INTRODUCTION

- Want to construct model which is like the original object in *essential* ways and can be unlike the original object in *irrelevant* ways.
- deductive thought is inferring conclusions from assumptions, whatever they are
- Symbolic logic is a mathematical model of deductive thought
- Want to know (a) how to prove sentences following logically from others, (b) gaps in provability, and (c) computability.
- Will see that the *first-order* logic model is well suited for mathematics i.e. common (axiomatic) math can be translated into first-order logic.

0. USEFUL FACTS ABOUT SETS

0.1. SETS.

Definition. abbreviations

- $\dashv$ means end of proof
- $\_\Rightarrow\_$ means if _ then _
- $\Leftarrow$ means converse implication
- $\Leftrightarrow$ means iff
- $\therefore$ means therefore
- / means denial e.g. $\neq$

Definition.

- A set is a collection of objects called members or elements; determined just by its members
- $t \in A$ means t is an element of set A
- $x = y$ means they are the same object
- $A; t$ means $A \cup \{t\}$
- $\emptyset$ means empty set
- $\mathbb{N} = \{0, 1, 2, \cdots\}$
- $\mathbb{Z} = \{\cdots, -1, 0, 1, \cdots\}$
- $\{x | \_x\_\}$ means the set of all x s.t. $\_x\_$
- $A \subseteq B$ means $x \in A \Rightarrow x \in B$
- $\mathcal{P}A = \{x | x \subseteq A\}$ ("power set")

Example.

- $A = B \Leftrightarrow (\forall t, t \in A \Leftrightarrow t \in B)$
  note: the converse is called the principle of extensionality.
- $t \in A \Leftrightarrow A; t = A$
- $\forall$ finite number of elements $x_1, \cdots, x_n$, $\exists$ set $\{x_1, ..., x_n\}$
  e.g. singleton set $\{x\}$
- $\{x, y\} = \{y, x\}$ i.e. order doesnt matter
- $\emptyset$ is a subset of every set
  Pf: vacuously true since no elements to check
- $\mathcal{P}\emptyset = \{\emptyset\}$
  $\mathcal{P}\{\emptyset\} = \{\emptyset, \{\emptyset\}\}$

0.2. UNIONS AND INTERSECTIONS.

Definition.

- $A \cup B = \{x | x \in A \text{ or } x \in B \text{ or both}\}$
- $A \cap B = \{x | x \in A \text{ and } x \in B\}$
- A, B disjoint means $A \cap B = \emptyset$
- A disjoint collection of sets means pairwise disjoint
- Let A be a collection of sets
  $\cup A = \{x | x \in \text{some member of } A\}$
  $\cap A = \{x | x \in \text{all members of } A\}$
  $\cup_n A_n = \{x \in A_n \quad \forall n \in \mathbb{N}\}$ (?)

Example.

- Let $A = \{\{0, 1, 5\}, \{1, 6\}, \{1, 5\}\}$
  then $\cup A = \{0, 1, 5, 6\}$, $\cap A = \{1\}$
- $A \cup B = \cup\{A, B\}$
  $\cup\mathcal{P}A = A$

0.3. ORDERED PAIRS.

Definition.

- ordered pair $\langle x, y\rangle$ must satisfy property $\langle x, y\rangle = \langle u, v\rangle \Leftrightarrow x = u, y = v$
  standard definition: $\langle x, y\rangle = \{x, \{x, y\}\}$
- ordered triple defined $\langle x, y, z\rangle = \langle\langle x, y\rangle, z\rangle$
- n-tuple defined recursively by $\langle x_1, \cdots, x_n\rangle = \langle\langle x_1, \cdots, x_{n-1}\rangle, x_n\rangle$
  also, $\langle x\rangle = x$
- finite sequence ("string") of members of A is $S = \langle x_1, \cdots, x_n\rangle$ where $x_i \in A$ and $n \in \mathbb{N}$
  (can also define as finite function)
- segment of a finite sequence $S = \langle x_1, \cdots, x_n\rangle$ is a finite sequence $\langle x_k, x_{k+1}, \cdots, x_{m-1}, x_m\rangle$ where $1 \le k < m \le n$
- initial segment means $k = 1$
- proper segment means $1 < k < m < n$

Claim.

- $\langle x_1, \cdots, x_n\rangle = \langle y_1, \cdots, y_n\rangle \Rightarrow x_i = y_i$ *for* $1 \le i \le n$
  Pf: induction on n, basic properties of ordered pairs
- $\langle x_1, \cdots, x_m\rangle = \langle y_1, \cdots, y_n\rangle \not\Rightarrow m = n$
  Pf: every ordered triple is also an ordered pair.
  Note: some x_i is a finite sequence of y_i's $\Rightarrow m \neq n$

Lemma. *0A*
$\langle x_1, \ldots, x_m\rangle = \langle y_1, \ldots, y_{m+k}\rangle$
$\Rightarrow x_1 = \langle y_1, \ldots, y_{k+1}\rangle$

Proof. induction on m
$m = 1$ then conclusion is immediate
if $\langle x_1, \ldots, x_m, x_{m+1}\rangle = \langle y_1, \ldots, y_{m+k}, y_{m+k+1}\rangle$
then $\langle x_1, \ldots, x_m\rangle = \langle y_1, \ldots, y_{m+k}\rangle$
Apply inductive hypothesis

Example. Let set A, no member of A is a finite sequence of its other members.
if $\langle x_1, \ldots, x_m\rangle = \langle y_1, \ldots, y_n\rangle$ where $x_i, y_j \in A$
then $m = n$ and $x_i = y_i$
Pf: Lemma 0A

Definition.

- cartesian product of sets $A \times B = \{\langle x, y\rangle | x \in A, y \in B\}$
- A^n is the set of all n-tuples of members of A
  e.g. $A^3 = (A \times A) \times A$

0.4. RELATIONS.

Definition.

- relation R is a set of ordered pairs
  e.g. ordering relation on $\{0, 1, 2\}$ is $\{\langle 0, 1\rangle, \langle 0, 2\rangle, \langle 1, 2\rangle\}$
- The domain of relation R $\operatorname{dom} R = \{x | \langle x, y\rangle \in R\}$
  The range of relation R $\operatorname{ran} R = \{y | \langle x, y\rangle \in R\}$
  The field of relation R $\operatorname{fld} R = \operatorname{dom} R \cup \operatorname{ran} R$
- An n-ary relation on set A is a subset of A^n
  a 1-ary ("unary") relation is just a subset of A
  a 2-ary ("binary") relation is a subset of $A \times A$
  e.g. equality relation $\{\langle x, x\rangle | x \in A\}$
- Let n-ary relation R on A; $B \subseteq A$
  The restriction of R to B is $R \cap B^n$
  e.g. above ordering relation on $\{0, 1, 2\}$ is restriction on $\mathbb{N}$

Definition. Enderton put these in section (0.5)

- reflexive relation R on A means $\langle x, x\rangle \in R \forall x \in A$
- symmetric relation R means $\langle x, y\rangle \in R \Rightarrow \langle y, x\rangle \in R$
- transitive relation R means $\langle x, y\rangle \in R, \langle y, z\rangle \in R \Rightarrow \langle x, z\rangle \in R$
- equivalence relation R on A means R is a binary relation which is reflexive on A, symmetric, transitive
- relation R satisfies trichotomy on A means $\forall x, y \in A$, exactly one holds:
  $(1)\langle x, y\rangle \in R$, $(2) x = y$, $(3)\langle y, x\rangle \in R$
- ordering relation R on A means R is transitive and satisfies trichotomy on A
- Let equivalence relation R on A
  The equivalence class of $x \in A$ wrt R is $[x] = \{y | \langle x, y\rangle \in R\}$
  Note: equivalence classes partition A i.e. equivalence classes are subsets of A and each $x \in A$ belongs to exactly one equivalence class; $[x] = [y]$ iff $\langle x, y\rangle \in R$

0.5. FUNCTIONS.

Definition.

- A function is a relation F with the single-valued property $(\forall x \in \operatorname{dom} R, \exists! y (= \text{"}f(x)\text{"}) \text{ s.t. } \langle x, y\rangle \in F)$
- composition $f \circ g$ of functions is the function whose value at x is $f(g(x))$
- f maps A into B, $f : A \to B$, means f is a function, $\operatorname{dom} F = A$, $\operatorname{ran} F \subseteq B$
  f maps A onto B, $f : A \to B$, means f is a function, $\operatorname{dom} F = A$, $\operatorname{ran} F = B$
- f is one-to-one ("1-1") means $\forall y \in \operatorname{ran} F, \exists! x$ s.t. $\langle x, y\rangle \in F$
- $f(\langle x, y\rangle)$ (="$f(x, y)$") means $\langle x, y\rangle \in \operatorname{dom} f$
  $f(x_1, x_2, \cdots, x_n) = f(\langle x_1, x_2, \cdots, x_n\rangle)$ similarly
- n-ary operation on A is a function mapping A^n into A
  e.g. a binary operation on $\mathbb{N}$ is addition
  e.g. a unary operation on $\mathbb{N}$ is successor $S(n) = n + 1$
  e.g. a unary operation on A is Identity function $Id = \{\langle x, x\rangle | x \in A\}$; $Id(x) = x \forall x \in A$
- Let f be an n-ary operation on A.
  The restriction of f to $B \subseteq A$ is the function g with domain B^n and agrees with f at each point of B^n
  thus $g = f \cap (B^n \times A)$
- this g is an n-ary relation on B iff B is closed under f i.e. $f(b_1, \cdots, b_n) \in B$
  then $g = f \cap B^{n+1}$
  e.g. the addition operation on $\mathbb{N}$ is addition on $\mathbb{R}$ restricted to $\mathbb{N}$, contains $\langle\langle 2, 3\rangle, 5\rangle$

0.6. SIZES OF SETS.

Definition.

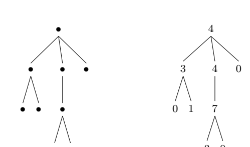
- finite set A means $\exists f : A \to \{0, 1, \cdots, n-1\}$ 1-1 and onto
- countable set A means $\exists f : A \to \mathbb{N}$ 1-1
  e.g. finite $\Rightarrow$ countable

Claim. *If A infinite countable set, $f : A \to \mathbb{N}$ 1-1 then $\exists f' : A \to \mathbb{N}$ 1-1, onto*
Pf: $A = \{a_0, a_1, a_2, \ldots\}$, define $f'(a_n)$ to be the $(n+1)^{th}$ least member of $\operatorname{ran} f$

Theorem. *0B*
if A countable set
then the set S of all finite sequences of members of A is countable

Proof. let $S = \cup_{n \in \mathbb{N}} A^{n+1}$
A countable $\Rightarrow \exists f : A \to \mathbb{N}$ 1-1
define map $S \mapsto \mathbb{N}$ 1-1 by assigning $\langle a_0, a_1, \cdots, a_m\rangle$ the number $2^{f(a_0)+1} 3^{f(a_1)+1} \cdots p_m^{f(a_m)+1}$ where p_m is the $(m+1)^{\text{th}}$ prime
See book for fixing some details.

Remark. informally, a tree is a picture of a finite partial ordering R s.t. if $\langle a, b\rangle \in R$ then a is lower than b and connected by a line



where the right tree labeled by $f : \{\text{vertices}\} \to \mathbb{N}$

Remark. will need axiom of choice when generalizing past countable languages.
Of the many equivalent statements, Zorn's lemma will be especially useful.

Lemma. *(Zorn's) Let set A s.t. for any chain (?) $C \subseteq A$ we have $\cup C \subseteq A$*
then $\exists m \in A$ which is maximal *(not a subset of any other elements of A)*

0.7. CARDINAL NUMBERS.

Remark. Want to compare sizes of sets

Definition. Sets A, B are equinumerous, $A \sim B$, means $\exists f : A \to B$ 1-1 and onto i.e. $\exists$ equivalence relation
e.g. finite sets, just compare number of elements
e.g. $\mathbb{N} \sim \mathbb{Z}$

Remark. sizes of finite sets are natural numbers
sizes of infinite sets are cardinal numbers

Definition.

- cardinality of A, card A, is an object assigned to A s.t. card $A =$ card B iff $A \sim B$
  Note: common def satisfies requirement by card $A =$ least ordinal (wont define) equinumerous with A, but this requires axiom of choice
- cardinal number ("cardinal") is something that is card A for some A
- A is dominated by B, $A \preceq B$, means A is equinumerous with a subset of B i.e. $\exists f : A \to B$ 1-1, into
  in terms of cardinals, $A \preceq B$ means card $A \leq$ card B
  note: dominance is reflexive, transitive
  e.g. $A \preceq \mathbb{N}$ iff A countable

Theorem. *(Schroder-Bernstein)*
a) for any sets A, B, $[A \preceq B, B \preceq A \Rightarrow A = B]$
b) for any cardinal numbers λ, κ, $[\lambda \leq \kappa, \kappa \leq \lambda \Rightarrow \lambda = \kappa]$

Theorem. *0C (equivalent to axiom of choice)*
a)for any sets A, B, either $A \preceq B$ or $B \preceq A$
b)for any cardinal numbers κ, λ, either $\lambda \leq \kappa$ or $\kappa \leq \lambda$

Remark. So any nonempty set of cardinal numbers contains a smallest one.
The cardinal numbers in order of size are:
$0, 1, 2, \cdots, \aleph_0 (= \mathbb{N}), \aleph_1, \cdots$
where $\aleph_0$ is the smallest infinity size

Example. card $\mathbb{N} <$ card $\mathbb{R} = 2^{\aleph_0}$

Definition. addition and multiplication generalized to nonfinite cardinals
Let A, B disjoint, $\kappa =$ card A, $\lambda =$ card B
Then $\kappa + \lambda = \text{card}(A \cup B)$ and $\kappa \cdot \lambda = \text{card}(A \times B)$
note: these are well-defined since only depends on κ, λ and not on choice of A, B.

Theorem. *(cardinal arithmetic)*
Normal for finite cardinals. For infinite cardinals, if $\kappa \leq \lambda$ then $\kappa + \lambda = \lambda$ and $\kappa \cdot \lambda = \lambda$ (for $\kappa \neq 0$)
e.g. If κ is infinite cardinal, then $\kappa \cdot \aleph_0 = \kappa$

Theorem. *0D*
Let infinite set A.
Then set $\cup_n A^{n+1}$ of all finite sequences of elements of A has cardinality equal to card A

Proof. For countable A, see thm 0B
Each A^{n+1} has cardinality equal to card A by applying cardinal arithmetic theorem n times
So we have union of $\aleph_0$ sets of this size
So $\aleph_0 \cdot$ card $A =$ card A

Example.

- The set of algebraic numbers (roots of polynomials in one variable) has cardinality $\aleph_0$
  Pf: each polynomial (in one variable) over the integers is identified with the sequence of coefs
  thm implies there are $\aleph_0$ such polynomials
  each polynomial has a finite number of roots
  even if each polynomial had $\aleph_0$ roots, $\aleph_0 \cdot \aleph_0 = \aleph_0$
- card $\mathbb{R} = 2^{\aleph_0}$
  So card{transcendental numbers} $= 2^{\aleph_0}$

1. SENTENTIAL LOGIC

INFORMAL REMARKS ON FORMAL LANGUAGES

Definition.

- natural language is ambiguous, very expressive
- formal language is precise, less expressive

Example. can translate between formal and (declarative parts and compounds of) natural language
Let declarative $K=$ Potassium was observed and $C=$ Chlorine was observed
Given truth and falsity of declarative parts, can compute truth or falsity of compound

K	C	$(\neg(C \vee K))$	$((\neg C) \wedge (\neg K))$	$(K \to (\neg C))$	$(C \wedge K)$
F	F	T	T		
F	T	F	F		
T	F	F	F		
T	T	F	F		

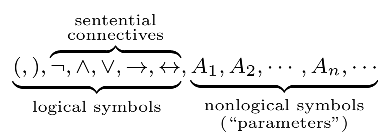
Note: $(\neg(C \vee K))$ and $((\neg C) \wedge (\neg K))$ are the same

Remark. To describe formal language, need:
1) alphabet (ie set of symbols) (e.g. sentential logic: $(,), \neg, \wedge, \vee, \to, \leftrightarrow, A_1, A_2, \ldots$)
2) Grammar rules (ie whether a finite sequence of symbols is a well formed formula wff)
3) (optional) translation between symbols $A_1, A_2, \cdots$ and declarative english

Example. binary language wff: 011010001
assembly language wff: STEP#ADDIMAX, A
C++ language wff: while(*s++);

1.1. THE LANGUAGE OF SENTENTIAL LOGIC.

Definition. The symbols of sentential logic is an infinite sequence of objects, each atomic (not finite combination of other symbols)



symbols	verbose name	English
(	left parentheses	punctuation
)	right parentheses	
$\neg$	negation	not
$\wedge$	conjunction	and
$\vee$	disjunction	or (inclusive)
$\to$	conditional	if_then_
$\leftrightarrow$	biconditional	iff
A_1	first sentence	
A_2	second sentence	
$\vdots$		
A_n	nth sentence	
$\vdots$		

Remark.

- logical symbols never change roles; whereas parameters are open to interpretation
- could use one sentence symbol and a prime ie $A, A', A'', \cdots$ and have 9 total symbols could also have arbitrary set of sentence symbols, even uncountably many, but §1.7 requires countably many
- "propositional logic" is similar, calling each A_n the "n^{th} proposition symbol"
  then "sentence" is a particular utterance and "proposition" is that which a "sentence" asserts
- the ontological status of symbols remain neutral
  e.g. conditional symbol $\to$ may not have any geometric property of an arrow
  e.g. symbols themselves can be sets, numbers, marbles, or objects from a universe of linguistic objects
  sentence symbols can even be formulas from another language (ch. 2)

- symbols are atomic
  this allows any finite sequence of symbols to be uniquely decomposable
  so if $\langle a_1, \cdots, a_m \rangle = \langle b_1, \cdots, b_n \rangle$ where each a_i, b_j are symbols
  then $m = n$ and $a_i = b_i$
  Pf: Lemma 0A

Definition. expression is a finite sequence of symbols, specified by concatenating symbols (e.g. $\langle (, \neg, A_1,) \rangle$ is $(\neg A_1)$) or sequences of symbols (e.g. Let $\alpha = (\neg A_1), \beta = A_2$ then $(\alpha \to \beta)$ is $((\neg A_1) \to A_2)$).

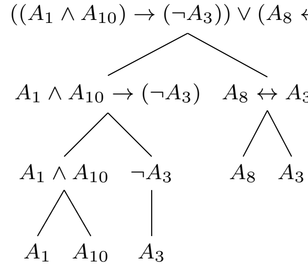
Example. translations between English
(see book for many examples)
caution: English sentences can have different interpretations in different contexts

Remark. Some expressions are nonsense. Need definition for wff.

Definition.

- formula-building operations on expressions α, β are
$$\varepsilon_\neg(\alpha) = (\neg\alpha)$$
$$\varepsilon_\wedge(\alpha, \beta) = (\alpha \wedge \beta)$$
$$\varepsilon_\vee(\alpha, \beta) = (\alpha \vee \beta)$$
$$\varepsilon_\to(\alpha, \beta) = (\alpha \to \beta)$$
$$\varepsilon_\leftrightarrow(\alpha, \beta) = (\alpha \leftrightarrow \beta)$$
- grammatically correct expression ("formula","wff") satisfies
  a) every sentence symbol is a wff
  b) if α and β are wff, then so are expressions built by applying formula-building operations e.g. $(\neg\alpha), (\alpha \wedge \beta)$, etc.
  c) otherwise, not wff (ie "compulsion")
  note: empty sequence is not a wff, so smallest wff is a sentence symbol

Example. Ancestral tree
$((A_1 \wedge A_{10}) \to (\neg A_3)) \vee (A_8 \leftrightarrow A_3)$



Remark.

- We have a mathematical construction of taking building blocks (here, sentence symbols) and "closing" under operations (here, formula building operations)
  see §1.4 for this sort of construction in more general setting
- want to elaborate on building

Definition.

- construction sequence is a finite sequence $\langle \varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n \rangle$ of expressions s.t. $\forall i \leq n$, we have at least one of:
  (a) ε_i is a sentence symbol
  (b) $\varepsilon_i = \varepsilon_\neg(\varepsilon_j)$ for some $j < i$, or
  (c) $\varepsilon_i = \varepsilon_\square(\varepsilon_j, \varepsilon_k)$ for some $j, k < i$ where $\square$ is a binary connective $\wedge, \vee, \to, \leftrightarrow$
  intuition: ε_i is the expression at "stage i" in the building process
  intuition: obtain a construction sequence by squashing its ancestral tree into a linear ordering?
- Characterize wff as an expression α s.t. some construction sequence ends in α

Remark. this construction yields an induction principle

Theorem. *(induction principle)*
Say set S is closed under a "two-place" function f means $[x, y \in S \Rightarrow f(x, y) \in S]$ and similarly for "one-place" function
If S is a set of $wffs$ containing all the sentence symbols and closed under all five formula building operations
then S is the set of all $wffs$

Proof. given arbitrary wff α, make ancestral tree, starting from bottom, each node is expression $\in S$, build up to α, so $\alpha \in S$
(can prove similarly using construction sequence and strong induction)

Claim. *any wff has a balanced number of left, right parentheses*

Proof. start with sentence symbol, apply formula building operations which add parentheses only in matched pairs, induction principle

Remark. formula building operations only build up, not down
ie $\varepsilon_{\Box}(\alpha, \beta)$ always includes sequences α, β as entire segments
This will help, given a wff φ, how it was built up

1.2. TRUTH ASSIGNMENTS.

Remark. what does it mean for one wff to "follow logically" from other $wffs$?
e.g. $(A_1 \wedge A_2)$ true means A_1 true

Definition.

- truth values are set of two distinct points, say $\{F, T\}$
- truth assignment for a set S of sentence symbols is a function $v : S \to \{F, T\}$
- (two-valued logic is logic with truth values $\{F, T\}$ can also have three-valued, n-valued, $\aleph_0$-valued, $[0, 1]$-valued, but we focus on two-valued since it is the most significant)
- $\bar{S}$= set of all wff that can be built up from S by the five formula building operations
- extension to v is $\bar{v} : \bar{S} \to \{F, T\}$ s.t.
  0)$\forall A \in S$, $\bar{v}(A) = v(A)$ i.e. "extension"
  and $\forall \alpha, \beta \in \bar{S}$
  1)$\bar{v}((\neg\alpha)) = \begin{cases} T & \text{if } \bar{v}(\alpha) = F \\ F & \text{ow} \end{cases}$
  2)$\bar{v}((\alpha \wedge \beta)) = \begin{cases} T & \text{if } \bar{v}(\alpha) = T and \bar{v}(\beta) = T \\ F & \text{ow} \end{cases}$
  3)$\bar{v}((\alpha \vee \beta)) = \begin{cases} T & \text{if } \bar{v}(\alpha) = T \text{ or } \bar{v}(\alpha) = T \text{ or both} \\ F & \text{ow} \end{cases}$
  4)$\bar{v}((\alpha \to \beta)) = \begin{cases} F & \text{if } \bar{v}(\alpha) = F \text{ and } \bar{v}(\beta) = T \\ T & \text{ow} \end{cases}$
  5)$\bar{v}((\alpha \leftrightarrow \beta)) = \begin{cases} T & \text{if } \bar{v}(\alpha) = \bar{v}(\beta) \\ F & \text{ow} \end{cases}$

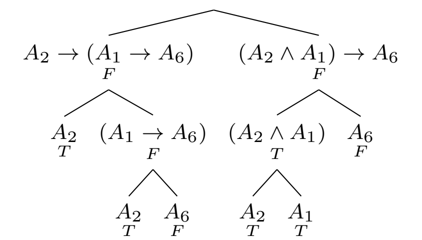
Corresponding truth table

α	β	$(\neg\alpha)$	$(\alpha \wedge \beta)$	$(\alpha \vee \beta)$	$(\alpha \to \beta)$	$(\alpha \leftrightarrow \beta)$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

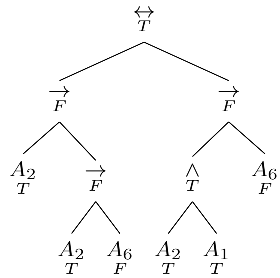
Remark.

- $(\alpha \to \beta)$ is "vacuously" true when $\alpha = F$ intuitively, its a promise for α true, and α false never breaks the promise
- ignore nuances of everyday speech

Example. Let $v(A_1) = T$, $v(A_2) = T$, $v(A_6) = F$
Compute $\bar{v}(\alpha)$ for $\alpha = (A_2 \to (A_1 \to A_6)) \leftrightarrow ((A_2 \wedge A_1) \to A_6))$
Soln: method1: build ancestral tree and work from bottom computing $\bar{v}$ at each vertex.



method 2: same but less writing



method 3: same but even less writing

$$(\underset{T}{A_2} \underset{F}{\to} (\underset{T}{A_1} \underset{F}{\to} \underset{F}{A_6})) \underset{T}{\leftrightarrow} ((\underset{T}{A_2} \underset{T}{\wedge} \underset{T}{A_1}) \underset{F}{\to} \underset{F}{A_6}))$$

Theorem. *12A (preview of $\exists!\bar{v}$)*
Let set S
$\forall v : S \to \{F, T\}$, $\exists!\bar{v} : \bar{S} \to \{F, T\}$ meeting conditions 0-5.

Proof. Will emerge in §1.3-4
existence intuitive from eg above. uniqueness more difficult to prove

Definition.

- truth assignment $\bar{v}$ satisfies wff φ means $\bar{v}(\varphi) = T$
- Let Σ be a set of wffs, τ be a wff
  Σ tautologically implies τ, $\Sigma \models \tau$, means every truth assignment for the sentence symbols in Σ and τ that satisfies every member of Σ also satisfies τ
  intuition: conclusion τ satisfied whenever hypotheses Σ satisfied
  notation: for singleton $\Sigma = \{\sigma\}$, write $\sigma \models \tau$
- τ is a tautology, $\models \tau$, means wff that is always satisfied
  intuition $\emptyset \models \tau$ since $\emptyset$ is vacuously satisfied
- tautologically equivalent, $\sigma \models\!\!\!\!\!\dashv \tau$, means both $\sigma \models \tau$ and $\tau \models \sigma$

Example.

- above example satisfied for $v(A_1) = T$, $v(A_2) = T$, $v(A_6) = F$
  it is also satisfied for the other seven truth assignments for $\{A_1, A_2, A_6\}$, so tautology
- $\{A, (\neg A)\} \models \tau$ vacuously true $\forall \tau$
  since $\{A, (\neg A)\}$ never satisfied
- $\{A, (A \to B)\} \models B$ is tautology since both satisfied only when $v(A) = v(B) = T$, and then $v(B) = T$
- from beginning of chapter, $(\neg(C \vee K)) \models\!\!\!\!\!\dashv ((\neg C) \vee (\neg K))$

Theorem. *(preview of compactness thm)*
Let infinite set Σ of $wffs$
if every finite subset of Σ satisfiable
then Σ itself is satisfiable

Proof. see §1.7
This then asserts compactness of a certain topological space, and can prove using a theorem (Tychonoff's thm on product spaces) from topology

1.2.1. *TRUTH TABLES.*

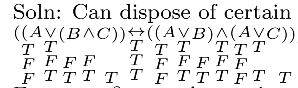
Remark. want systematic procedure to check whether $\{\sigma_1, \cdots, \sigma_k\} \models \tau$, including case $k = 0$.
General method make table for the 2^k truth assignments.

Example.

- Show $(\neg(A \wedge B)) \models ((\neg A) \vee (\neg B))$

	A	B	$(\neg(A \wedge B))$	$((\neg A) \vee (\neg B))$
	T	T	FTTT	FTFFT
Soln:	T	F	TTFF	FTTTF
	F	T	TFFT	TFTTF
	F	F	TFFF	TFTTF

- Show $(\neg(A \wedge B)) \not\models ((\neg A) \wedge (\neg B))$
  Soln: just need one truth assignment satisfying LHS but not RHS
- Show $\models ((A \vee (B \wedge C)) \leftrightarrow ((A \vee B) \wedge (A \vee C)))$
  Soln: Can dispose of certain cases



- For $\alpha \to \beta$, can also notice strong antecedent α so weak conditional? see book.

Remark. General truth table method requires 2^n rows (exponential time).
Dont know whether there is polynomial time method, this is Pvs.NP related.

1.2.2. *A SELECTED LIST OF TAUTOLOGIES.*

1) Associative and commutative laws for $\wedge, \vee, \leftrightarrow$ (?)
2) Distributive laws
   $((A \wedge (B \vee C)) \leftrightarrow ((A \wedge B) \vee (A \wedge C)))$
   $((A \vee (B \wedge C)) \leftrightarrow ((A \vee B) \wedge (A \vee C)))$
3) Negation
   $((\neg(\neg A)) \leftrightarrow A)$
   $((\neg(A \to B)) \leftrightarrow (A \wedge (\neg B)))$
   $((\neg(A \leftrightarrow B)) \leftrightarrow ((A \wedge (\neg B)) \vee ((\neg A) \wedge B)))$
   DeMorgan's Laws
   $((\neg(A \wedge B)) \leftrightarrow ((\neg A) \vee (\neg B)))$
   $((\neg(A \vee B)) \leftrightarrow ((\neg A) \wedge (\neg B)))$
4) Other
   Excluded middle: $(A \vee (\neg A))$
   Contradiction: $(\neg(A \wedge (\neg A)))$
   Contraposition: $((A \to B) \leftrightarrow ((\neg B) \to (\neg A)))$
   Exportation: $(((A \wedge B) \to C) \leftrightarrow (A \to (B \to C)))$

1.3. A PARSING ALGORITHM.

Remark.

- This subsection: will prove that we have used enough parentheses to remove ambiguity
  $\Rightarrow$ each wff is formed in a unique way e.g. $\bar{v}(A_1 \vee A_2 \wedge A_3)$ depends on where you put parentheses for $v(A_1) = T, v(A_2) = F$
  $\Rightarrow \exists\bar{v}$
- Instead of formulas,
  could just use ordered pairs (eg $\langle \neg, \alpha \rangle$) and triples (eg $\langle \alpha, \wedge, \beta \rangle$) and their concatenations,
  then it is immediate that formulas have unique decompositions
  but we want to use formulas

Lemma. *13A*
Every wff has the same number of left and right parentheses

Proof. end of §1.1

Lemma. *13B*
Any proper initial segment of a wff contains an excess of left parentheses
thus no proper initial segment of a wff can itself be a wff

Proof. Let S= set of wff s.t. any initial segment contains an excess of left parentheses
Any sentence symbol is in S vacuously
list all proper initial segments of $(\alpha \wedge \beta) \in S$.

S is closed under $\varepsilon_\wedge$ (see book)
similar for $\varepsilon_\neg, \varepsilon_\vee, \varepsilon_\rightarrow, \varepsilon_\leftrightarrow$

1.3.1. *A PARSING ALGORITHM.*

Remark.

- Want a procedure that, given an expression, will determine whether it is well formed and if so, (?)
- will construct an ancestral tree using formula-building operations
- will see that this tree is unique for given wff (which $\Rightarrow$ we have enough parentheses to remove ambiguity)

Algorithm. Input: an expression
Output: "not a wff" or a tree

1) If all bottom vertices are sentence symbols, then done
else take a bottom vertex
2) The first symbol must be (
if second symbol is $\neg$, then jump to step 4
3) scan expression from left to right to find "(α" where α is nonempty expression with balanced parentheses
next symbol must be $\wedge, \vee, \rightarrow$, or $\leftrightarrow$, i.e. the principle connective
remainder must be "β)"
extend tree by forming two new vertices below present one, with "left child" α and "right child" β
return to step 1
4) first two symbols must be "($\neg$"
remainder must be "β)"
extend tree with new vertex β
return to step 1

Claim.

1) algorithm halts after finite number of steps
2) algorithms choices could not have been made differently
3) if expressions didnt obey "must"
then not wff
4) if all "must"'s are obeyed
then expression is wff

Proof.

1) any vertex contains shorter expressions than above
so tree depth bounded by expression length
2) in step 3, could not use more or less than α, otherwise violates lemmas 13A,B
so choice of principle connective is inevitable
so this algorithm constructs the only possible tree
3) rejection is correct since the only possible attempt to make the tree failed
4) working up inductively, each vertex is a wff, including top vertex

Theorem. *(unique readability)*
$\exists!$ tree for each wff

Proof. can construct tree using the algorithm, and unique since claim 2

Claim. *given v, $\exists!\bar{v}$*

Proof. given v and any wff φ,
can build its unique tree with algorithm
then define $\bar{v}$ at each vertex starting from bottom s.t. satisfies conditions 0-5 above
see exercise 1.2.14 for uniqueness of $\bar{v}$

Remark. $\bar{v}$ is tricky because it is defined recursively ie nested in itself
more on general recursion in next section

1.3.2. *POLISH NOTATION.*

Remark.

- can avoid parentheses without ambiguity
define set of <u>p-wff's</u> generated by operations $D_\neg(\alpha) = \neg\alpha, D_\square(\alpha,\beta) = \square\alpha\beta$ where $\square = \wedge, \vee, \rightarrow$, or $\leftrightarrow$
e.g. $\rightarrow \wedge AD \vee \neg B \leftrightarrow CB$ (see §2.3 for unique readability thm for such an expression)
- computer compilers usually begin by converting to polish notation

1.3.3. *OMITTING PARENTHESES.*

Remark. For neatness, will henceforth omit some parentheses. Conventions:

1) can omit outermost parentheses
2) negation symbol applies to as little as possible
3) given 2, the conjunction and disjunction symbols apply to as little as possible
e.g. $A \wedge B \rightarrow \neg C \vee D$ is $((A \wedge B_\rightarrow((\neg C) \vee D))$
4) when one connective symbol is repeated, grouping is to the right
e.g. $\alpha \wedge \beta \wedge \gamma$ is $\alpha \wedge (\beta \wedge \gamma)$

1.4. INDUCTION AND RECURSION.

Remark. This section can be postponed

1.4.1. *INDUCTION.*

Remark. Given some set, want to construct smallest subset closed under repeating same operation a finite number of times on some initial elements of the set
Will consider abstractly, but we are specifically interested in sentential stuff

Definition.

- some set U
For sentential, $U = \{\text{all expressions}\}$
- <u>initial elements</u> $B \subseteq U$
For sentential, $B = \{\text{the sentence symbols}\}$
- <u>operations</u> $\mathscr{F} = \{$<u>unary $g: U \rightarrow U$, binary $f: U \times U \rightarrow U$</u>$\}$ (need not be finite set, will generalize to more operations in ch 2, and exercise 1.4.3)
For sentential, $\mathscr{F} = \{\varepsilon_\neg, \varepsilon_\wedge, \varepsilon_\vee, \varepsilon_\rightarrow, \varepsilon_\leftrightarrow\}$
- <u>constructed set</u> C is the set <u>generated from B by</u> ("closed under") $\mathscr{F}$.
intuition: given bricks $B = \{a, b\}$ and mortar f, g, want $C = \{a, b, g(a), f(b,b), g(f(b,b)), \dots\}$ to contain the things we can build
eg sentential $C = \{\text{all } wff\}$
Two constructions:
  1) "from top down"
  Let $S \subseteq U$
  Say S is <u>closed</u> under f, g means $[x, y \in S \Rightarrow f(x,y), g(x) \in S]$
  Say S is <u>inductive</u> means $B \subseteq S$ and S is closed under f, g
  Define $C^* = \cap\{S | S \text{ is inductive subset of } U\}$
  Note: C^* is the smallest inductive set
  2) "from bottom up"
  define <u>construction sequence</u> to be a finite sequence $\langle x_1, \cdots, x_n\rangle$ of elements of U s.t. $\forall i \leq n$, at least one of: (a) $x_i \in B$, (b) $x_i = f(x_j, x_k)$ for some $j, k < i$, (c) $x_i = g(x_j)$ for some $j < i$
  Define $C_* = \{x | \text{some construction sequence ends with } x\}$
  in other words, C_* contains things reachable from B by applying f and g a finite number of times.
  in other words, let $C_n = \{x | \text{some construction sequence of length } n \text{ ends with } x\}$
  then $C_1 = B$ and $C_1 \subseteq C_2 \subseteq \cdots$
  Let $C_* = \cup_n C_n$
  e.g. $(g(f(a, f(b,b)))) \in C_5$, which can be written as a tree

Example.

1) $U = \mathbb{R}, B = \{0\}, \mathscr{F} = \{S(x) = x+1\}$
Then $C_* = \mathbb{N}$
2) $U = \mathbb{R}, B = \{0\}, \mathscr{F} = \{S(x) = x+1, P(x) = x-1\}$
then $C_* = \mathbb{Z}$
note: there is more than one way to obtain each $x \in C_*$
3) $U = \{\text{functions with domain,range } \subseteq \mathbb{R}\}$
$B = \{\text{identity function, all const functions}\}$
$\mathscr{F} = \{\text{add, multiply, divide, and root extract}\}$ which are operations on functions
then C_* is a class of algebraic functions
4) $U = \{\text{all expressions}\}$
$B = \{\text{the sentence symbols}\}$
$\mathscr{F} = \{\varepsilon_\neg, \varepsilon_\wedge, \varepsilon_\vee, \varepsilon_\rightarrow, \varepsilon_\leftrightarrow\}$
then $C_* = \{\text{all wffs}\}$

Claim. $C^* = C_* =$ *"C"*

Proof. ($C^* \subseteq C_*$): C_* is inductive since $B = C_1 \subseteq C_*$ and C_* closed under f since $[x, y \in C_* \Rightarrow f(x,y) \in C_*]$ and g since $[x \in C_* \Rightarrow g(x) \in C_*]$. And can append $f(x,y)$ or $g(x)$ to any construction sequence.
($C_* \subseteq C^*$): consider a point $x_n \in C_*$ and a construction sequence $\langle x_0, \dots, x_n\rangle$
induction: base: $x_0 \in B \subseteq C^*$. induction on i: $x_i \in C^*$, $i \leq n$ since C^* is closed under the functions.

Theorem. *(induction principle)*
Let C generated from B by $\mathscr{F}$
If $S \subseteq C$ is inductive (ie $B \subseteq S$ and S closed under the functions in $\mathscr{F}$)
then $S = C$

Proof. ($C \subseteq S$): S is inductive so $C = C^* \subseteq S$
($S \subseteq C$): hypothesis (inclusion?)

Example.

- α, β are proper segments of $\varepsilon_\wedge(\alpha, \beta)$
- C_{special} generated from $\{A_2, A_3, A_5\}$ by $\{\varepsilon_\neg, \varepsilon_\rightarrow\}$
claim: every wff either belongs to C_{special} or it is not *special* (expression with only $A_2, A_3, A_5, \neg, \rightarrow$)
pf: induction principle

1.4.2. *RECURSION.*

Remark.

- recall, U is some set, $B \subseteq U$ is initial set, $\mathscr{F} = \{f: U \times U \rightarrow U, g: U \rightarrow U\}$, C is set generated from B by $\mathcal{F}$
- want to define a function $\bar{h}$ on C recursively

Definition. $\bar{h}$ is defined <u>recursively</u> given
1) rules for computing <u>$\bar{h}(x)$</u> $\forall x \in B$
2a) rules for computing $\bar{h}(f(x,y))$ making use of $\bar{h}(x)$ and $\bar{h}(y)$
2b) rules for computing $\bar{h}(g(x))$ making use of $\bar{h}(x)$

Remark. Uniqueness obvious (?), but may not exist

Example.

- $\bar{v}$ in §1.2
- might not exist
Let $U = \mathbb{R}$, $B = \{0\}$, $f(x,y) = x \cdot y$, $g(x) = x+1$
then $C = \mathbb{N}$
Let $\bar{h}$ s.t. (1)$\bar{h}(0) = 0$, (2a)$\bar{h}(f(x,y)) = f(\bar{h}(x), \bar{h}(y))$, (2b)$\bar{h}(g(x)) = \bar{h}(x) + 2$
no such $\bar{h}$ can exist since ambiguity with $\bar{h}(1) = 2$ using $g(0) = 1$ and $\bar{h}(1) = 4$ using $f(g(0), g(0)) = 1$
- G= group generated from B by the group multiplication and inverse operator
then arbitrary map B into group H is not necessarily extendable to a homomorphism of G into H
but if G is a free group with set B of indep generators

then any such map is extendable to a homomorphism of the entire group

Definition. C is freely generated from B by f, g means it is generated and the restrictions f_C, g_C of f, g to C meet (1) f_C, g_C are 1-1 and (2) the range of f_C, the range of g_C, and B are piecewise disjoint

Theorem. *(recursion theorem)*
(short version: if C is freely generated then a function h on B always has an extension $\bar{h}$ on C)
Assume $C \subseteq U$is freely generated from B by $f : U \times U \to U$ and $g : U \to U$
Assume V is a set, and let functions $h : B \to V$, $F : V \times V \to V$, $G : V \to V$
then $\exists! \bar{h} : C \to V$ s.t. (i) $\forall x \in B, \bar{h}(x) = h(x)$ and (ii) $\forall x, y \in C$, $\bar{h}(f(x,y)) = F(\bar{h}(x), \bar{h}(y))$, $\bar{h}(g(x)) = G(\bar{h}(x))$

Proof. pages 41-44

Remark.

- intuition from algebra: any map $h : B$ into V can be extended to homomorphism $\bar{h} : C$ (with operators f, g) into V (with operators F, G)
- intuition chromatically: $\bar{h}$ paints each member of C some color
  1) h tells you how to color the initial elements B
  2) F tells you how to combine colors x, y to obtain color $f(x,y)$ i.e. F gives $\bar{h}(f(x,y))$ in terms of $\bar{h}(x), \bar{h}(y)$
  3) G tells you how to convert color x to color $g(x)$
  note: C is freely generated, so avoid conflict when F says green but G says red for some point unlucky enough to be equal both to $f(x,y)$ and $g(z)$ for some x, y, z
- (continue 1-4 above) 1) $C = \mathbb{N}$ is freely generated from $\{0\}$ by $S(x) = x + 1$
  since S is 1-1 and $0 \notin$ range of S
  so recursion thm $\Rightarrow \forall$ set $V, \forall a \in V, \forall F : V \to V$, $\exists! \bar{h} : \mathbb{N} \to V$ s.t. $\bar{h}(0) = a, \bar{h}(S(x)) = F(\bar{h}(x)) \forall x \in \mathbb{N}$
  e.g. $\exists! \bar{h} : \mathbb{N} \to \mathbb{N}$ s.t. $h(0) = 0$ and $\bar{h}(x) = 1 - \bar{h}(x)$; intuition: $\bar{h} = 0$ for x even and 1 for x odd
  2) $C = \mathbb{R}$ is not freely generated from $\{0\}$ by $S(x) = x + 1, P(x) = x - 1$
  3) freeness fails
  4) wffs are freely generated from $\{$sentence symbols$\}$ by $\{\varepsilon_\neg, \varepsilon_\wedge, \varepsilon_\vee, \varepsilon_\to, \varepsilon_\leftrightarrow, \}$ (this was implicit from parsing algorithm in §1.3)

Theorem. *(unique readability)*
The five formula building operations, when restricted to wff,
a) have ranges that are disjoint from each other and from the set of sentence symbols
b) are 1-1
in other words, the set of wff is freely generated from $\{$sentence symbols$\}$ by $\{\varepsilon_\neg, \varepsilon_\wedge, \varepsilon_\vee, \varepsilon_\to, \varepsilon_\leftrightarrow, \}$

Proof. pg 40-41

Claim. *Let S be set of all or some sentence symbols, truth assignments $v : S \to \{F, T\}$, $\bar{S}$ generated from S by fbos*
Then $\exists!$ extension $\bar{v} : \bar{S} \to \{F, T\}$ of v with the desired properties

Proof. apply unique readability and recursive theorems

Example. want function which gives length of each wff
recursive thm $\Rightarrow \exists! h$ defined on the set of wff s.t.:
a)$\bar{h}(A) = 1$ for each sentence symbol A
b)$\bar{h}((\neg A)) = 3 + \bar{h}(\alpha)$
c)$\bar{h}((\alpha \square \beta)) = 3 + \bar{h}(\alpha) + \bar{h}(\beta)$ where $\square$ is $\wedge, \vee, \to, \leftrightarrow$

Remark.

- our induction principle is not the only one possible
- can give proofs by induction (and definitions by recursion) for length of expressions, number of places with connective symbols, etc. This is less basic but may be necessary.

1.5. SENTENTIAL CONNECTIVES.

Remark. Would we gain anything by adding more connectives to the language? Would we lose anything by omitting some we already have?
Will make these questions precise and give some answers.

Example. sometimes adding connective results in no gain
define sixth formula building operation $\varepsilon_\#(\alpha, \beta, \gamma) = (\#\alpha\beta\gamma)$ (note: 3-place connective) and $\bar{v}((\#\alpha\beta\gamma)) =$ majority of $\bar{v}(\alpha), \bar{v}(\beta), \bar{v}(\gamma)$
but this is tautologically equivalent to $(\alpha \wedge \beta) \vee (\alpha \wedge \gamma) \vee (\beta \wedge \gamma)$

Remark. easiest to define everything in terms of functions

Definition.

- k-space boolean function ("boolean function") is a function from $\{F, T\}^k$ into $\{F, T\}$
- 0-space boolean function is F or T themselves

Example. Let $X \in \{F, T\}$. Then some boolean functions are:
$I_i^n(X_1, \ldots, X_n) = X_i$
$N(F) = T, N(T) = F$
$K(T, T) = T, K(F, X) = K(X, F) = F$
$A(F, F) = F, A(T, X) = A(X, T) = T$
$C(T, F) = F, C(F, X) = C(X, T) = T$
$E(X, X) = T, E(T, F) = E(F, T) = F$

Definition. let wff α with sentence symbols $A_1, \ldots, A_n$
n-place boolean function $B_\alpha^n(X_1, \ldots, X_n)$ realized by α matches truth value of α when $v(A_i) = X_i$.
in other words, $B_\alpha^n(X_1, \ldots, X_n) = \bar{v}(\alpha)$ where truth assignments $v(A_i) = X_i$
i.e. fix α, B_α^n is defined $\bar{v}(\alpha)$ as a function of v
notation: can omit superscript n when unambiguous

Example.

- $B_\alpha(\mathbf{X})$ extracted from wff $\alpha = A_1 \wedge A_2$, where $\mathbf{X} \in \{F, T\}^2$, has truth values matching α

A_1	A_2	$(A_1 \wedge A_2)$	$B_\alpha(\mathbf{X})$
F	F	F	F
F	T	F	F
T	F	F	F
T	T	T	T

- From example above,
  $I_i^n = B_{A_i}^n$, $N = B_{\neg A_1}^1$, $K = B_{A_1 \wedge A_2}^2$, $A = B_{A_1 \vee A_2}^2$, $C = B_{A_1 \to A_2}^2$, $E = B_{A_1 \leftrightarrow A_2}^2$
- composing boolean functions
  $B_{\neg A_1 \vee \neg A_2}^2(X_1, X_2) =$
  $= A(N(I_1^2(X_1, X_2)), N(I_2^2(X_1, X_2)))$
  check with Polish notation

Remark. will soon ask whether every boolean function is obtainable in this fashion

Definition. ordering on $\{F, T\}$ by $F < T$ i.e. natural order if $F = 0, T = 1$

Theorem. *15A*
Let α, β be wff where sentence symbols are among $A_1, \cdots, A_n$
then
a) $\alpha \models \beta$ iff $\forall \mathbf{X} \in \{F, T\}^n, B_\alpha^n(\mathbf{X}) \leq B_\beta(\mathbf{X})$
b) $\alpha \models\!\!\!= \beta$ iff $B_\alpha = B_\beta$
c) $\models \alpha$ iff β_α is constant function with value T

Proof. Recall def of $\alpha \models \beta$
$\alpha \models \beta$
$\Leftrightarrow \forall 2^n$ assignments v, $\bar{v}(\alpha) = T \Rightarrow \bar{v}(\beta) = T$
$\Leftrightarrow \forall 2^n$ n-tuples $\mathbf{X}$, $B_\alpha(\mathbf{X}) = T \Rightarrow B_\beta(\mathbf{X}) = T$
$\Leftrightarrow \forall 2^n$ n-tuples $\mathbf{X}$, $B_\alpha(\mathbf{X}) \leq B_\beta(\mathbf{X})$
where $F < T$

Remark. now can identify tautologically equivalent wffs, and dont need the formal language
are there boolean functions which are not realized by a wff? need thm...

Theorem. *15B*
Let G be an n-place boolean function, $n \geq 1$
can find a wff s.t. $G = B_\alpha^n$ i.e. s.t. α realizes G
intuition: every boolean function is realizable

Proof. pg 47-48

Remark. α which realizes G need not be unique, can be any tautologically equivalent wff to α.
It may be of interest to choose α to be as short as possible

Corollary. *we have enough (in fact, more than enough) sentential connectives*

Proof. suppose we can expand the language with connectives (like majority #)
Any wff φ in the expanded language realizes B_φ^n
by thm 15B, we have a wff α in the original language s.t. $B_\varphi^n = B_\alpha^n$
by thm 15C, B_φ^n and B_α^n are tautologically equivalent

Remark. actually, α can be build only with connectives $\wedge, \vee, \neg$

Definition. α is in disjunctive normal form (DNF) means α is disjunction $\alpha = \gamma_1 \vee \cdots \vee \gamma_k$ where each γ_i in a conjunction $\gamma_i = \beta_{i1} \wedge \cdots \wedge \beta_{in}$ where each β_{ij} is a sentence symbol or the negation of a sentence symbol

Remark. advantage of DNF: they explicitly list truth assignments satisfying the formula

Corollary. *15C*
For any wff φ,
can find a tautologically equivalent wff α in DNF

Definition. complete set of connectives means every function $G : \{F, T\}^n \to \{F, T\}$ for $n \geq 1$ can be realized by using only those connectives

Example. $\{\wedge, \vee, \neg\}$ is complete
Pf: use boolean functions K, A, N?
note: this is a property of corresponding boolean functions K, A, N, but connective symbols are more convenient?
now can rewrite any wff in terms of $\{\wedge, \vee, \neg\}$

Theorem. *both $\{\neg, \wedge\}$ and $\{\neg, \vee\}$ are complete*

Proof. book

Remark. showing set of connectives is _not_ complete is usually difficult
method: first show (by induction usually) that for any wff α, using only these connectives, the function B_α^n has some peculiarity
second find some boolean function which lacks that peculiarity

Example. $\{\wedge, \to\}$ is _not_ complete
pf: if sentence symbols are all T, then any formula is also T (by induction) but then there is no tautology for $\neg$
note: the same argument shows that $\{\wedge, \vee, \to, \leftrightarrow\}$ is not complete

Remark. for each n, there are 2^{2^n} n-place boolean functions (and 2^{2^n} n-ary connectives)
want to catalog those for $n \leq 2$

n=0 0-ary connectives

$2^{2^0} = 2$ 0-place boolean functions, call them F and T
call corresponding connective symbols are $\top, \bot$
these are also $wffs$ by themselves $\bar{v}(\bot) = F$ and $\bar{v}(\top) = T$ for every v
e.g. $(A \to \bot) \models (\neg A)$ (check with 2-line truth table)

n=1 unary connectives

$2^{2^1} = 4$ 1-place boolean functions: negation, identity, and two constants
and 4 corresponding connectives
the only interesting one is $\neg$

n=2 binary connectives

$2^{2^2} = 16$ 2-place boolean functions (and 16 corresponding connectives)

symbols	Equivalent	Remark	
	$\top$	2-place constant	essentially
	$\bot$	2-place constant	0-ary
	A	projection	essentially
	B	projection	
	$\neg A$	negation	unary
	$\neg B$	negation	
$\wedge$	$A \wedge B$	and (if $F = 0, T = 1$, this is multiplication in the field $\{0, 1\}$)	
$\vee$	$A \vee B$	or (inclusive)	
$\to$	$A \to B$	conditional	really
$\leftrightarrow$	$A \leftrightarrow B$	biconditional	
$\leftarrow$	$A \leftarrow B$	reversed conditional	
$+$	$(A \vee B) \wedge \neg(A \wedge B)$	exclusive or (A or B but not both) (if $F = 0, T = 1$, this is addition modulo 2 on field $\{0, 1\}$)	binary
$\downarrow$	$\neg(A \vee B)$	nor (neither nor)	
$\mid$	$\neg(A \wedge B)$	nand (not both A and B)	
$<$	$(\neg A) \wedge B$	ordering where $F < T$	
$>$	$A \wedge (\neg B)$	ordering where $F < T$	

n=3 ternary connectives

$2^{2^3} = 256$ ternary connectives
2 are essentially 0-ary, $6(=2 \cdot \binom{3}{1})$ are essentially binary, $30(=10 \cdot \binom{3}{2})$ are essentially binary, so 218 are ternary
e.g. majority (#), minority, ternary addition modulo 2 ($+^3$), see exercises 7,8

Example.

- $\{\mid\}$ and $\{\downarrow\}$ are complete
  pf: for $\mid$, $[\neg\alpha \models\Dashv \alpha \mid \alpha]$, $[\alpha \vee \beta \models\Dashv (\neg\alpha) \mid (\neg\beta)]$
  so $\neg, \vee$, which are complete, can be simulated
  actually, only need $\mid$, which is itself complete
- $\{\neg, \to\}$ is complete
  note: actually 8 of the 10 really binary connectives form a complete set with $\neg$, the two exceptions are $+$ and $\leftrightarrow$ (see exercise 5)
- $\{\bot, \to\}$ is complete
  it is supercomplete since can even realize the two 0-ary binary operations

1.6. SWITCHING CIRCUITS.

Remark.

- consider electrical device, n inputs and 1 output, all inputs, output are either 1 or 0, memoryless
- e.g. electrical AND, OR, NOT gates
- can combine electrical devices into tree with each output represented as a wff
- given some (compete) set of connectives and desired output, want to find the most efficient (in time, space, etc) tautologically equivalent circuit
- bridge circuits are bilateral (pass current either way) so our methods don't apply

1.7. COMPACTNESS AND EFFECTIVENESS.

1.7.1. *COMPACTNESS.*

Theorem. *compactness theorem (recall from §1.2)*
A set Σ of wff is satisfiable (ie $\exists$ a truth assignment satisfying every member) iff every finite subset is satisfiable

Definition. set Σ of wff is finitely satisfiable means every finite subset is satisfiable
then can restate compactness thm: satisfiable $\Leftrightarrow$ finitely satisfiable

Proof. compactness thm
satisfiable $\Rightarrow$ finitely satisfiable
Σ finite then finitely satisfiable $\Rightarrow$ satisfiable
hard part: Σ infinite and finitely satisfiable $\Rightarrow$ satisfiable
See book

Corollary. *17A*
if $\Sigma \models \tau$
then $\exists$ a finite $\Sigma_0 \subseteq \Sigma$ s.t. $\Sigma_0 \models \tau$

Proof. see book
note: this corollary is equivalent to compactness thm (see exercise 3)

1.7.2. *EFFECTIVENESS AND COMPUTABILITY.*

Remark. even though truth table methods are cumbersome, their existence is important

Definition. (informal)
To decide whether $\Sigma \models \tau$, an effective procedure meets three conditions:
1) there must be a program (exact instructions) of finite number of steps (so finite space, time)
2) the program must be executed by a non-technical clerk or a computer, no randomness or approximations
3) must output F or T (or 0 or 1; or yes or no)
note: ignore practical limitations of run time and memory space
note: this informal def of effective is sufficient for assertions "there does exist an effective procedure of a certain sort", but not sufficient for negative results (see ch.3 for *effective*'s precise counterpart recursive)

Theorem. *17B*
$\exists$ an effective procedure that, given finite expression ε, will decide whether or not it is a wff

Proof. see parsing algorithm in §1.3

Remark. change alphabet to $(,), \neg, \wedge, \vee, \to, \leftrightarrow, A, '$ where $A_5 = A''''$ etc.
so 9 symbols corresponding to digits 1-9, and 0 digit for separating expressions

Definition. A set Σ of expressions is decidable means $\exists$ an effective procedure that, given an expression α, will decide whether $\alpha \in \Sigma$

Remark. thm 17B restated: the set of $wffs$ is decidable

Claim.

- *any finite set is decidable*
  pf: procedure: enumerate and check against each member
- *some infinite sets are decidable but not all*
  pf: there are $2^{\aleph_0}$ sets of expressions, and only countably many effective procedures (since the procedure is completely determined by its finite instructions, and there are only $\aleph_0$ finite sequences of letters, and the instructions, when written out, form a finite sequence of letters)

Theorem. *17C*
$\exists$ an effective procedure that, given finite set of $wffs$ $\Sigma; \tau$, will decide whether $\Sigma \models \tau$
note: "given" implies finite since cant be given infinite object

Proof. truth table method in §1.2 is effective

Corollary. *17D*
For a finite set Σ, the set of tautological consequences of Σ is decidable
in fact, the set of tautologies is decidable

Remark.

- if Σ is infinite set - even a decidable one then in general its set of tautological consequences may not be decidable (see ch3)
- but we can get a weaker result ie half of decidability

Definition. set A of expressions is effectively enumerable means $\exists$ an effective procedure that lists, in some order, the members of A

Remark. if set A is infinite, effectively enumerable then the procedure can never finish
but for any specified member of A, it must eventually (in finite time) appear in the list

Theorem. *17E*
set A of expression is effectively enumerable
iff $\exists$ an effective procedure that, given any expression ε, produces a "yes" iff $\varepsilon \in A$

Remark. if $\varepsilon \notin A$, then the procedure will go on forever without producing any answer, this procedure is called half of a decision procedure

Definition. set A of expressions is semidecidable means $\exists$ an effective procedure that, given any expression ε, produces a "yes" iff $\varepsilon \in A$

Theorem. *17E restated*
a set is effectively enumerable iff it is semidecidable

Proof. thm 17E
see book

Remark. decidable $\Rightarrow$ semidecidable
intuition: "no" bulb is burned out

Theorem. *17F (Kleene's)*
A set of expressions is decidable
iff it and its complement (relative to the set of all expressions) are effectively enumerable

Proof. exercise 8

Remark. The class of decidable sets is closed under union, intersection, and complementation.
Pf: if sets A, B are effectively enumerable sets, then so are $A \cup B$, $A \cap B$
For complementation, see thm 17F

Theorem. *17 G*
if Σ is a decidable set of $wffs$ (or even if Σ is effectively enumerable)
then the set of tautological consequences of Σ is effectively enumerable

Proof. see book (its short)

Remark. later, will want effective procedures to compute functions
will say f is effectively enumerable ("computable") iff $\exists$ an effective procedure that, given an input x, will eventually produce the correct output $f(x)$