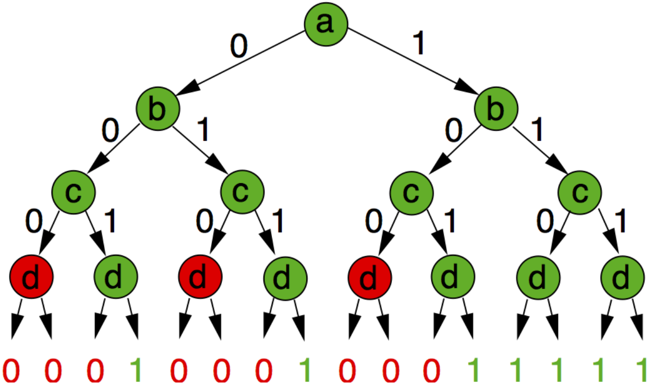
Compactness

If $X \models F$, then there is finite subset Y of X such that $Y \models F$

- IF $X \models F$ the $X \cup \{\neg F\}$ is unsatisfiable
- By previous argument, some finite subset Y' of $X \cup \{\neg F\}$ is unsatisfiable
- Choose $Y = Y' \setminus \{\neg F\}$
- Clearly $Y \cup \{\neg F\}$ is not satisfiable
- Hence $Y \models F$

Boolean functions

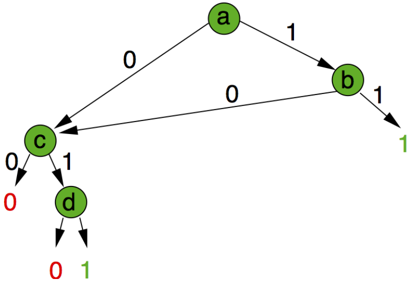
Ordered decision tree for $f(a,b,c,d) = (a \land b) \lor (c \land d)$



Binary decision diagram (BDD)

Compact representation of boolean functions ([Bryant 1986])

- Reduced ordered binary decision diagram for $f(a, b, c, d) = (a \wedge b) \vee (c \wedge d)$
- Key idea
  Combine equivalent subcases



BDDs ...

- BDD for f is canonical (for a fixed variable order)
  - Check if $f = g$ by comparing their BDDs
- Efficient algorithms for combining BDDs
  - Build BDD for f **op** g for boolean operator **op** from BDDs for f, g
  - e.g., given BDD for f and g, can build BDD for $f \wedge g$
- Use BDDs to represent and manipulate state spaces
  - Symbolic model checking ([Clarke, McMillan et al])
  - More useful for hardware model checking than software model checking
  - Still at the heart of many tools