

# Boolean Functions and Truth-Functional Completeness

Mathematical Logic I  
Fall 2021  
Robert Rynasiewicz

September 30, 2021

# Other Connectives

So far we've seen five sentential connectives, one monadic ( $\neg$ ) and 4 dyadic ( $\rightarrow$ ,  $\wedge$ ,  $\vee$  and  $\leftrightarrow$ ).

An  $n$ -adic connective is used to take  $n$  sentences to form a new sentence.

Examples from natural language:

## Monadic

- ▶ It is the case that \_\_\_\_.
- ▶ It is necessary that \_\_\_\_.
- ▶ It is possible that \_\_\_\_.
- ▶ It is obligatory that \_\_\_\_.
- ▶ It is permissible that \_\_\_\_.
- ▶ God knows that \_\_\_\_.
- ▶ Trump believes that \_\_\_\_.
- ▶ Putin wants it to be the case that \_\_\_\_.

# Other Natural Language Connectives (cont.)

## Dyadic

- ▶ Neither \_\_\_ nor \_\_\_.
- ▶ Not both \_\_\_ and \_\_\_.
- ▶ \_\_\_ because \_\_\_.
- ▶ That \_\_\_ caused it to be the case that \_\_\_.
- ▶ \_\_\_ in light of the fact that \_\_\_.
- ▶ \_\_\_ despite the fact that \_\_\_.
- ▶ That \_\_\_ truth-functionally entails that \_\_\_.

## Triadic

- ▶ \_\_\_, \_\_\_ and \_\_\_.
- ▶ \_\_\_, \_\_\_ or \_\_\_.

# An Artificial Triadic Connective

The majority connective: # \_\_\_ \_\_\_ \_\_\_

with the recursion clause:  $\bar{v}(\#\alpha\beta\gamma) = M(\alpha, \beta, \gamma)$ , where

$$M(x, y, z) = \begin{cases} 1 & \text{if } x + y + z > 1 \\ 0 & \text{otherwise} \end{cases}$$

$\alpha$	$\beta$	$\gamma$	$\#\alpha\beta\gamma$
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0

# Truth-Functional Connectives

**Def.** An  $n$ -adic connective  $*$  is truth-functional iff the truth value of  $*\alpha_1 \cdots \alpha_n$  is completely determined by the truth values of  $\alpha_1, \dots, \alpha_n$ .

In other words  $*$  is truth-functional just in case you can write out a truth table for  $*\alpha_1 \cdots \alpha_n$ .

Which of the connectives on earlier slides are truth functional and which are not?

- ▶ It is the case that \_\_\_\_\_. **YES**,  $\bar{v}$ (it is the case that  $\alpha$ ) =  $\bar{v}(\alpha)$ .

$\alpha$	it is the case that $\alpha$
1	1
0	0

# Truth-Functional or Not?

It is necessary that \_\_\_\_\_. **NO:**

Let  $\Box$  be 'it is necessary that'.

$0 = 0$	TRUE	The DJ is over 22,000	TRUE
$\Box 0 = 0$	TRUE	$\Box$ the DJ is over 22,000	FALSE

Whether or not

It is necessary that  $P$

is true depends not just on the truth value of  $P$ , but on the what is the case in other "possible worlds".

# The Remainder

- ▶ It is possible that \_\_\_\_\_. **NO**
- ▶ It is obligatory that \_\_\_\_\_. **NO**
- ▶ It is permissible that \_\_\_\_\_. **NO**
- ▶ God knows that \_\_\_\_\_. **YES**
- ▶ Trump believes that \_\_\_\_\_. **NO**
- ▶ Putin wants it to be the case that \_\_\_\_\_. **NO**

## The Remainder (cont.)

- ▶ Neither \_\_\_ nor \_\_\_. **Yes.**

Joint denial, NOR, Symbol:  $\downarrow$

- ▶ Not both \_\_\_ and \_\_\_. **Yes.**

Sheffer Stroke, NAND, Symbol:  $|$

$\alpha$	$\beta$	$(\alpha \downarrow \beta)$	$(\alpha   \beta)$
1	1	0	0
1	0	0	1
0	1	0	1
0	0	1	1

## The Remainder (cont.)

- ▶ \_\_\_ because \_\_\_. **NO.**
- ▶ That \_\_\_ truth-functionally entails that \_\_\_. **NO.**

TRUE: Trump won.

TRUE: Trump won or Hillary won.

TRUE: That Trump won truth-functionally entails that Trump won or Hillary won.

TRUE: Van Hollen won.

FALSE: That Trump won truth-functionally entails that Van Hollen won.

# Boolean Functions

**Def.** For each  $n \in \mathbb{N}$ , an  $n$ -ary Boolean function is an operation  $F : \{0, 1\}^n \rightarrow \{0, 1\}$ .

Each  $n$ -ary truth-functional connective realizes an  $n$ -ary Boolean operation.

Conversely, each  $n$ -ary Boolean operation is realized by some  $n$ -ary truth-functional connective.

Let  $N$  be the Boolean function associated with the connective  $\neg$ :

$x$	$N(x)$
1	0
0	1

# The Familiar Binary Boolean Functions

Since we've seen four dyadic connectives, you're already implicitly familiar with the four corresponding binary Boolean functions.

$x_1$	$x_2$	$C(x_1, x_2)$	$K(x_1, x_2)$	$A(x_1, x_2)$	$E(x_1, x_2)$
1	1	1	1	1	1
1	0	0	0	1	0
0	1	1	0	1	0
0	0	1	0	0	1

# Some Numerology

How many unary Boolean functions / t-f connectives are there? The domain  $\{0, 1\}$  has 2-elements, and each element can receive one of two possible values.

$x$	$B(x)$
1	
0	

How many ways are there to fill in the blanks with 0's and 1's? **4**

$x$	$B_1^1(x)$	$B_2^1(x)$	$B_3^1(x)$	$B_4^1(x)$
1	1	1	0	0
0	1	0	1	0

## Numerology (cont.)

For binary Boolean functions, we have our four familiar and two corresponding to NOR and NAND, respectively. That's 6. But how many total?

$x_1$	$x_2$	$B_7^2(x_1, x_2)$
1	1	
1	0	
0	1	
0	0	

Again, how many ways are there to fill in the blanks with 0's and 1's? **16**

How many distinct ternary Boolean functions? **256**

How many  $n$ -ary for a fixed  $n$ ?  $2^{(2^n)}$

How many of these can we “express” using our five original Boolean functions? E.g., ...

## “Expressing” “New” Boolean Functions

We can “express” the Boolean functions corresponding to NOR and NAND as follows.

$\alpha$	$\beta$	$(\alpha \downarrow \beta)$	$(\neg\alpha \wedge \neg\beta)$	$(\alpha   \beta)$	$\neg(\alpha \wedge \beta)$
1	1	0	0	0	0
1	0	0	0	1	1
0	1	0	0	1	1
0	0	1	1	1	1

We need to give the question, can we “express” such and such Boolean function, some precision.

Toward that end, whatever our stock  $\mathcal{S}$  of sentence letters happens to be, let's impose a well-ordering on  $\mathcal{S}$ . For uncountable languages, this presupposes the Axiom of Choice. But for a countably infinite  $\mathcal{S}$ , we can adopt Enderton's notation:  $A_1, A_2, \dots$ , so that  $A_i$  is the  $i$ -th letter in the ordering.

# Realizing Boolean Functions

**Projection Functions.** For each pair  $(n, i)$  s.t.  $1 \leq i \leq n$  there is a special  $n$ -ary Boolean operation, viz. the  $i$ -th projection function  $I_i^n$  which takes on the value of the  $i$ -th argument, i.e.,

$$I_i^n(x_1, \dots, x_i, \dots, x_n) = x_i.$$

**Def.** Let  $A_n$  be the “largest” sentence letter occurring in  $\mathcal{S}(\alpha)$ . Then  $B_\alpha^n$  is the  $n$ -ary Boolean function s.t. for any valuation  $v : \{A_1, \dots, A_n\} \rightarrow \{0, 1\}$ ,

$$B_\alpha^n(v(A_1), \dots, v(A_n)) = \bar{v}(\alpha)$$

and  $\alpha$  is said to **realize**  $B_\alpha^n$ .

# Examples

- ▶  $B_{(A_1 \vee \neg A_1)}^1(x_1) = A(x_1, N(x_1))$
- ▶  $B_{A_3}^3(x_1, x_2, x_3) = I_3^3(x_1, x_2, x_3)$
- ▶  $B_{(A_1 \wedge A_4)}^4(x_1, x_2, x_3, x_4) = K(I_1^4(x_1, x_2, x_3, x_4), I_4^4(x_1, x_2, x_3, x_4))$
- ▶  $B_{(\neg A_2 \rightarrow A_1)}^2(x_1, x_2) = C(N(I_2^2(x_1, x_2)), I_1^2(x_1, x_2))$
- ▶  $B_{(A_3 \leftrightarrow (A_1 \vee A_3))}^3(x_1, x_2, x_3) =$   
 $E(I_3^3(x_1, x_2, x_3), A(I_1^3(x_1, x_2, x_3), I_3^3(x_1, x_2, x_3)))$

# Characterization Theorem for Entailments

**Theorem.** Let  $A_n$  be the “largest” sentence symbol in  $\mathcal{S}(\alpha)$ ,  $A_m$  the “largest” in  $\mathcal{S}(\beta)$ , and  $k = \max(n, m)$ . Then

1.  $\alpha \models \beta$  iff for all  $x_1, \dots, x_k \in \{0, 1\}$

$$B_\alpha^n(x_1, \dots, x_n) \leq B_\beta^m(x_1, \dots, x_m).$$

2.  $\alpha \models \neg \beta$  iff for all  $x_1, \dots, x_k \in \{0, 1\}$

$$B_\alpha^n(x_1, \dots, x_n) = B_\beta^m(x_1, \dots, x_m).$$

3.  $\alpha \models \alpha$  iff for all  $x_1, \dots, x_n \in \{0, 1\}$

$$B_\alpha^n(x_1, \dots, x_n) = 1.$$

# Proof of the Characterization Theorem

**Proof.** (1)  $\Rightarrow$ . Suppose  $\alpha \models \beta$ . Then for any  $v : \{A_1, \dots, A_k\} \rightarrow \{0, 1\}$ ,  $\bar{v}(\alpha) \leq \bar{v}(\beta)$ . Now choose  $x_1, \dots, x_k$  arbitrarily and let  $v$  be s.t.  $v(A_i) = x_i$  for  $1 \leq i \leq k$ . Then

$$\begin{aligned} B_{\alpha}^n(v(A_1), \dots, v(A_n)) &= \bar{v}(\alpha) \\ &\leq \bar{v}(\beta) \\ &= B_{\beta}^m(v(A_1), \dots, v(A_m)). \end{aligned}$$

$\Leftarrow$ . Suppose  $B_{\alpha}^n(x_1, \dots, x_n) \leq B_{\beta}^m(x_1, \dots, x_m)$ . for all  $x_1, \dots, x_k \in \{0, 1\}$ . Let  $v$  be any valuation on  $\{A_1, \dots, A_k\}$ . Then

$$\begin{aligned} \bar{v}(\alpha) &= B_{\alpha}^n(v(A_1), \dots, v(A_n)) \\ &\leq B_{\beta}^m(v(A_1), \dots, v(A_m)) \\ &= \bar{v}(\beta). \end{aligned}$$

## Proof of the Characterization Theorem (cont.)

(2) From (1) we have that  $\alpha \models \beta$  iff for arbitrary  $x_1, \dots, x_k \in \{0, 1\}$  both

$$B_\alpha^n(x_1, \dots, x_n) \leq B_\beta^m(x_1, \dots, x_m)$$

and

$$B_\beta^m(x_1, \dots, x_m) \leq B_\alpha^n(x_1, \dots, x_n),$$

which is equivalent to

$$B_\alpha^n(x_1, \dots, x_n) = B_\beta^m(x_1, \dots, x_m).$$

(3) From the definition of what it is for  $\alpha$  to realize  $B_\alpha^n$  we have:

$$\begin{aligned} \models \alpha & \text{ iff } \bar{v}(\alpha) = 1 \text{ for all } v \text{ on } \{A_1, \dots, A_n\} \\ & \text{ iff } B_\alpha^n(x_1, \dots, x_n) = 1 \text{ for all } x_1, \dots, x_n \in \{0, 1\}. \end{aligned}$$



# The Million Dollar Question

Earlier we asked if various Boolean functions can be “expressed” by wff’s. Now we attain precision by replacing ‘expressed’ with ‘realized’ and sequentially ask:

## Questions.

1. Can each 1-ary Boolean function be realized by some wff?
2. Can each 2-ary Boolean function be realized by some wff?
3. Can each 3-ary Boolean function be realized by some wff?
- ⋮
- ⋮
- ⋮
- ⋮
- ⋮
- ⋮

## Answers.

1. Yes
2. Yes
3. Yes
- ⋮

# Toward Canonical Disjunctive Normal Form

Towards showing why, it helps to introduce a special syntactic form, viz., **canonical disjunctive normal form**.

We build up the definition of it in stages:

**Def.** A **basic sentence** (a.k.a. a *literal*) is a sentence letter or the negation of a sentence letter.

**Def: Generalized Disjunction.** This is defined by recursion:

$$\bigvee_1 \alpha = \alpha \quad (1)$$

$$\bigvee_{n+1} \alpha_1 \cdots \alpha_{n+1} = \alpha_{n+1} \vee \bigvee_n \alpha_1 \cdots \alpha_n \quad (2)$$

**Def: Generalized Conjunction.** Same as above *mutatis mutandis*.

# Disjunctive Normal Form

**Def.**  $\alpha$  is in **disjunctive normal form** (DNF) iff  $\alpha$  is a (generalized) disjunction of (generalized) conjunctions of basic sentences.

**Exercise.** Which of the following are in DNF?

- ▶  $A_1$ ? Yes.
- ▶  $\neg A_1$  Yes.
- ▶  $A_1 \wedge \neg A_1$  Yes.
- ▶  $A_1 \vee \neg A_1$  Yes.
- ▶  $A_1 \vee (A_2 \wedge A_1)$  Yes.
- ▶  $A_1 \vee (A_2 \wedge A_1) \vee A_1$  Yes.
- ▶  $(A_1 \wedge A_5) \vee (A_1 \wedge \neg A_5) \vee (\neg A_1 \wedge \neg A_5)$  Yes.
- ▶  $(A_1 \wedge A_5) \vee (A_1 \wedge \neg A_5) \vee \neg(A_1 \vee A_5)$  No.

# Canonical Disjunctive Normal Form

We continue to assume that  $\mathcal{S}$  is well ordered (no matter what  $\mathcal{S}$  happens to be).

**Def.** Let  $n$  be the cardinality of  $\mathcal{S}(\alpha)$ . Then  $\alpha$  is in **canonical disjunctive normal form** (CDNF) iff

1.  $\alpha$  is in DNF,
2. each disjunct of  $\alpha$  is truth-functionally satisfiable,
3. no two disjuncts are truth-functionally equivalent,
4. each conjunction  $\kappa$  is an  $n$ -ary conjunction  $\beta_1 \wedge \cdots \wedge \beta_n$  s.t.
  - (a) for each sentence letter  $A_i \in \mathcal{S}(\alpha)$ , either  $A_i$  or  $\neg A_i$  occurs as a  $\beta_j$  in  $\kappa$  and
  - (b) for all  $i, j$  s.t.  $1 \leq i < j \leq n$ ,  $\mathcal{S}(\beta_i) < \mathcal{S}(\beta_j)$ .

# Exercise on Canonical Disjunctive Normal Form

**Exercise.** Which of the following are in CDNF?

- ▶  $A_1$ ? Yes.
- ▶  $\neg A_1$  Yes.
- ▶  $A_1 \wedge \neg A_1$  No.
- ▶  $A_1 \vee \neg A_1$  Yes.
- ▶  $A_1 \vee (A_2 \wedge A_1)$  No.
- ▶  $A_1 \vee (A_2 \wedge A_1) \vee A_1$  No.
- ▶  $(A_1 \wedge A_5) \vee (A_1 \wedge \neg A_5) \vee (\neg A_1 \wedge \neg A_5)$  Yes.
- ▶  $(A_1 \wedge A_5) \vee (A_1 \wedge \neg A_5) \vee \neg(A_1 \vee A_5)$  No.

# The Main Lemma

**Def.** Say that an  $n$ -ary Boolean function is non-degenerate iff  $B(x_1, \dots, x_n) = 1$  for some  $\langle x_1, \dots, x_n \rangle \in \{0, 1\}^n$ .

**Lemma.** Every non-degenerate  $n$ -ary Boolean function is realized by some wff in CDNF in a language with at least  $n$  sentence letters. Every Boolean function is realized by some wff in DNF.

**Illustration.** Consider the following Boolean function  $B$ .

$x_1$	$x_2$	$x_3$	$B(x_1, x_2, x_3)$
1	1	1	
1	1	0	
1	0	1	
1	0	0	
0	1	1	
0	1	0	
0	0	1	
0	0	0	

# Proof of the Main Lemma

**Proof.** Let  $B$  be an  $n$ -ary Boolean function and let  $S = \{A_1, \dots, A_n\}$ .

There are  $2^n$  possible values for  $\langle x_1, \dots, x_n \rangle \in \{0, 1\}^n$ .

Enumerate these as  $r_1, \dots, r_{2^n}$ . Let  $B(r_i)$  be the value of  $B$  at  $r_i$ , and assume that  $B$  is non-degenerate.

For each  $i$  s.t.  $B(r_i) = 1$ , form the conjunction  $\kappa_i = \lambda_1 \wedge \dots \wedge \lambda_n$ , where, for each  $j$  s.t.  $1 \leq j \leq n$ ,

$$\lambda_j = \begin{cases} A_j & \text{if } x_j = 1 \\ \neg A_j & \text{otherwise.} \end{cases}$$

Finally, let  $\alpha$  be the disjunction over the  $\kappa_i$ 's for which  $B(r_i) = 1$ . Note that  $\alpha$  is in CDNF.

## Proof (cont.)

We need to verify that  $B_\alpha^n(r_i) = B(r_i)$  for each  $i$  s.t.  $0 \leq i \leq 2^n$ .

Let  $v_i : \mathcal{S} \rightarrow \{0, 1\}$  be such that  $v_i(A_j) = l_j^n(r_i)$  for all  $j$  s.t.  $1 \leq j \leq n$ .  
Then

$$\begin{aligned} B(r_i) = 1 & \text{ iff } \bar{v}_i(\lambda_j) = 1 \text{ for each } 1 \leq j \leq n \\ & \text{ iff } \bar{v}_i(\kappa_i) = 1 \\ & \text{ iff } \bar{v}_i(\alpha) = 1 \\ & \text{ iff } B_\alpha^n(v_i(A_1), \dots, v_i(A_n)) = 1 \\ & \text{ iff } B_\alpha^n(r_i) = 1. \end{aligned}$$

If  $B$  is degenerate, let  $\alpha$  be  $A_1 \wedge \neg A_1$ . Although not in CDNF,  $\alpha$  is in DNF. ■

# A Corollary

**Corollary.** Every satisfiable wff has a truth-functional equivalent in CDNF. Every unsatisfiable wff has a truth-functional equivalent in DNF.

**Proof.** Let  $\varphi$  be a satisfiable wff and  $A_n$  the largest sentence letter occurring in  $\varphi$ . Then  $\varphi$  realizes  $B_\varphi^n$ , which is non-degenerate. Hence  $B_\varphi^n$  is realized by some sentence  $\alpha$  in CDNF. Thus  $B_\varphi^n = B_\alpha^n$ . By clause (2) of the Characterization Theorem, it follows that  $\varphi \models \alpha$ .

If  $\varphi$  is unsatisfiable, then  $\varphi \models (A_n \wedge \neg A_n)$ , and the latter wff is in DNF. ■

# Boolean Complete Sets of Connectives

**Def.** A set of connectives is **Boolean complete** (or *truth-functionally complete*) iff any Boolean function can be realized by a wff in which at most only those connectives occur.

**Corollary** The set  $\{\neg, \wedge, \vee\}$  is Boolean complete.

**Proof.** Any  $n$ -ary Boolean function ( $0 < n$ ) can be realized by a wff in CDNF or in DNF, and these these are the only connectives that can occur in these syntactic forms. ■

**Corollary.** Each of the following sets of connectives is Boolean complete:  $\{\neg, \wedge\}$ ,  $\{\neg, \vee\}$ ,  $\{\neg, \rightarrow\}$ ,  $\{\downarrow\}$  and  $\{\mid\}$ .

**Proof.** Exercise.

## The Case $n = 0$

Earlier we said that for each  $n \in \mathbb{N}$  the number of distinct  $n$ -adic truth-functional connectives is  $2^{(2^n)}$ . This entails that there are two 0-ary truth functional connectives. What are they?

Since with each  $n$ -ary connective  $*$  there is an associated  $n$ -ary expression building operation  $\mathcal{E}_* : U^n \rightarrow U$ , we need to get clear on the status of 0-ary operations in general.

**Def.** A 0-ary operation on a set  $A$  is just a member of  $A$ .

Thus there are two 0-ary Boolean functions, viz.,  $\top$  and  $\perp$ .

The two 0-ary truth-functional connectives are  $\top$  and  $\perp$ , commonly called 'top' and 'bottom', respectively.

Thus,  $\mathcal{E}_{\top} = \top$  and  $\mathcal{E}_{\perp} = \perp$ . A set  $\Sigma$  of expressions is closed under  $\mathcal{E}_{\top}$  (resp.,  $\mathcal{E}_{\perp}$ ) just in case  $\top \in \Sigma$  (resp.,  $\perp \in \Sigma$ ).

Semantically, we add to the recursion clauses: (i)  $\bar{v}(\top) = 1$ , and (ii)  $\bar{v}(\perp) = 0$ .