

# G22.3033.11 — Logic and Verification

## Instructor:

Clark W. Barrett

[barrett@cs.nyu.edu](mailto:barrett@cs.nyu.edu)

# Outline

---

- Goals and Organization
- A Motivating Example
- Propositional Logic: Syntax
- Induction
- Propositional Logic: Well-Formed Formulas
- Recursion
- Propositional Logic: Semantics

Material is drawn from Sections 1.1–1.4 of Enderton.

# Course Goals

- Prerequisites:
  - Familiarity with discrete mathematics (sets, functions, induction, graphs)
  - Working acquaintance with the language of logic
  - Ability to read and write sophisticated programs
- Contents of the Course
  - A precise and formal treatment of two logics: propositional and first-order
  - An introduction to the mathematical tools required to reason within these formal frameworks
  - An emphasis on *Interesting Applications* of formal logic, including the *real-world* example of *Formal Verification* of systems.

# Course Information

## Webpage:

<http://home.nyu.edu>: click on the **Academics** tab. You should see a link to the web page under **Classes**. If not, send me email with your NYU ID.

## Book:

Enderton, Herbert B. **A Mathematical Introduction to Logic**

## Assignments:

There will be a weekly assignment which will generally be due the following week (unless otherwise indicated).

## Project:

There will be a final project which will require using one of the tools we be discuss to solve a verification problem.

## A Motivating Example

Recall that a *graph* consists of a set  $V$  of vertices and a set  $E$  of edges, where each edge is an unordered pair of vertices.

A *complete graph* on  $n$  vertices is a graph with  $|V| = n$  such that  $E$  contains all possible pairs of vertices.

## A Motivating Example

Recall that a *graph* consists of a set  $V$  of vertices and a set  $E$  of edges, where each edge is an unordered pair of vertices.

A *complete graph* on  $n$  vertices is a graph with  $|V| = n$  such that  $E$  contains all possible pairs of vertices.

How many edges?

## A Motivating Example

Recall that a *graph* consists of a set  $V$  of vertices and a set  $E$  of edges, where each edge is an unordered pair of vertices.

A *complete graph* on  $n$  vertices is a graph with  $|V| = n$  such that  $E$  contains all possible pairs of vertices.

How many edges?

The *Ramsey number*  $r(x_1, \dots, x_n)$  is the smallest integer  $p$  such that if a complete graph  $G$  on  $p$  vertices is colored with  $n$  colors, then for some  $i$ ,  $1 \leq i \leq n$ , there must exist a complete subgraph of  $G$  with  $x_i$  vertices, all of whose edges have the same color.

## A Motivating Example

Recall that a *graph* consists of a set  $V$  of vertices and a set  $E$  of edges, where each edge is an unordered pair of vertices.

A *complete graph* on  $n$  vertices is a graph with  $|V| = n$  such that  $E$  contains all possible pairs of vertices.

How many edges?

The *Ramsey number*  $r(x_1, \dots, x_n)$  is the smallest integer  $p$  such that if a complete graph  $G$  on  $p$  vertices is colored with  $n$  colors, then for some  $i$ ,  $1 \leq i \leq n$ , there must exist a complete subgraph of  $G$  with  $x_i$  vertices, all of whose edges have the same color.

What is  $r(3, 3)$ ?

## A Motivating Example

Recall that a *graph* consists of a set  $V$  of vertices and a set  $E$  of edges, where each edge is an unordered pair of vertices.

A *complete graph* on  $n$  vertices is a graph with  $|V| = n$  such that  $E$  contains all possible pairs of vertices.

How many edges?

The *Ramsey number*  $r(x_1, \dots, x_n)$  is the smallest integer  $p$  such that if a complete graph  $G$  on  $p$  vertices is colored with  $n$  colors, then for some  $i$ ,  $1 \leq i \leq n$ , there must exist a complete subgraph of  $G$  with  $x_i$  vertices, all of whose edges have the same color.

What is  $r(3, 3)$ ?

What is  $r(3, 3, 3)$ ?

# Logic

---

A formal logic is defined by its *syntax* and *semantics*.

## Syntax

- An *alphabet* is a set of symbols.
- A finite sequence of these symbols is called an *expression*.
- A set of rules for forming *well-formed* expressions.

## Semantics

- Semantics gives meaning to well-formed expressions, suggesting ways of translating mathematical English into formal logic and vice versa.
- There are different ways to define the semantics for a logic.
- Typically depend on formal notions of induction and recursion.

# Propositional (Sentential) Logic

Propositional logic is simple but extremely important in Computer Science

1. It is the basis for day-to-day reasoning (in programming, LSATs, etc.)
2. It is the theory behind digital circuits.
3. Many problems can be translated into propositional logic.
4. It is an important part of more complex logic (such as *first-order logic*, also called *predicate logic*, which we'll discuss later.)

# Propositional Logic: Syntax

## Alphabet

(	Left parenthesis	Begin group
)	Right parenthesis	End group
$\neg$	Negation symbol	English: not
$\wedge$	Conjunction symbol	English: and
$\vee$	Disjunction symbol	English: or (inclusive)
$\rightarrow$	Conditional symbol	English: if, then
$\leftrightarrow$	Bi-conditional symbol	English: if and only if
$A_1$	First propositional symbol	
$A_2$	Second propositional symbol	
...		
$A_n$	$n$ th propositional symbol	
...		

We are assuming a *countable* alphabet, but most of our conclusions hold equally well for an *uncountable* alphabet.

# Propositional Logic: Syntax

## Alphabet

- *Propositional connective* symbols:  $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ .
- *Logical* symbols:  $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, (, )$ .
- *Parameters* or *nonlogical symbols*:  $A_1, A_2, A_3, \dots$

The meaning of logical symbols is always the same. The meaning of nonlogical symbols depends on the context.

## Propositional Logic: Syntax

An *expression* is a sequence of symbols. A sequence is denoted explicitly by a comma separated list enclosed in angle brackets:  $\langle a_1, \dots, a_m \rangle$ .

Examples

$\langle (, A_1, \wedge, A_3, ) \rangle$

$\langle (, (, \neg, A_1, ), \rightarrow, A_2, ) \rangle$

$\langle ), ), \leftrightarrow, ), A_5 \rangle$

## Propositional Logic: Syntax

An *expression* is a sequence of symbols. A sequence is denoted explicitly by a comma separated list enclosed in angle brackets:  $\langle a_1, \dots, a_m \rangle$ .

Examples

$$\begin{aligned} \langle (, A_1, \wedge, A_3, ) \rangle & \quad (A_1 \wedge A_3) \\ \langle (, (, \neg, A_1, ), \rightarrow, A_2, ) \rangle & \quad ((\neg A_1) \rightarrow A_2) \\ \langle ), ), \leftrightarrow, ), A_5 \rangle & \quad )) \leftrightarrow) A_5 \end{aligned}$$

For convenience, we will write these sequences as a simple string of symbols, with the understanding that the *formal* structure represented is a sequence containing exactly the symbols in the string.

The formal meaning becomes important when trying to prove things about expressions.

## Propositional Logic: Syntax

An *expression* is a sequence of symbols. A sequence is denoted explicitly by a comma separated list enclosed in angle brackets:  $\langle a_1, \dots, a_m \rangle$ .

Examples

$$\begin{aligned} \langle (, A_1, \wedge, A_3, ) \rangle & \quad (A_1 \wedge A_3) \\ \langle (, (, \neg, A_1, ), \rightarrow, A_2, ) \rangle & \quad ((\neg A_1) \rightarrow A_2) \\ \langle ), ), \leftrightarrow, ), A_5 \rangle & \quad )) \leftrightarrow) A_5 \end{aligned}$$

For convenience, we will write these sequences as a simple string of symbols, with the understanding that the *formal* structure represented is a sequence containing exactly the symbols in the string.

The formal meaning becomes important when trying to prove things about expressions.

We want to restrict the kinds of expressions that will be allowed.

## Propositional Logic: Syntax

We define the set  $W$  of *well-formed formulas* (*wff*'s) as follows.

- (a) Every expression consisting of a single propositional symbol is in  $W$ .
- (b) If  $\alpha$  and  $\beta$  are in  $W$ , so are  $(\neg\alpha)$ ,  $(\alpha \wedge \beta)$ ,  $(\alpha \vee \beta)$ ,  $(\alpha \rightarrow \beta)$ , and  $(\alpha \leftrightarrow \beta)$ .
- (c) No expression is in  $W$  unless forced by (a) or (b)

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

## Propositional Logic: Syntax

We define the set  $W$  of *well-formed formulas* (*wff*'s) as follows.

- (a) Every expression consisting of a single propositional symbol is in  $W$ .
- (b) If  $\alpha$  and  $\beta$  are in  $W$ , so are  $(\neg\alpha)$ ,  $(\alpha \wedge \beta)$ ,  $(\alpha \vee \beta)$ ,  $(\alpha \rightarrow \beta)$ , and  $(\alpha \leftrightarrow \beta)$ .
- (c) No expression is in  $W$  unless forced by (a) or (b)

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

How would you use this definition to prove that  $((\leftrightarrow)A_5)$  is not a *wff*?

## Propositional Logic: Syntax

We define the set  $W$  of *well-formed formulas* (*wff*'s) as follows.

- (a) Every expression consisting of a single propositional symbol is in  $W$ .
- (b) If  $\alpha$  and  $\beta$  are in  $W$ , so are  $(\neg\alpha)$ ,  $(\alpha \wedge \beta)$ ,  $(\alpha \vee \beta)$ ,  $(\alpha \rightarrow \beta)$ , and  $(\alpha \leftrightarrow \beta)$ .
- (c) No expression is in  $W$  unless forced by (a) or (b)

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

How would you use this definition to prove that  $((\leftrightarrow)A_5)$  is not a *wff*?

Item (c) is too vague for our purposes. There are two ways to make it more precise: *top-down* and *bottom-up*. Both require a formal notion of *induction*.

# Induction

---

Suppose we have a property  $P$  which is defined in terms of a natural number  $n$ . We wish to show that  $P$  holds for all natural numbers.

## Base case

Show that  $P$  holds for 0.

## Inductive case

Show that if  $P$  holds for  $n$ , then  $P$  holds for  $n + 1$ .

## Example

$P(n)$  is the property  $\sum_{i=0}^n i = \frac{n(n+1)}{2}$ .

## Example

$P(n)$  is the property  $\sum_{i=0}^n i = \frac{n(n+1)}{2}$ .

**Base case:**  $\sum_{i=0}^0 i = 0 = \frac{0(0+1)}{2}$ .

## Example

$P(n)$  is the property  $\sum_{i=0}^n i = \frac{n(n+1)}{2}$ .

**Base case:**  $\sum_{i=0}^0 i = 0 = \frac{0(0+1)}{2}$ .

**Inductive case:** Assume  $P(k)$ :  $\sum_{i=1}^k i = \frac{k(k+1)}{2}$ .

## Example

$P(n)$  is the property  $\sum_{i=0}^n i = \frac{n(n+1)}{2}$ .

**Base case:**  $\sum_{i=0}^0 i = 0 = \frac{0(0+1)}{2}$ .

**Inductive case:** Assume  $P(k)$ :  $\sum_{i=1}^k i = \frac{k(k+1)}{2}$ .

$$\text{Then } \sum_{i=1}^{k+1} i = \sum_{i=1}^k i + (k+1)$$

## Example

$P(n)$  is the property  $\sum_{i=0}^n i = \frac{n(n+1)}{2}$ .

**Base case:**  $\sum_{i=0}^0 i = 0 = \frac{0(0+1)}{2}$ .

**Inductive case:** Assume  $P(k)$ :  $\sum_{i=1}^k i = \frac{k(k+1)}{2}$ .

$$\begin{aligned} \text{Then } \sum_1^{k+1} i &= \sum_1^k i + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1) \end{aligned}$$

## Example

$P(n)$  is the property  $\sum_{i=0}^n i = \frac{n(n+1)}{2}$ .

**Base case:**  $\sum_{i=0}^0 i = 0 = \frac{0(0+1)}{2}$ .

**Inductive case:** Assume  $P(k)$ :  $\sum_{i=1}^k i = \frac{k(k+1)}{2}$ .

$$\begin{aligned} \text{Then } \sum_{i=1}^{k+1} i &= \sum_{i=1}^k i + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \end{aligned}$$

## Example

$P(n)$  is the property  $\sum_{i=0}^n i = \frac{n(n+1)}{2}$ .

**Base case:**  $\sum_{i=0}^0 i = 0 = \frac{0(0+1)}{2}$ .

**Inductive case:** Assume  $P(k)$ :  $\sum_{i=1}^k i = \frac{k(k+1)}{2}$ .

$$\begin{aligned} \text{Then } \sum_{i=1}^{k+1} i &= \sum_{i=1}^k i + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \\ &= \frac{(k+1)(k+2)}{2} \end{aligned}$$

## Example

$P(n)$  is the property  $\sum_{i=0}^n i = \frac{n(n+1)}{2}$ .

**Base case:**  $\sum_{i=0}^0 i = 0 = \frac{0(0+1)}{2}$ .

**Inductive case:** Assume  $P(k)$ :  $\sum_{i=1}^k i = \frac{k(k+1)}{2}$ .

$$\begin{aligned} \text{Then } \sum_{i=1}^{k+1} i &= \sum_{i=1}^k i + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \\ &= \frac{(k+1)(k+2)}{2} \end{aligned}$$

Since  $P(0)$  holds and  $P(k) \rightarrow P(k+1)$ , it follows that  $P(n)$  holds for all natural numbers  $n$ .

# Induction

---

Mathematical induction is a special case of a more general principle.

In general, whenever a set can be defined inductively, induction can be used to prove things about elements in the set.

What is an inductive definition?

# Induction

---

Let  $U$  be some *universal* set, and suppose we wish to define some subset  $C$  of  $U$  inductively. This can be done as follows.

- $B$  is an initial subset of  $U$ .
- $F$  is a family of relations on  $U$ .

Informally,  $B$  is the base case for our inductive definition. These are the elements we are starting with. The set  $F$  describes how to obtain new elements from old elements. The set  $C$  is the set of all elements that are either in  $B$  or can be obtained from  $B$  using the relations in  $F$ .

## Example

The natural numbers  $\mathcal{N}$  can be defined as follows:

Let  $U$  be the set of all real numbers,  $B = \{0\}$  and  $F = \{succ\}$ , where *succ* is the successor function defined as  $succ = \{(x, x + 1)\}$ , or  $succ(x) = x + 1$ .

# Induction

---

## General Inductive Definition

- $U$  is a universal set
- $B$  is an initial subset of  $U$ .
- $F$  is a family of relations on  $U$ .

How do we use this to obtain the desired set  $C$ ?

We can define  $C^*$ , the *top-down* version of  $C$  as follows:

- A set  $S$  is *closed* under  $F$  iff for each  $R \in F$ , if  $x_1, \dots, x_n \in S$  and  $R(x_1, \dots, x_n, y)$  for some  $y \in U$ , then  $y \in S$ .
- A set  $S$  is *inductive* if  $B \subseteq S$  and  $S$  is closed under  $F$ .
- The set  $C^*$  is defined as the intersection of all inductive subsets of  $U$ .

This is *top-down* because we take something too big (inductive sets) and use their intersection to construct the desired set.

# Induction

---

## Example

Recall our inductive definition of the natural numbers:

- $U = \mathcal{R}$ , where  $\mathcal{R}$  is the set of real numbers.
- $B = \{0\}$
- $F = \{succ\}$ , where  $succ(x) = x + 1$ .

$\mathcal{R}$  is closed under  $succ$  and is also inductive because  $0 \in \mathcal{R}$ .

# Induction

---

## Example

Recall our inductive definition of the natural numbers:

- $U = \mathcal{R}$ , where  $\mathcal{R}$  is the set of real numbers.
- $B = \{0\}$
- $F = \{succ\}$ , where  $succ(x) = x + 1$ .

$\mathcal{R}$  is closed under  $succ$  and is also inductive because  $0 \in \mathcal{R}$ .

What about

- The set of all (positive and negative) integers?

# Induction

---

## Example

Recall our inductive definition of the natural numbers:

- $U = \mathcal{R}$ , where  $\mathcal{R}$  is the set of real numbers.
- $B = \{0\}$
- $F = \{succ\}$ , where  $succ(x) = x + 1$ .

$\mathcal{R}$  is closed under  $succ$  and is also inductive because  $0 \in \mathcal{R}$ .

What about

- The set of all (positive and negative) integers?
- The set  $\{1, 2, 3, \dots\}$ ?

# Induction

---

## Example

Recall our inductive definition of the natural numbers:

- $U = \mathcal{R}$ , where  $\mathcal{R}$  is the set of real numbers.
- $B = \{0\}$
- $F = \{succ\}$ , where  $succ(x) = x + 1$ .

$\mathcal{R}$  is closed under  $succ$  and is also inductive because  $0 \in \mathcal{R}$ .

What about

- The set of all (positive and negative) integers?
- The set  $\{1, 2, 3, \dots\}$ ?
- The set  $\{0.5, 1.5, 2.5, \dots\}$ ?

# Induction

---

## Example

Recall our inductive definition of the natural numbers:

- $U = \mathcal{R}$ , where  $\mathcal{R}$  is the set of real numbers.
- $B = \{0\}$
- $F = \{succ\}$ , where  $succ(x) = x + 1$ .

$\mathcal{R}$  is closed under  $succ$  and is also inductive because  $0 \in \mathcal{R}$ .

What about

- The set of all (positive and negative) integers?
- The set  $\{1, 2, 3, \dots\}$ ?
- The set  $\{0.5, 1.5, 2.5, \dots\}$ ?

It is not hard to see that  $\mathcal{N}$  is the smallest set which is closed and inductive.

# Induction

## General Inductive Definition

- $U$  is a universal set
- $B$  is an initial subset of  $U$ .
- $F$  is a family of relations on  $U$ .

We can define  $C_*$ , the *bottom-up* version of  $C$  as follows:

- A *construction sequence* is a finite sequence  $\langle x_0, \dots, x_n \rangle$  of elements of  $U$  such that for each  $i < n$ , one of the following holds:
  - $x_i \in B$
  - $R(x_{j_0}, \dots, x_{j_m})$  where  $0 \leq j_0 < \dots < j_m = i$  for some  $R \in F$
- Let  $C_n$  be the set of elements  $x$  such that some construction sequence of length  $n$  ends with  $x$ . Note that  $C_1 = B$ .
- The set  $C_*$  is defined as the set of all elements  $x$  such that some construction sequence ends with  $x$  (i.e.  $C_* = \bigcup C_n$ ).

This is *bottom-up* because we show how to construct each element and then put them together to get  $C_*$ .

# Induction

---

## Example

Recall our inductive definition of the natural numbers:

- $U = \mathcal{R}$ , where  $\mathcal{R}$  is the set of real numbers.
- $B = \{0\}$
- $F = \{succ\}$ , where  $succ(x) = x + 1$ .

The construction sequences of this definition are

- $\langle 0 \rangle$
- $\langle 0, 1 \rangle$
- $\langle 0, 1, 2 \rangle$
- ...

The set of all the last items in the construction sequences gives  $\mathcal{N}$ .

## Induction

---

As you may have suspected, given any inductive definition, it is always the case that  $C^* = C_*$ .

### Proof

$C^* \subseteq C_*$ : We show that  $C_*$  is inductive. Clearly  $B \subseteq C_*$  since  $C_1 = B$ . Suppose  $x_1, \dots, x_n \in C_*$  and  $R(x_1, \dots, x_n, y)$  for some  $R \in F$  then we can concatenate the construction sequences for each  $x_i$  and append  $y$  to get a valid construction sequence for  $y$ . Thus,  $C_*$  is closed under  $F$ , and thus  $C_*$  is inductive. Since  $C^*$  is the intersection of all inductive sets, it follows that  $C^* \subseteq C_*$ .

$C_* \subseteq C^*$ : We show that if  $\langle x_0, \dots, x_n \rangle$  is any construction sequence, then  $x_n \in C^*$ . We use ordinary induction on  $n$ . For the base case, when  $n = 0$ , we have that  $x_0 \in B$ , so it follows that  $x_0 \in C^*$ . For the induction case, consider a sequence  $\langle x_0, \dots, x_{n+1} \rangle$ . We know that  $R(x_{j_0}, \dots, x_{j_m})$  where  $0 \leq j_0 < \dots < j_m = n + 1$  for some  $R \in F$ , but by the induction hypothesis, each  $x_{j_i} \in C^*$  for  $i < m$ , so, since  $C^*$  is closed under  $F$  it follows that  $x_{n+1} \in C^*$ .

# Induction

---

Since  $C_* = C^*$ , we can call the set simply  $C$ . We also refer to it as the set *generated from  $B$  by  $F$* . The following principle will often be used to prove theorems.

## Induction Principle

If  $C$  is the set generated from  $B$  by  $F$  and  $S$  is a set which includes  $B$  and is closed under  $F$  (i.e.  $S$  is inductive), then  $C \subseteq S$ .

**Proof** Since  $S$  is inductive, and  $C = C^*$  is the intersection of all inductive sets, it follows that  $C \subseteq S$ .

We will give an example of applying this principle shortly.

## Propositional Logic: Well-Formed Formulas

We can now use a formal inductive definition to define the set  $W$  of well-formed formulas in propositional logic.

- $U =$
- $B =$
- $F =$

## Propositional Logic: Well-Formed Formulas

We can now use a formal inductive definition to define the set  $W$  of well-formed formulas in propositional logic.

- $U =$  the set of all expressions.
- $B =$
- $F =$

## Propositional Logic: Well-Formed Formulas

We can now use a formal inductive definition to define the set  $W$  of well-formed formulas in propositional logic.

- $U =$  the set of all expressions.
- $B =$  the set of expressions consisting of a single propositional symbol.
- $F =$

# Propositional Logic: Well-Formed Formulas

We can now use a formal inductive definition to define the set  $W$  of well-formed formulas in propositional logic.

- $U$  = the set of all expressions.
- $B$  = the set of expressions consisting of a single propositional symbol.
- $F$  = the set of formula-building operations:
  - $\mathcal{E}_{\neg}(\alpha) = (\neg\alpha)$
  - $\mathcal{E}_{\wedge}(\alpha, \beta) = (\alpha \wedge \beta)$
  - $\mathcal{E}_{\vee}(\alpha, \beta) = (\alpha \vee \beta)$
  - $\mathcal{E}_{\rightarrow}(\alpha, \beta) = (\alpha \rightarrow \beta)$
  - $\mathcal{E}_{\leftrightarrow}(\alpha, \beta) = (\alpha \leftrightarrow \beta)$

## Propositional Logic: Well-Formed Formulas

Given our inductive definition of well-formed formulas, we can use the induction principle to prove things about the set  $W$  of well-formed formulas.

### Example

Prove that any *wff* has the same number of left parentheses and right parentheses.

### Proof

Let  $l(\alpha)$  be the number of left parentheses and  $r(\alpha)$  the number of right parentheses in an expression  $\alpha$ . Let  $S$  be the set of all expressions  $\alpha$  such that  $l(\alpha) = r(\alpha)$ . We wish to show that  $W \subseteq S$ . This follows from the induction principle if we can show that  $S$  is inductive.

### Base Case:

We must show that  $B \subseteq S$ . Recall that  $B$  is the set of expressions consisting of a single propositional symbol. It is clear that for such expressions,  $l(\alpha) = r(\alpha) = 0$ .

## Propositional Logic: Well-Formed Formulas

### Inductive Case:

We must show that  $S$  is closed under each formula-building operator in  $F$ .

# Propositional Logic: Well-Formed Formulas

## Inductive Case:

We must show that  $S$  is closed under each formula-building operator in  $F$ .

- $\mathcal{E}_{\neg}$

Suppose  $\alpha \in S$ . We know that  $\mathcal{E}_{\neg}(\alpha) = (\neg\alpha)$ . It follows that  $l(\mathcal{E}_{\neg}(\alpha)) = 1 + l(\alpha)$  and  $r(\mathcal{E}_{\neg}(\alpha)) = 1 + r(\alpha)$ .

But because  $\alpha \in S$ , we know that  $l(\alpha) = r(\alpha)$ , so it follows that  $l(\mathcal{E}_{\neg}(\alpha)) = r(\mathcal{E}_{\neg}(\alpha))$ , and thus  $\mathcal{E}_{\neg}(\alpha) \in S$ .

# Propositional Logic: Well-Formed Formulas

## Inductive Case:

We must show that  $S$  is closed under each formula-building operator in  $F$ .

- $\mathcal{E}_{\neg}$

Suppose  $\alpha \in S$ . We know that  $\mathcal{E}_{\neg}(\alpha) = (\neg\alpha)$ . It follows that  $l(\mathcal{E}_{\neg}(\alpha)) = 1 + l(\alpha)$  and  $r(\mathcal{E}_{\neg}(\alpha)) = 1 + r(\alpha)$ .

But because  $\alpha \in S$ , we know that  $l(\alpha) = r(\alpha)$ , so it follows that  $l(\mathcal{E}_{\neg}(\alpha)) = r(\mathcal{E}_{\neg}(\alpha))$ , and thus  $\mathcal{E}_{\neg}(\alpha) \in S$ .

- $\mathcal{E}_{\wedge}$

Suppose  $\alpha, \beta \in S$ . We know that  $\mathcal{E}_{\wedge}(\alpha, \beta) = (\alpha \wedge \beta)$ . Thus  $l(\mathcal{E}_{\wedge}(\alpha, \beta)) = 1 + l(\alpha) + l(\beta)$  and  $r(\mathcal{E}_{\wedge}(\alpha, \beta)) = 1 + r(\alpha) + r(\beta)$ .

As before, it follows from the inductive hypothesis that  $\mathcal{E}_{\wedge}(\alpha, \beta) \in S$

# Propositional Logic: Well-Formed Formulas

## Inductive Case:

We must show that  $S$  is closed under each formula-building operator in  $F$ .

- $\mathcal{E}_{\neg}$   
Suppose  $\alpha \in S$ . We know that  $\mathcal{E}_{\neg}(\alpha) = (\neg\alpha)$ . It follows that  $l(\mathcal{E}_{\neg}(\alpha)) = 1 + l(\alpha)$  and  $r(\mathcal{E}_{\neg}(\alpha)) = 1 + r(\alpha)$ .  
But because  $\alpha \in S$ , we know that  $l(\alpha) = r(\alpha)$ , so it follows that  $l(\mathcal{E}_{\neg}(\alpha)) = r(\mathcal{E}_{\neg}(\alpha))$ , and thus  $\mathcal{E}_{\neg}(\alpha) \in S$ .
- $\mathcal{E}_{\wedge}$   
Suppose  $\alpha, \beta \in S$ . We know that  $\mathcal{E}_{\wedge}(\alpha, \beta) = (\alpha \wedge \beta)$ . Thus  $l(\mathcal{E}_{\wedge}(\alpha, \beta)) = 1 + l(\alpha) + l(\beta)$  and  $r(\mathcal{E}_{\wedge}(\alpha, \beta)) = 1 + r(\alpha) + r(\beta)$ .  
As before, it follows from the inductive hypothesis that  $\mathcal{E}_{\wedge}(\alpha, \beta) \in S$ .
- The arguments for  $\mathcal{E}_{\vee}$ ,  $\mathcal{E}_{\rightarrow}$ , and  $\mathcal{E}_{\leftrightarrow}$  are exactly analogous to the one for  $\mathcal{E}_{\wedge}$ .

# Propositional Logic: Well-Formed Formulas

## Inductive Case:

We must show that  $S$  is closed under each formula-building operator in  $F$ .

- $\mathcal{E}_{\neg}$   
Suppose  $\alpha \in S$ . We know that  $\mathcal{E}_{\neg}(\alpha) = (\neg\alpha)$ . It follows that  $l(\mathcal{E}_{\neg}(\alpha)) = 1 + l(\alpha)$  and  $r(\mathcal{E}_{\neg}(\alpha)) = 1 + r(\alpha)$ .  
But because  $\alpha \in S$ , we know that  $l(\alpha) = r(\alpha)$ , so it follows that  $l(\mathcal{E}_{\neg}(\alpha)) = r(\mathcal{E}_{\neg}(\alpha))$ , and thus  $\mathcal{E}_{\neg}(\alpha) \in S$ .
- $\mathcal{E}_{\wedge}$   
Suppose  $\alpha, \beta \in S$ . We know that  $\mathcal{E}_{\wedge}(\alpha, \beta) = (\alpha \wedge \beta)$ . Thus  $l(\mathcal{E}_{\wedge}(\alpha, \beta)) = 1 + l(\alpha) + l(\beta)$  and  $r(\mathcal{E}_{\wedge}(\alpha, \beta)) = 1 + r(\alpha) + r(\beta)$ .  
As before, it follows from the inductive hypothesis that  $\mathcal{E}_{\wedge}(\alpha, \beta) \in S$ .
- The arguments for  $\mathcal{E}_{\vee}$ ,  $\mathcal{E}_{\rightarrow}$ , and  $\mathcal{E}_{\leftrightarrow}$  are exactly analogous to the one for  $\mathcal{E}_{\wedge}$ .

Since  $S$  includes  $B$  and is closed under the operations in  $F$ , it is inductive. It follows by the induction principle that  $W \subseteq S$ .

## Propositional Logic: Well-Formed Formulas

Now we can return to the question of how to prove that an expression is not a *wff*.

How do we know that  $((\leftrightarrow)A_5)$  is not a *wff*?

## Propositional Logic: Well-Formed Formulas

Now we can return to the question of how to prove that an expression is not a *wff*.

How do we know that  $(( \leftrightarrow )A_5$  is not a *wff*?

It does not have the same number of left and right parentheses.

It follows from the theorem we just proved that  $(( \leftrightarrow )A_5$  is not a *wff*.

# Recursion

Suppose we wish to define a function whose domain is an inductively defined set. The natural way to do this is using *recursion*.

Assume an inductive definition with universal set  $U$ , base set  $B \subseteq U$ , and a family of functions  $F$  which take one or more arguments from  $U$  and return an element of  $U$  (note that this is less general than the family of relations we have been dealing with). Let  $C$  be the set defined by this definition.

Now, we wish to define a function  $h$  whose domain is  $C$ . We can do this as follows.

- For each  $x \in B$ , explicitly define  $h(x)$ .
- For each function  $f(x_0, \dots, x_n) \in F$ , give a rule for computing  $h(f(x_0, \dots, x_n))$  given  $h(x_0), \dots, h(x_n)$ .

# Recursion

---

## Examples

Recall our inductive definition of  $\mathcal{N}$ :  $U = \mathcal{R}$ ,  $B = \{0\}$ ,  $F = \{succ\}$ . Suppose we wish to define the factorial function *fact*. We can do this as follows:

- $fact(0) = 1$
- $fact(succ(n)) = (n + 1) \times fact(n)$

# Recursion

---

## Examples

Recall our inductive definition of  $\mathcal{N}$ :  $U = \mathcal{R}$ ,  $B = \{0\}$ ,  $F = \{succ\}$ . Suppose we wish to define the factorial function *fact*. We can do this as follows:

- $fact(0) = 1$
- $fact(succ(n)) = (n + 1) \times fact(n)$

In general, however, a recursive definition like this one does not guarantee the existence of such a function. Consider the following alternative inductive definition of  $\mathcal{N}$ :  $U = \mathcal{R}$ ,  $B = \{0\}$ ,  $F = \{succ, mult\}$ , where  $mult(x, y) = x \times y$ . Now we define a function *h* as follows:

- $h(0) = 0$
- $h(succ(n)) = h(n) + 2$
- $h(mult(m, n)) = h(m) \times h(n)$

What is  $h(1)$ ?

# Recursion

---

## Examples

Recall our inductive definition of  $\mathcal{N}$ :  $U = \mathcal{R}$ ,  $B = \{0\}$ ,  $F = \{succ\}$ . Suppose we wish to define the factorial function *fact*. We can do this as follows:

- $fact(0) = 1$
- $fact(succ(n)) = (n + 1) \times fact(n)$

In general, however, a recursive definition like this one does not guarantee the existence of such a function. Consider the following alternative inductive definition of  $\mathcal{N}$ :  $U = \mathcal{R}$ ,  $B = \{0\}$ ,  $F = \{succ, mult\}$ , where  $mult(x, y) = x \times y$ . Now we define a function *h* as follows:

- $h(0) = 0$
- $h(succ(n)) = h(n) + 2$
- $h(mult(m, n)) = h(m) \times h(n)$

What is  $h(1)$ ?

How can we be sure that a recursive definition is *well-defined*?

# Recursion

To ensure well-defined recursive definitions, an inductive definition of a set  $C$  must satisfy some additional constraints.

- The restriction of each function  $f \in F$  to  $C$  must be one-to-one.
- The range of the same restriction of each function in  $F$  is disjoint from the range of all other restricted functions in  $F$  and from  $B$ .

If these conditions are met, we say that  $C$  is *freely* generated from  $B$  by  $F$ .

## Recursion Theorem

Suppose  $C$  is freely generated from  $B$  by  $F$ . Suppose also that  $V$  is a set and  $h$  is a function from  $B$  to  $V$ . Suppose further that for each function  $f : U^n \rightarrow U$  in  $F$ , there is a corresponding function  $\bar{f} : V^n \rightarrow V$ . Then there exists a unique function  $\bar{h} : C \rightarrow V$  such that

- for  $x \in B$ ,  $\bar{h}(x) = h(x)$ , and
- for each  $f : U^n \rightarrow U$  in  $F$  and  $x_1, \dots, x_n \in C$ ,  
$$\bar{h}(f(x_1, \dots, x_n)) = \bar{f}(\bar{h}(x_1), \dots, \bar{h}(x_n)).$$

## Recursion

Given  $C$  freely generated from  $B$  by  $F$ , show that  $h : B \rightarrow V$  and  $\bar{f} : V^n \rightarrow V$  for each  $f : U^n \rightarrow U$  in  $F$  determine a unique function  $\bar{h} : C \rightarrow V$ .

### Proof sketch

A function  $g : D \rightarrow E$  is called *acceptable* if  $D \subseteq C$ ,  $E \subseteq V$ , and

- for  $x \in B \cap D$ ,  $g(x) = h(x)$ , and
- for each  $f : U^n \rightarrow U$  in  $F$  and  $x_1, \dots, x_n \in C$ , if  $f(x_1, \dots, x_n) \in D$ , then  $x_1, \dots, x_n \in D$  and  $g(f(x_1, \dots, x_n)) = \bar{f}(g(x_1), \dots, g(x_n))$ .

Let  $K$  be the collection of all acceptable functions, and let  $\bar{h}$  be the union of  $K$ . Then  $\bar{h}$  meets our requirements. Specifically,

- $\bar{h}$  is a function.
- $\bar{h}$  is an acceptable function.
- The domain of  $\bar{h}$  is all of  $C$ .
- $\bar{h}$  is unique.

Details omitted (requires repeated use of induction principle).

## Propositional Logic: Semantics

Intuitively, given a *wff*  $\alpha$  and a value (either *true* or *false*) for each propositional symbol in  $\alpha$ , we should be able to determine the value of  $\alpha$ .

How do we make this precise?

Let  $v$  be a function from  $B$  to  $\{0, 1\}$ , where  $0$  represents *false* and  $1$  represents *true*. Recall that in the inductive definition of *wff*'s,  $B$  contains the propositional symbols.

Now, we define  $\bar{v}$ , a function from  $W$  to  $\{0, 1\}$  as follows

- For each propositional symbol  $A_i$ ,  $\bar{v}(A_i) = v(A_i)$ .
- $\bar{v}(\mathcal{E}_{\neg}(\alpha)) = 1 - \bar{v}(\alpha)$
- $\bar{v}(\mathcal{E}_{\wedge}(\alpha, \beta)) = \min(\bar{v}(\alpha), \bar{v}(\beta))$
- $\bar{v}(\mathcal{E}_{\vee}(\alpha, \beta)) = \max(\bar{v}(\alpha), \bar{v}(\beta))$
- $\bar{v}(\mathcal{E}_{\rightarrow}(\alpha, \beta)) = \max(1 - \bar{v}(\alpha), \bar{v}(\beta))$
- $\bar{v}(\mathcal{E}_{\leftrightarrow}(\alpha, \beta)) = 1 - |\bar{v}(\alpha) - \bar{v}(\beta)|$

The recursion theorem guarantees that  $\bar{v}$  is well-defined.

## Propositional Logic: Semantics

Intuitively, given a *wff*  $\alpha$  and a value (either *true* or *false*) for each propositional symbol in  $\alpha$ , we should be able to determine the value of  $\alpha$ .

How do we make this precise?

Let  $v$  be a function from  $B$  to  $\{0, 1\}$ , where  $0$  represents *false* and  $1$  represents *true*. Recall that in the inductive definition of *wff*'s,  $B$  contains the propositional symbols.

Now, we define  $\bar{v}$ , a function from  $W$  to  $\{0, 1\}$  as follows

- For each propositional symbol  $A_i$ ,  $\bar{v}(A_i) = v(A_i)$ .
- $\bar{v}(\mathcal{E}_{\neg}(\alpha)) = 1 - \bar{v}(\alpha)$
- $\bar{v}(\mathcal{E}_{\wedge}(\alpha, \beta)) = \min(\bar{v}(\alpha), \bar{v}(\beta))$
- $\bar{v}(\mathcal{E}_{\vee}(\alpha, \beta)) = \max(\bar{v}(\alpha), \bar{v}(\beta))$
- $\bar{v}(\mathcal{E}_{\rightarrow}(\alpha, \beta)) = \max(1 - \bar{v}(\alpha), \bar{v}(\beta))$
- $\bar{v}(\mathcal{E}_{\leftrightarrow}(\alpha, \beta)) = 1 - |\bar{v}(\alpha) - \bar{v}(\beta)|$

The recursion theorem guarantees that  $\bar{v}$  is well-defined... under what conditions?

# Propositional Logic: Semantics

## Unique Readability Theorem

Given our inductive definition of the set  $W$  of *wff*'s,  $W$  is freely generated from  $B$  by  $F$ . Specifically, the restriction of each operation in  $F$  to  $W$  is one-to-one and has a range disjoint from the range of the other restricted operations in  $F$  and from  $B$ .

First we need the following lemma.

## Lemma

Any proper initial segment of a *wff* contains an excess of left parentheses, and is therefore not a *wff*.

## Proof

Let  $S$  be the set of *wff*'s which have this property. We will show that  $S$  is inductive. First note that the elements of  $B$  all consist of a single symbol so it is not possible to construct a proper initial segment of any of them. Thus,  $B \subseteq S$  vacuously.

# Propositional Logic: Semantics

## Proof, continued

To show that  $S$  is closed under  $\mathcal{E}_\wedge$ , suppose that  $\alpha, \beta \in S$  and consider a proper initial segment of  $\mathcal{E}_\wedge(\alpha, \beta)$ . There are 6 possibilities:

- (
- $(\alpha_0$ , where  $\alpha_0$  is a proper initial segment of  $\alpha$ .
- $(\alpha$
- $(\alpha \wedge$
- $(\alpha \wedge \beta_0$ , where  $\beta_0$  is a proper initial segment of  $\beta$ .
- $(\alpha \wedge \beta$

By using the inductive hypothesis and the fact (proved earlier) that all *wff*'s have the same number of left and right parentheses, each of these cases can be seen to have more left parentheses than right. Thus,  $\mathcal{E}_\wedge(\alpha, \beta) \in S$ .

The cases for  $\mathcal{E}_\neg$ ,  $\mathcal{E}_\vee$ ,  $\mathcal{E}_\rightarrow$ , and  $\mathcal{E}_{\leftrightarrow}$  are similar.

# Propositional Logic: Semantics

## Unique Readability Theorem

Given our inductive definition of the set  $W$  of *wff*'s,  $W$  is freely generated from  $B$  by  $F$ . Specifically, the restriction of each operation in  $F$  to  $W$  is one-to-one and has a range disjoint from the range of the other restricted operations in  $F$  and from  $B$ .

## Proof

To show that the operation  $\mathcal{E}_\wedge$  restricted to  $W$  is one-to-one, suppose that  $(\alpha \wedge \beta) = (\gamma \wedge \delta)$ , where  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  are *wff*'s.

Since both start with a left parenthesis, it follows that  $(\alpha \wedge \beta) = (\gamma \wedge \delta)$ . Since  $\alpha$  and  $\gamma$  are *wff*'s, the previous lemma implies that neither one can be a prefix of the other, and thus  $\alpha = \gamma$ . The same argument then shows that  $\beta = \delta$ .

Similar arguments can be applied to show that the other operations are one-to-one and that their ranges are all disjoint.