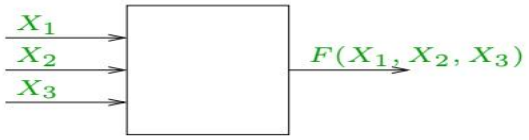
Propositional Logic: Motivation

Consider an electrical device having n inputs and one output. Assume that to each input we apply a signal that is either 1 or 0, and that this uniquely determines whether the output is 1 or 0.

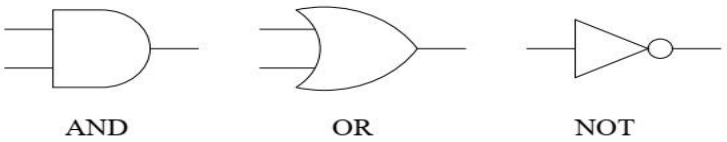


The behavior of such a device is described by a Boolean function:

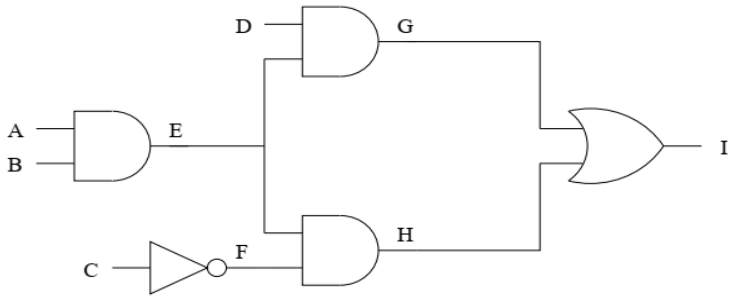
$F(X_1, \ldots, X_n) =$ the output signal given the input signals $X_1, \ldots, X_n$.

We call such a device a *Boolean gate*.

The most common Boolean gates are *AND*, *OR*, and *NOT* gates.



The inputs and outputs of Boolean gates can be connected together to form a *combinational Boolean circuit*.

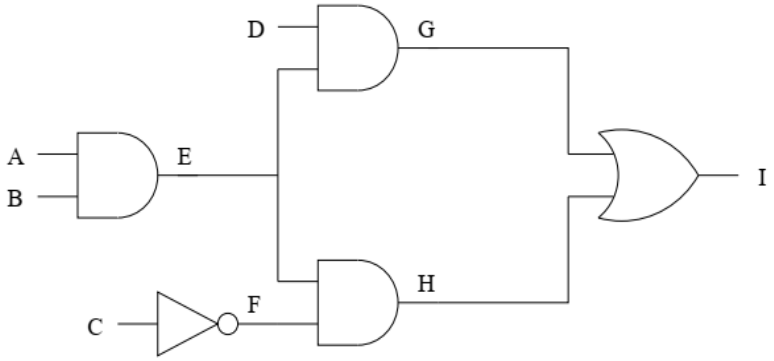


A combinational Boolean circuit corresponds to a *directed acyclic graph* (DAG) whose leaves are *inputs* and each of whose nodes is labeled with the name of a Boolean gate. One or more of the nodes may be identified as outputs.

A common question with Boolean circuits is whether it is possible to set an output to true (e.g. when the output represents an *error* signal).

Suppose your job was to find out if the output of a large Boolean circuit could ever be true. How would you do it?

The inputs and outputs of Boolean gates can be connected together to form a *combinational Boolean circuit*.



A combinational Boolean circuit corresponds to a *directed acyclic graph* (DAG) whose leaves are *inputs* and each of whose nodes is labeled with the name of a Boolean gate. One or more of the nodes may be identified as outputs.

A common question with Boolean circuits is whether it is possible to set an output to true (e.g. when the output represents an *error* signal).

Suppose your job was to find out if the output of a large Boolean circuit could ever be true. How would you do it?

Propositional Logic provides the formalism to answer such questions.

Propositional (or Sentential) 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 logics (such as *first-order logic*, also called *predicate logic*, which we'll discuss later.)

What is 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 defines the *well-formed* expressions.

Semantics

- Gives meaning to well-formed expressions
- Formal notions of induction and recursion are required to provide a rigorous semantics.

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	nth propositional symbol	
...		

Alphabet

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

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

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

Examples

$<(, A_1, \wedge, A_3,)>$
$<(, (, \neg, A_1,), \rightarrow, A_2,)>$
$<),), \leftrightarrow,), A_5>$

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

Examples

$<(, A_1, \wedge, A_3,)>$	$(A_1 \wedge A_3)$
$<(, (, \neg, A_1,), \rightarrow, A_2,)>$	$((\neg A_1) \rightarrow A_2)$
$<),), \leftrightarrow,), A_5>$	$)) \leftrightarrow) A_5$

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.

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

Examples

$<(, A_1, \wedge, A_3,)>$	$(A_1 \wedge A_3)$
$<(, (, \neg, A_1,), \rightarrow, A_2,)>$	$((\neg A_1) \rightarrow A_2)$
$<),), \leftrightarrow,), A_5>$	$)) \leftrightarrow) A_5$

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.

Not all expressions make sense. Part of the job of defining a syntax is to *restrict* the kinds of expressions that will be allowed.

We define the set W of *well-formed formulas* (*wffs*) as follows.

(a) Every expression consisting of a single propositional symbol is in W.

(b) If α and β 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.

We define the set W of *well-formed formulas* (*wffs*) as follows.

(a) Every expression consisting of a single propositional symbol is in W.

(b) If α and β 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*?

We define the set W of *well-formed formulas* (*wffs*) as follows.

(a) Every expression consisting of a single propositional symbol is in W.

(b) If α and β 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 $P(k)$: $\sum_{i=0}^{k} i = \frac{k(k+1)}{2}$.

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 $P(k)$: $\sum_{i=0}^{k} i = \frac{k(k+1)}{2}$.

Then $\sum_{0}^{k+1} i = \sum_{0}^{k} i + (k+1)$

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 $P(k)$: $\sum_{i=0}^{k} i = \frac{k(k+1)}{2}$.

Then $\sum_{0}^{k+1} i = \sum_{0}^{k} i + (k+1)$

$= \frac{k(k+1)}{2} + (k+1)$

Example

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

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

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 $P(k)$: $\sum_{i=0}^{k} i = \frac{k(k+1)}{2}$.

$$\begin{aligned} \text{Then } \sum_{0}^{k+1} i &= \sum_{0}^{k} i + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \\ &= \frac{(k+1)(k+2)}{2} \end{aligned}$$

Inductive case: Assume $P(k)$: $\sum_{i=0}^{k} i = \frac{k(k+1)}{2}$.

$$\begin{aligned} \text{Then } \sum_{0}^{k+1} i &= \sum_{0}^{k} i + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \end{aligned}$$

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 $P(k)$: $\sum_{i=0}^{k} i = \frac{k(k+1)}{2}$.

$$\begin{aligned} \text{Then } \sum_{0}^{k+1} i &= \sum_{0}^{k} i + (k+1) \\ &= \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \\ &= \frac{(k+1)(k+2)}{2} \end{aligned}$$

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?

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 functions 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 functions 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 $succ(x) = x + 1$.

General Inductive Definition

- U is a universal set
- B is an initial subset of U.
- F is a family of functions 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 $f \in F$, if $x_1, \ldots, x_n \in S$ and $f(x_1, \ldots, x_n) = y$ 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.

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 $0 \in \mathcal{R}$.

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 (including negative) integers?

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 (including negative) integers? *closed, inductive*

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 (including negative) integers? *closed, inductive*
- The set $\{1, 2, 3, \ldots\}$?

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 (including negative) integers? *closed, inductive*
- The set $\{1, 2, 3, \ldots\}$? *closed, not inductive*

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 (including negative) integers? *closed, inductive*
- The set $\{1, 2, 3, \ldots\}$? *closed, not inductive*
- The set $\{0.5, 1.5, 2.5, \ldots\}$?

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 (including negative) integers? *closed, inductive*
- The set $\{1, 2, 3, \ldots\}$? *closed, not inductive*
- The set $\{0.5, 1.5, 2.5, \ldots\}$? *closed, not inductive*

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 (including negative) integers? *closed, inductive*
- The set $\{1, 2, 3, \ldots\}$? *closed, not inductive*
- The set $\{0.5, 1.5, 2.5, \ldots\}$? *closed, not inductive*

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

General Inductive Definition

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

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

- A *construction sequence* is a finite sequence $< x_0, \ldots, x_n >$ of elements of U such that for each $i < n$, one of the following holds:
  - $x_i \in B$
  - $f(x_{j_0}, \ldots, x_{j_m}) = x_i$ where $0 \leq j_k < i$ (for $k = 0 \ldots m$) for some $f \in F$
- 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_*.

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 $succ(x) = x + 1$.

Some construction sequences of this definition are

- $< 0 >$
- $< 0, 1, 2, 3 >$
- $< 0, 0, 1, 0 >$
- ...

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

Thought question: *Is it always the case that $C^* = C_*$?*

How can you be sure?

Proof or *Counterexample*

Proof

$C^* \subseteq C_*$: We show that C_* is inductive. Clearly $B \subseteq C_*$ since $C_1 = B$. Suppose $x_1, \ldots, x_n \in C_*$ and $f(x_1, \ldots, x_n) = y$ for some $f \in F$. 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 $< x_0, \ldots, x_n >$ 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 $< x_0, \ldots, x_{n+1} >$. We know that $f(x_{j_0}, \ldots, x_{j_m}) = x_{n+1}$ where $0 \leq j_k < n + 1$ (for each $k = 0 \ldots m$) for some $f \in F$, but by the induction hypothesis, each $x_{j_i} \in C^*$ for $i < m$, so, since C^* is closed under F it follows that $x_{n+1} \in C^*$.

Since $C_* = C^*$, we can call the set simply C. We also refer to it as the set *generated from B by F*.

Now, given any inductive definition of a set, we can prove things about that set using the following principle.

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 often use the induction principle to show that an inductive set C has a particular property. The argument looks like this: (i) Define S to be the subset of U with some property P; (ii) Show that S is inductive.

This proves that $C \subseteq S$ and thus all elements of C have property P.

Example

We can now show how mathematical induction is a special case of the induction principle. Consider again the natural numbers defined inductively as follows:

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

Let $\mathcal{N}$ be the set generated by B from F.

Let S be the set of all real numbers n for which $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

Our earlier proof shows that $0 \in S$ and if $k \in S$ then $k + 1 \in S$. In other words, S is inductive.

Therefore, by the induction principle, $\mathcal{N} \subseteq S$.

Thus, $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$ for all natural numbers $n \in \mathcal{N}$.

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.

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

to

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

to

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

to

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

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 α. Let S be the set of all expressions α 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, $l(\alpha) = r(\alpha) = 0$.

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 $W \subseteq S$.

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. 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 $f(x_0, \ldots, x_n) \in F$, give a rule for computing $h(f(x_0, \ldots, x_n))$ given $h(x_0)$, . . ., $h(x_n)$.

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:

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

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 $h(1)$?

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 must be 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 $f : U^n \to U$ in F, there is a corresponding function $\overline{f} : V^n \to V$. Then there exists a unique function $\overline{h} : C \to V$ such that

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

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

Proof sketch

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

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

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

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

Details omitted (requires repeated use of induction principle). □

Propositional Logic: Well-Formed Formulas

Recall our inductive definition of the set W of well-formed formulas in propositional logic. Given the alphabet $\{(,), \neg, \wedge, \vee, \rightarrow, \leftrightarrow, A_1, A_2, \ldots\}$,

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

Propositional Logic: Well-Formed Formulas

Recall our inductive definition of the set W of well-formed formulas in propositional logic. Given the alphabet $\{(,), \neg, \wedge, \vee, \rightarrow, \leftrightarrow, A_1, A_2, \ldots\}$,

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

Propositional Logic: Well-Formed Formulas

Recall our inductive definition of the set W of well-formed formulas in propositional logic. Given the alphabet $\{(,), \neg, \wedge, \vee, \rightarrow, \leftrightarrow, A_1, A_2, \ldots\}$,

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

Propositional Logic: Well-Formed Formulas

Recall our inductive definition of the set W of well-formed formulas in propositional logic. Given the alphabet $\{(,), \neg, \wedge, \vee, \rightarrow, \leftrightarrow, A_1, A_2, \ldots\}$,

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

Unique Readability Theorem

Theorem

Given our inductive definition of the set W of *wffs*, 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 *wffs* 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.

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 α_0 is a proper initial segment of α.
- $(\alpha$
- $(\alpha\wedge$
- $(\alpha \wedge \beta_0$, where β_0 is a proper initial segment of β_0.
- $(\alpha \wedge \beta$

By using the inductive hypothesis and the fact (proved earlier) that all *wffs* 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.

Unique Readability Theorem

Proof, continued

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

Since both start with a left parenthesis, it follows that $\alpha \wedge \beta) = \gamma \wedge \delta)$. Since α and γ are *wffs*, 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.