# CS 477: Background and Propositional Logic <br> Sasa Misailovic 

Based on previous slides by Elsa Gunter, which were based on earlier slides by Gul Agha, and Mahesh Viswanathan

University of Illinois at Urbana-Champaign

## From Real World to Formal Methods

```
Source Program:
```

```
int binsearch(int x, int v[], int n)
    int low, high, mid;
    1 low = 0;
    high = n - 1;
        high = n - l; 
        {hil
            3 mid = (low + high)/2;
            3 if (x < v[mid])
            high = mid - 1; 4
            5 else if (x > v[mid])
            7 else return mid; 1; |
            7 else return mid;
        }}\mathrm{ return -1; |
} |
```


$\forall \mathrm{n}>0$. low $\leq$ high $\wedge$ high $\leq n$

## Set

A collection of elements

- Reminder: empty set, singleton, subset, powerset, cardinality
- Natural numbers N, Integers Z, Reals R, Machine numbers
- Operations: Union, Intersection, Complement, Cartesian Product


## Relation

- Relation over $\mathrm{X}_{1}, \mathrm{X}_{2}, \ldots \mathrm{X}_{\mathrm{n}}$ : A subset of $\mathrm{X}_{1} \times \mathrm{X}_{2} \times \ldots \times \mathrm{X}_{\mathrm{n}}$
- Binary: $(\mathrm{x}, \mathrm{y}) \in R, R \subseteq X \times Y$
- Operators: relations are sets => set operators
- Properties: Transitive, Reflexive, Symmetric, Asymmetric,...
- Equivalence relation: reflexive+symmetric+transitive
- Partial Order relation: reflexive+antisymmetric+transitive
- Preorder: reflexive+transitive


## Function

- Special relation: a particular input has a single output - $y=f(x)$, also $(x, y) \in f, f \subseteq X \times Y$ where $x \in X, y \in Y$
- $\forall x \in X, y \in Y, z \in Z . \quad(x, y) \in f \wedge(x, z) \in f \Rightarrow y=z$
- Domain X and codomain Y ; type: $X \rightarrow Y$; total and partial functions
- Input/results of a function can be another function


## Language

- Alphabet (set $\Sigma$ ), language ( $\subseteq \Sigma^{*}$ ) and words ( $w \in \Sigma \times \cdots \times \Sigma$ )
- Empty word ( $\varepsilon$ ), length, prefix/suffix, concatenation( $w$ w), alternation ( $w \mid w$ )
- Regular expression
- $\mathrm{E}::=\varepsilon|\mathrm{A} \in \mathrm{\Sigma}| \mathrm{EE} \mid \mathrm{E}$ "|"E|E*
- Equivalent to finite state machine (automaton)

Syntax (words in language) + Semantics (meaning of those words: relating to other math objects)

Simple Imperative Programming Language (more complex)

- I $\in$ Identifiers
- $\mathrm{N} \in$ Numerals
- $B$ ::= true | false
$|\mathrm{B} \& \mathrm{~B}| \mathrm{B}$ or $\mathrm{B}|\operatorname{not} \mathrm{B}| \mathrm{E}<\mathrm{E} \mid \mathrm{E}=\mathrm{E}$
- $\mathrm{E}::=\mathrm{N}|\mathrm{I}| \mathrm{E}+\mathrm{E}|\mathrm{E} * \mathrm{E}| \mathrm{E}-\mathrm{E} \mid-\mathrm{E}$
- C::= skip | C; C | $1::=\mathrm{E}$
| if B then C else C fi \| while B do C od


## Program Representations:

- Graphs: (V,E) - set of vertices V and set of edges E

Some models of computation:

- Automaton: represents a computation
- Pushdown automaton: automaton with a stack (for CFGrammars)
- Turning machine: an automaton with a memory tape

Some representations of execution (static/dynamic):

- Parse tree and Abstract syntax tree: represent syntax
- Traces/Paths: a sequence of executed instructions or states
- Transition system: represents possible executions
- Control-flow graph: succinctly represents paths in a program


## Propositional Logic

- Syntax
- Semantics (truth tables)
- Proofs


## Propositional Logic

The Language of Propositional Logic

- Constants $\{\mathrm{T}, \mathrm{F}\}$
- Countable set AP of propositional variables ( $x, y, z$ ), a.k.a. propositional atoms, a.k.a. atomic propositions
- logical connectives: $\wedge$ (and); $\vee$ (or); $\sim(n o t) ; ~ \Rightarrow$ (implies); $\Leftrightarrow$ (if and only if)


## Propositional Logic (cont.)

The set of propositional formulae PROP is the inductive closure of the previous elements as follows:

- $\{T, F\} \subseteq P R O P$
- AP $\subseteq$ PROP
- if $A \in P R O P$ then ( $A$ ) $\in P R O P$ and $\neg A \in P R O P$
- if $A \in P R O P$ and $B \in P R O P$ then ( $A \wedge B$ ) $\in P R O P$, ( $\mathrm{A} \vee \mathrm{B}$ ) $\in P R O P$, $(A \Rightarrow B) \in P R O P,(A \Leftrightarrow B) \in P R O P$.
- Nothing else is in PROP
- Informal definition; formal definition requires math foundations, set theory, fixed point theorem ...


## Propositional Logic

We can write it as a grammar too:

- C : : = T | F
- AP ::=x|y|z|...
- PROP ::= C | AP | (PROP) | -PROP

PROP $\wedge$ PROP | PROP V PROP
PROP $\Rightarrow$ PROP | PROP $\Leftrightarrow$ PROP

We can get various "sentences" in this language.

$$
\text { E.g. } x \wedge y,(x \wedge y) \Rightarrow(x \vee y), x \vee \neg x \Leftrightarrow T
$$

But what is their meaning?

## Toward Propositional Logic Semantics

Model for Propositional Logic has three parts

- Mathematical set of values used as meaning of propositions
- Interpretation function giving meaning to props built from logical connectives, via structural recursion
- Standard Model of Propositional Logic
- Boolean values $B=\{$ true,false $\}$
- a valuation $v: A P \rightarrow B$

| $\mathbf{A P}$ | $\mathbf{B}$ |
| :--- | :--- |
| $x$ | true |
| $y$ | false |
| $z$ | true |

## Semantics of Propositional Logic

Standard interpretation $I_{v}$ defined by structural induction on formulae:

- $\mathrm{I}_{\mathrm{v}}(\mathrm{T})=$ true and $\mathrm{I}_{\mathrm{v}}(\mathrm{F})=$ false
- If $a \in A P$ then $I_{v}(a)=v(a)$
- For $p \in \operatorname{PROP}$, if $I_{v}(p)=$ true then $I_{v}(\neg p)=$ false, and if $I_{v}(p)=$ false then $I_{v}(\neg p)=$ true
- For $p, q \in \operatorname{PROP}$ :
-If $\mathrm{I}_{\mathrm{v}}(\mathrm{p})=$ true and $\mathrm{I}_{\mathrm{v}}(\mathrm{q})=$ true, then $\mathrm{I}_{\mathrm{v}}(\mathrm{p} \wedge \mathrm{q})=$ true, else $\mathrm{I}_{\mathrm{v}}(\mathrm{p} \wedge \mathrm{q})=$ false
-If $I_{v}(p)=$ true or $I_{v}(q)=$ true, then $I_{v}(p \vee q)=$ true, else $I_{v}(p \vee q)=$ false
-If $I_{v}(q)=$ true or $I_{v}(p)=$ false, then $I_{v}(p \Rightarrow q)=$ true, else $I_{v}(p \Rightarrow q)=$ false
-If $I_{v}(p)=I_{v}(q)$ then $I_{v}(p \Leftrightarrow q)=$ true, else $I_{v}(p \Leftrightarrow q)=$ false


## Example

- $\mathrm{I}_{\mathrm{v}}(\mathrm{T})=$ true and $\mathrm{I}_{\mathrm{v}}(\mathrm{F})=$ false
- If $a \in A P$ then $I_{v}(a)=v(a)$
- For $p \in P R O P$, if $I_{v}(p)=$ true then $I_{v}(\neg p)=$ false, and if $I_{v}(p)=$ false then $I_{v}(\neg p)=$ true
- For $p, q \in \operatorname{PROP}$ :
- If $I_{v}(p)=$ true and $I_{v}(q)=$ true, then $I_{v}(p \wedge q)=$ true, else $I_{v}(p \wedge q)=$ false
- If $I_{v}(p)=$ true or $I_{v}(q)=$ true, then $I_{\gamma}(p \vee q)=$ true, else $I_{v}(p \vee q)=$ false
- If $I_{v}(q)=$ true or $I_{v}(p)=$ false, then $I_{v}(p \Rightarrow q)=$ true, else $I_{v}(p \Rightarrow q)=$ false
- If $I_{v}(p)=I_{v}(q)$ then $I_{v}(p \Leftrightarrow q)=$ true, else $I_{v}(p \Leftrightarrow q)=$ false

| $p$ | $q$ | $\neg p$ | $p \wedge q$ | $p \vee q$ | $p \Rightarrow q$ | $p \Leftrightarrow q$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| true | true |  |  |  |  |  |
| true | false |  |  |  |  |  |
| false | true |  |  |  |  |  |
| false | false |  |  |  |  |  |
|  |  |  |  |  |  |  |

## Example

- $\mathrm{I}_{\mathrm{v}}(\mathrm{T})=$ true and $\mathrm{I}_{\mathrm{v}}(\mathrm{F})=$ false
- If $a \in A P$ then $I_{v}(a)=v(a)$
- For $p \in \operatorname{PROP}$, if $I_{v}(p)=$ true then $I_{v}(\neg p)=$ false, and if $\mathrm{I}_{\mathrm{v}}(\mathrm{p})=$ false then $\mathrm{I}_{\mathrm{v}}(\neg \mathrm{p})=$ true
- For $p, q \in \operatorname{PROP}$ :
- If $I_{v}(p)=$ true and $I_{v}(q)=$ true, then $I_{v}(p \wedge q)=$ true, else $I_{v}(p \wedge q)=$ false
- If $I_{v}(p)=$ true or $I_{v}(q)=$ true, then $I_{v}(p \vee q)=$ true, else $I_{v}(p \vee q)=$ false
- If $I_{v}(q)=$ true or $I_{v}(p)=$ false, then $I_{v}(p \Rightarrow q)=$ true, else $I_{v}(p \Rightarrow q)=$ false
- If $I_{v}(p)=I_{v}(q)$ then $I_{v}(p \Leftrightarrow q)=$ true, else $I_{v}(p \Leftrightarrow q)=$ false

| $p$ | $q$ | $\neg p$ | $p \wedge q$ | $p \vee q$ | $p \Rightarrow q$ | $p \Leftrightarrow q$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| true | true | false | true | true | true | true |
| true | false | false | false | true | false | false |
| false | true | true | false | true | true | false |
| false | false | true | false | false | true | true |

## Semantics of Propositional Logic

$(\mathrm{B}, \mathrm{I})$ is the standard model of proposition logic

- Satisfaction relation $\vDash$ : Given valuation $v$ and proposition $p \in P R O P$, we write $\mathrm{v} \vDash \mathrm{p}$ iff $\mathrm{I}_{\mathrm{v}}(\mathrm{p})=$ true (the $\vDash$ symbol name is called "double turnstile")
- More fully written as $\mathrm{B}, \mathrm{I}, \mathrm{v} \vDash \mathrm{p}$.
- Can also write (B,I,v,p) $\in \models$
- Say valuation v satisfies $p$, or $v$ models $p$
- Write v $\neq p$ if $I_{v}(p)=$ false
- $p$ is satisfiable if there exists valuation $v$ such that $v \vDash p$
- $p$ is valid, a.k.a. a tautology if for every valuation $v$ we have $v \vDash p$
- p is logically equivalent to $\mathrm{q}, \mathrm{p} \equiv \mathrm{q}$ if for every valuation, v , we have $\mathrm{v} \vDash \mathrm{p}$ iff $\mathrm{v} \vDash \mathrm{q}$. Claim: Logical equivalence is an equivalence relation

We can have other models of this logic, e.g. defined via sets

## Example Tautology

$$
A \Rightarrow((A \Rightarrow B) \Rightarrow B)
$$

| $A$ | $B$ | $A \Rightarrow B$ | $(A \Rightarrow B) \Rightarrow B$ | $A \Rightarrow((A \Rightarrow B) \Rightarrow B)$ |
| :---: | :---: | :---: | :---: | :---: |
| true | true | true | true | true |
| true | false | false | true | true |
| false | true | true | true | true |
| false | false | true | false | true |

## Some Useful Logical Equivalences

$$
\begin{array}{rlrl}
\neg \neg A & \equiv A & \neg \mathbf{T} \equiv \mathbf{F} & \neg \mathbf{F} \equiv \mathbf{T} \\
(A \vee A) & \equiv A & & (A \vee B) \vee C \equiv A \vee(B \vee C) \\
(A \wedge A) & \equiv A & & (A \wedge B) \wedge C \equiv A \wedge(B \wedge C) \\
A \vee B & \equiv B \vee A & \neg(A \vee B) \equiv(\neg A) \wedge(\neg B) \\
A \wedge B & \equiv B \wedge A & \neg(A \wedge B) \equiv(\neg A) \vee(\neg B) \\
(A \wedge \neg A) & \equiv \mathbf{F} & A \vee(B \wedge C) \equiv(A \vee B) \wedge(A \vee C) \\
(A \vee \neg A) & \equiv \mathbf{T} & & (A \wedge B) \vee C \equiv(A \vee C) \wedge(B \vee C) \\
(\mathbf{T} \wedge A) & \equiv A & & A \wedge(B \vee C) \equiv(A \wedge B) \vee(A \wedge C) \\
(\mathbf{T} \vee A) & \equiv \mathbf{T} & & (A \wedge B) \vee C \equiv(A \wedge C) \vee(B \wedge C) \\
(\mathbf{F} \wedge A) & \equiv \mathbf{F} & & (\mathbf{F} \vee A) \equiv A
\end{array}
$$

## Normal Forms

Conjunctive normal form (CNF):

$$
\bigwedge_{i \in I} \bigvee_{j \in J_{i}} A P_{i, j}
$$

Disjunctive normal form (DNF):

Question:<br>What is the computational complexity of finding a satisfying assignment of variables?

where $I$ and $J$ are index sets

## Proofs in Propositional Logic

Constructing the full truth table is expensive!

Natural Deduction proof is tree and a discharge function

- Nodes are instances of inference rules
- Leaves are assumptions of subproofs
- Discharge function maps each leaf of the tree to an ancestor as prescribed by the inference rules


## Proofs in Propositional Logic

- Inference rule has hypotheses and conclusion
- Conclusion (C) is a single proposition
- Hypotheses (H) are zero or more propositions, possibly with (discharged) hypotheses
- Rule with no hypotheses is called an axiom (A)
- Inference rule graphically presented as



## Natural Deduction Inference Rules

Inference rules associated with connectives
Two main kinds of inference rules:

- Introduction: says how to conclude proposition made from connective is true

- Eliminations: says how to use a proposition made from connective to prove result



## Why: Conjunction?

Introduction:
$\frac{A \wedge}{A \wedge B}$ And I

Elimination: *simplified

- Local soundness: if we introduce a conjunction and then eliminate it, we should be able to remove the whole subtree [local reduction $\Rightarrow_{R}$ ]
- Local completeness: we can eliminate the conjunction such that we can still reconstruct it by applying introduction [local expansion $\Rightarrow_{E}$ ]
$\frac{A \wedge B}{A} \quad \operatorname{And}_{L} \mathrm{E}$
$\frac{A \wedge B}{B} \quad \operatorname{And}_{R} \mathrm{E}$
*From https://www.cs.cmu.edu/~fp/courses/atp/handouts/ch2-natded.pdf for full description


## Why: Conjunction?

Introduction:


Elimination: *simplified
*soundness:
$\frac{A \wedge B}{B} \quad \operatorname{And}_{R} \mathrm{E}$

*completeness:

- Local soundness: if we introduce a conjunction and then eliminate it, we should be able to remove the whole subtree [local reduction $\Rightarrow_{R}$ ]
- Local completeness: we can eliminate the conjunction such that we can still reconstruct it by applying introduction [local expansion $\Rightarrow_{E}$ ]


[^0]
## Why: Conjunction?

Introduction:
$\frac{A \wedge}{A \wedge B}$ And I

Elimination:


$$
B
$$



- Local soundness: if we introduce a conjunction and then eliminate it, we should be able to remove the whole subtree [local reduction $\Rightarrow_{R}$ ]
- Local completeness: we can eliminate the conjunction such that we can still reconstruct it by applying introduction [local expansion $\Rightarrow_{E}$ ]


## Introduction Rules

Truth Introduction
And Introduction:

$$
\bar{T}^{T}
$$

$$
\frac{A \wedge B}{A \wedge B} \text { And } I
$$

Or Introduction:

$$
\frac{A}{A \vee B} \operatorname{Or}_{L} \mathrm{I} \quad \frac{B}{A \vee B} \operatorname{Or}_{R} 1
$$

Not Introduction:
Implication Introduction:


No False Introduction

$$
\overline{\mathbf{T}} \mathbf{T} \mathbf{I} \quad \frac{A \wedge}{A \wedge B} \text { And I }
$$

## Example 1

## Or Introduction:

$$
\begin{gathered}
\frac{A}{A \vee B} \mathrm{Or}_{L} \mathrm{I} \\
\text { Implication Introduction: }
\end{gathered} \frac{B}{A \vee B} \mathrm{Or}_{R} \mathrm{I}
$$

Not Introduction:
$A$
$\vdots$
$\frac{\mathbf{F}}{\neg A}$ Not I

$$
\begin{gathered}
A \\
\vdots \\
\frac{B}{A \Rightarrow B} \operatorname{Imp~I}
\end{gathered}
$$

No False Introduction

## $A \Rightarrow(B \Rightarrow(A \wedge B))$

## Example 1

$A \quad B$

- And I
$A \wedge B$
$\longrightarrow$ Imp |
$B \Rightarrow(A \wedge B)$
Imp I
$A \Rightarrow(B \Rightarrow(A \wedge B))$


## Example 1

## A B

$-B$ And I
$A \wedge B$
$\bar{B}(A \wedge B)$ Imp I
$B \Rightarrow(A \wedge B)$
Imp I
$A \Rightarrow(B \Rightarrow(A \wedge B))$
All assumptions discharged; proof complete


[^0]:    *From https://www.cs.cmu.edu/~fp/courses/atp/handouts/ch2-natded.pdf for full description

