Lecture notes
Einführung in die Logik 2024W

Petermann

Contents

List of Abbreviations

CHAPTER 1

Propositional logic

Definition 1.1. ***Language of PL:*** The Language of Propositional logic is a set containing

- logical symbols: consisting of the **sentential connective** symbols $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ and parenthesis $(,)$
- non-logical symbols: $A_1, A_2, A_3, \ldots$ (also called sentential atoms, variables)

from which we assume (for unique readability) that no symbol is a finite sequence of any other symbols.

Note:

1. The role of the logical symbols doesn't change, the sentential atoms we see as variables, they function as placeholders or variables.
2. we assumed the set of non-logical symbols is countable, for most of our conclusions you could use any set of prop. atoms of any size

Definition 1.2. ***Expression / prop. sentence:*** An **expression** is a any finite sequence of symbols We define **grammatically correct exp.** recursive

1. every prop. atom is a prop. sentence
2. if α, β are prop. sentences, then also $\neg\alpha, \alpha \wedge \beta, \alpha \vee \beta, \alpha \rightarrow \beta, \alpha \leftrightarrow \beta$
3. nothing else

and call them **prop. sentences.** Equivalently stated every prop. sentence. is built up by applying finitly many operations TODO This allows us to symbolize the **expression tree**

Definition 1.3. ***Construction sequence:*** Given a prop. sentence α a construction sequence of α is a finite sequence $\langle \alpha_1, \ldots \alpha_{n-1}, \alpha \rangle$ such that for all $i \leq n$ the following holds

- α_i is a sentential atom
- or $\alpha_i = \varepsilon_\neg(\alpha_j)$ for some $j < i$
- or $\alpha_i = \varepsilon_\square(\alpha_j, \alpha_k)$ for some $j, k < i$ and $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$

Definition 1.4. ***:*** Let S be a set. We say S is **closed** under an n-ary operational symbol f iff for all $s \in S$ it holds $f(s) \in S$

Induction principle: Suppose S is a set of prop. sentences containing all prop. atoms and closed under the 5 formula building operations, then S is the set of all prop. sentences.

Proof. let $PS =$ set of all prop. sent.

$S \subseteq PS$: is clear

$S \supseteq PS$: let $\alpha \in PS$ then α has a construction seq. $\langle \alpha_1, \ldots \alpha_{n-1}, \alpha \rangle$ and $\alpha_1 \in S$ lets assume that α_k for $k < n$ is in S then α_{k+1} is either an atom and therefore in S or its obtained by one of the formula building operations and therefore $\alpha_{k+1} \in S$

$\square$

1.1 Truth assignments

We will answer the question when does a prop. sent. follow from other prop. sentences.

Definition 1.5. ***Truth assignment:*** Let $\{0,1\}$ be the set of truth values. A truth assignment (TA) for a set S of prop. atoms is a map $\nu : S \to \{0,1\}$

We now want to extend ν to $\overline{\nu} : \overline{S} \to \{0,1\}$, where $\overline{S}$ is the closure of S under the 5 fla. building operations such that

1. $\overline{\nu}(A) = \nu(A)$
2. $\overline{\nu}(\neg\alpha) = 1 - \nu(\alpha)$
3. $\overline{\nu}(\alpha \wedge \beta) = \begin{cases} 1 & \text{iff } \overline{\nu}(\alpha) = 1 = \overline{\nu}(\beta) \\ 0 & \text{otherwise} \end{cases}$
4. $\overline{\nu}(\alpha \vee \beta) = \begin{cases} 1 & \text{iff } \overline{\nu}(\alpha) = 1 \text{ or } \overline{\nu}(\beta) = 1 \\ 0 & \text{otherwise} \end{cases}$
5. $\overline{\nu}(\alpha \to \beta) = \begin{cases} 1 & \text{iff } \overline{\nu}(\alpha) = 0 \text{ or } \overline{\nu}(\beta) = 1 \\ 0 & \text{otherwise} \end{cases}$
6. $\overline{\nu}(\alpha \leftrightarrow \beta) = \begin{cases} 1 & \text{iff } \overline{\nu}(\alpha) = \overline{\nu}(\beta) \\ 0 & \text{otherwise} \end{cases}$

Theorem 1.1. ***:*** $\forall$ TA ν for a set S $\exists! \overline{\nu} : \overline{S} \to \{0,1\}$ satisfying the above properties

We will proof this later

Definition 1.6. ***Satisfaction:*** A TA ν satisfies a prop. sent. α iff $\overline{\nu}(\alpha) = 1$ (that is, provided that everery atom of α is in the domain of ν)

Definition 1.7. ***Tautological implication:*** Let Σ be a set of prop. sent. and α a prop. sent. then we say: Σ tautologically imlies α iff $\forall$ TA that satisfies Σ then α is also satisfied and we write $\Sigma \models \alpha$
If $\Sigma = \{\beta\}$, we simply write $\beta \models \alpha$ If $\Sigma = \varnothing$ then we write $\models \alpha$ for $\varnothing \models \alpha$ and α is called a **tautology**
α, β are called **tautologically equivalent** iff $\alpha \models \beta$ and $\beta \models \alpha$ we then write $\alpha \mathrel{=\!\mid\models} \beta$

Note: Suppose there is no TA that satisfies Σ, then we have $\Sigma \models \alpha$ for every prop. sent. α

Example 1.1. ***:*** $\{\neg A \vee B\} \mathrel{=\!\mid\models} A \to B$

Note: In order to check if a prop. sent. is satisfiable we need to check 2^N TAs, where $N = \#$ of atoms. It is unknown if this can be done by an algorithm in polynomial time. Answering this would settle the debate whether $P = NP$

TODO: Add section here?

Theorem 1.2. ***Compactness theorem:*** Let Σ be an infinite set op prop. sent. such that

$$\forall \Sigma_0 \subseteq \Sigma, \Sigma_0 \text{finite} \exists \text{ TA satisfying every member of } \Sigma_0$$

then there is a TA satsfying every member of Σ.

Proof. Let $\mathcal{A} = \{A_0, A_1, \dots\}$ be the set of all prop. atoms. We are going to identify TAs with elements in $\{0,1\}^{\mathcal{A}} := \{f : \mathcal{A} \to \{0,1\}\}$ TODO □

1.2 A PARSING ALGORITHM

To prove Thm. Theorem 1.1 we essentially need to show that we have enough parenthesis to make the reading of a prop. sent. unique. TODO Bsp

Lemma 1.1. *:* Every prop. sent. has the same number of left and right parenthesis.

Proof. Let $M =$ set of prop. sent. w/ # left parenthesis = # right parenthesis and
$PS =$ set of all prop. sent. We have $M \subseteq PS$. Since atoms have no parenthesis, they are in M. we just need to show that M is closed under the 5 construction operations.
$\varepsilon_\neg = (\neg\alpha)$ ... □

Lemma 1.2. *:* No proper initial segment of a prop. sent. is itself a prop. sent.

Proof. Let $\alpha = \alpha_1\alpha_2 \dots \alpha_n$ be a prop. sent. By proper initial segment we understand $\beta = \alpha_1 \dots \alpha_i$ for $1 \leq i < n$. We will prove that every proper initial segment has an excess of left parenthesis, then we use the previous lemma.

- Atoms: since the empty sequence is no prop. sent. they have no proper initial segment.
- If the above is true for α, β then the proper initial segments of $(\neg\alpha)$ are of the form

  $(\neg\alpha$
  $(\neg\alpha'$ where α' is a propper initial segment of α
  $($ or
  $(\neg$

  Therefore $\varepsilon_\neg$ preserves this property and under $\varepsilon_\wedge, \varepsilon_\vee, \varepsilon_\rightarrow, \varepsilon_\leftrightarrow$ this is also the case.

□

Parsing algorithm

We now give a parsing algorithm procedure. For input we take some expression τ and the algorithm will determine if τ is a prop. sent. If so, it will generate a unique construction tree (in form of a rooted tree) for τ.

0. create the root and label it τ
1. HALT if all leaves are labled w/ prop. atom and return: "τ is a prop. sent."
2. select a leaf of the graph which is not labled w/ prop. atom
3. if the first symbol of label under consideration is not a left parenthesis, then halt and return: "τ is not a prop. sent."
4. if the second symbol of the label is "$\neg$" then GOTO 6.
5. scan the expression from left to right
   if we reach a proper initial segment of the form "$(\beta$" where $\#lp(\beta) = \#rp(\beta)$ and β is followed by one of thesection $\wedge, \vee, \rightarrow, \leftrightarrow$ and the remainder of the expression is of the form $\beta')$, where $\#lp(\beta') = \#rp(\beta')$

   Then: create two child nodes (left,right) to the selected element and label them (left := β, right := β') GOTO 1.

   Else: HALT and return "τ is not a prop. sent."
6. if the expression is of the form $(\neg\beta)$ where $\#lp(\beta) = \#rp(\beta)$

   Then: construct one childnode and label it β and GOTO 1.

   Else: HALT and return: "τ is not a prop. sent."

Example 1.2. *TODO :*

Correctness of the parsing algorithm

- The algorithm always halts, because the length of a child is less than the label of a parent.
- If the algorithm halts with the conclusion that τ is a prop. sent. then we can prove inductively (starting from the leaves) that each label is a prop. sent
- Unique way to make choices in the algorithm: in particular β, β' in step 5. If there was a shorter choice for β it would be a proper initial segment of β but such prop. sent. can not exist. (This also works under the assumption that a longer choice exists).
- rejections are made correctly

Back to proving the existence and uniqueness of $\overline{\nu}$ in Theorem 1.1. Let α be a prop. sent. of $\overline{S}$. We apply the parsing algorithm to α to get a unique construction tree For the leaves, use ν go get the truth values then work our way up using the conditions (1-6) in Definition 1.5.

A more formal notation

TODO

1.3 Induction and recursion

A simple case: let $\mathcal{U}$ be a set and $B \subseteq \mathcal{U}$ our initial set. $\mathcal{F} = \{f, g\}$ a class of functions containing just f and g, where

$$f : \mathcal{U} \times \mathcal{U} \to \mathcal{U}, \qquad g : \mathcal{U} \to \mathcal{U}$$

We want to construct the smallest subset $\mathcal{C} \subseteq \mathcal{U}$ such that $B \subseteq \mathcal{C}$ and $\mathcal{C}$ is closed under all elements of $\mathcal{F}$.

Definition 1.8. ***Closedness, Inductiveness:*** We say $\mathcal{C}$ is

- **closed** under f and g iff $\forall x, y \in \mathcal{C}\ (f(x, y) \in \mathcal{C} \land g(x) \in \mathcal{C})$
- **inductive** if $B \subseteq \mathcal{C}$ and $\mathcal{C}$ is closed under $\mathcal{F}$

Big TODO

1.4 Sentential connectives

Definition 1.9. ***Tautological equivalence relation:*** For α, β prop. sent. we define $\alpha\ \beta$ iff $\alpha \mathrel{=\!\mid\!\models} \beta$. This defines an equivalent relation.

Example 1.3. ***:*** $A \to B \mathrel{=\!\mid\!\models} \neg A \lor B$

Note: A k-place boolean function is a functon of the form $f : \{0,1\}^k \to \{0,1\}$ and we define $0, 1$ as the 0-place boolean functions.
If α is a prop. sent. then it determines a k-place boolean function, where k is the number of atoms, α is built up from. If α is $A_1 \lor \neg A_2$ then $B_\alpha : \{0,1\}^2 \to \{0,1\}$ and asign its values corresponding a truth table. TODO extend / rearange function

Theorem 1.3. ***:*** If α, β are prop. sent. with at most n prop. Atoms (combined), then

1. $\alpha \models \beta$ iff $\forall x \in \{0,1\}^n$ it holds $B_\alpha(x) \leq B_\beta(x)$
2. $\alpha \mathrel{=\!\mid\!\models} \beta$ iff $\forall x \in \{0,1\}^n$ it holds $B_\alpha(x) = B_\beta(x)$
3. $\models \alpha$ iff $\forall x \in \{0,1\}^n$ it holds $B_\alpha(x) = 1$

Theorem 1.4. ***Realisation:*** Let G be an n-ary boolean function for $n \geq 1$. Then there is a prop. sent. α such that. $B_\alpha = G$. We say α realizes G.

Proof. 1. if G is constantly equal to 0 then set α to $A_1 \land \neg A_1$.

2. Otherwise the set of inputs $\{\vec{x}_1, \vec{x}_2, \dots \vec{x}_k\}$ for which $G(\vec{x}_i) = 1$ holds is not empty.
   We denote $\vec{x}_i = (x_{i1}, x_{i2}, \dots x_{in})$ and define a matrix $(x_{ij})_{k\times n}$ We further set $\beta_{ij} = \begin{cases} A_j & \text{iff } x_{ij} = 1 \\ \neg A_j & \text{iff } x_{ij} = 0 \end{cases}$
   Example:
   $$(x_{ij}) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \rightsquigarrow \begin{pmatrix} \neg A_1 & A_2 & \neg A_3 \\ A_1 & A_2 & \neg A_3 \end{pmatrix} = (\beta_{ij})$$
   We define γ_i as $\beta_{i1} \wedge \beta_{i2} \wedge \dots \beta_{in}$ for $1 \le i \le k$
   and α as $\gamma_1 \vee \gamma_2 \vee \dots \gamma_k = \vee_{i=1}^{k} \gamma_i$ Then $B_\alpha = G$ is fulfilled.

□

Note: α as constructed in the proof is in the so-called Disjunctive normal form (DNF).

Corollary 1.4. Every prop. sent. is tautologically equivalent to a sentence in DNF

Corollary 1.4. $\{\neg, \wedge, \vee\}$ is a complete set of logical connectives, i.e. every prop. sent. is tautologically equivalent to a sentence built up from atoms and $\neg, \wedge, \vee$.

Theorem 1.5. ***:*** Both $\{\neg, \wedge\}$ and $\{\neg, \vee\}$ are complete.

Proof. Its sufficient to show that every k-place boolean function is realisable by a prop. sent. built up using only $\neg$ and $\wedge$. This is, because $\alpha \wedge \beta \models\!\mid\models \neg(\neg\alpha \vee \neg\beta)$ We prove this by induction over the number of disjuctions of a prop. sent. α in DNF. Suppose the statement is true for $k \le n$. For $n+1$ and $\alpha = \bigvee_{j=1}^{n+1} \gamma_j$ there exists an $\alpha' \models\!\mid\models \bigvee_{j=1}^{n} \gamma_j$ and

$$\alpha = \bigvee_{j=1}^{n+1} \gamma_j \models\!\mid\models \alpha' \vee \gamma_{n+1} \models\!\mid\models \neg(\neg\alpha' \wedge \neg\gamma_{n+1})$$

□

Note: We used the observation that, if $\alpha \models\!\mid\models \beta$ and we replace a subsequence of α by a so called tautological equivalence then the result is also tautologically equivalent to β
TODO S.10

Example 1.4. $\{\to, \wedge\}$ ***is not complete.:*** Let $\alpha \in PS$ built up from only $\to, \wedge$ from the atoms $A_1, \dots A_n$ then we claim

$$A_1 \wedge A_2 \wedge \dots \wedge A_n \models \alpha$$

We can also say $\{\to, \wedge\}$ is not complete bc. $\neg A$ is not tautological equivalent to a sent. built up from $\to, \wedge$

Proof. Let $C := \{\alpha \in PS$ built up from $\to, \wedge$ and $A_1, \dots A_n$ for which $\bigwedge_{i=1}^{n} A_i \models \alpha\}$ we want to show that $C = \{\alpha \in PS$ built up from $\to, \wedge$ and $A_1, \dots A_n\}$

- We have $\{A_1, A_2 \dots, A_n\} \subseteq C$
- for $\alpha, \beta \in C$ it holds
  (1) $A_1 \wedge \dots \wedge A_n \models \alpha \to \beta$
  (2) $A_1 \wedge \dots \wedge A_n \models \alpha \wedge \beta$

Therefore C is closed under the fla. building operations and we have proven our claim. □

Note: $\{\wedge, \vee, \to, \leftrightarrow\}$ is still not complete.
Note: The number of n-ary boolean functions existing is 2^{2^n} We define a notation for $n = 0$: $\perp$ (for TV $= 0$) and $\top$ (for TV $= 1$) We can conclude that $\{\neg, \to\}$ and $\{\to, \perp\}$ are both complete, it holds $\neg A \models\!\mid\models A \to \perp$

Definition 1.10. ***Satisfiability:***
A set of prop. sent. Σ is called **satisfiable** iff $\exists$ TA that satisfies every member of Σ.

1.5 Compactness Theorem

Theorem 1.6. ***Compactness Theorem:*** Σ is satisfiable iff every finite subset $\Sigma_0 \subseteq \Sigma$ is satisfiable. (i.e. Σ is finitely satisfied)

Proof. Let Σ be a finitely satisfiable set of prop. sent. Outline of the proof:

1. extend Σ to a maximal finitely satisfiable set Δ of prop. sent.
2. construct a thruth assigment using Δ

1. Let $\alpha_1, \alpha_2, \ldots$ be an enumeration of all prop. sent. and define Δ_n inductively by $\Delta_0 := \Sigma$

$$\Delta_{n+1} := \begin{cases} \Delta_n \cup \{\alpha_{n+1}\} & \text{if satisfiable} \\ \Delta_n \cup \{\neg\alpha_{n+1}\} & \text{otherwise} \end{cases}$$

   Claim: Δ_n is finitely satisfiable for each n
   proof of claim. By regular induction over n. Δ_0 is finitely satisfiable. Let us assume Δ_n is finitely satisfiable. If $\Delta_{n+1} = \Delta_n \cup \{\alpha_{n+1}\}$ then we are finished. Otherwise let $\Delta' \subseteq \Delta_n$ be a finite set that $\Delta' \cup \{\alpha_{n+1}\}$ is not satisfiable. It holds $\Delta' \models \neg\alpha_{n+1}$. We assume that $\Delta_n \cup \{\neg\alpha_{n+1}\}$ is not finitely satisfiable. Then there exists a finite subset $\Delta'' \subseteq \Delta_n$ such that $\Delta'' \cup \{\neg\alpha_{n+1}\}$ is (finite and) not satisfiable. It therefore holds $\Delta'' \models \alpha_{n+1}$ But $\Delta' \cup \Delta''$ is a finite subset of Δ_n and by above observations $\Delta' \cup \Delta'' \models \alpha_{n+1}$ and $\Delta' \cup \Delta'' \models \neg\alpha_{n+1}$ A contradiction to the assumption that Δ_n is finitely satisfiable. ⊠

   We set $\Delta := \bigcup_{i \in \mathbb{N}} \Delta_i$ and get
   (a) $\Sigma \subseteq \Delta$
   (b) (Maximality): for every prop. sent. α it holds $\alpha \in \Delta$ or $\neg\alpha \in \Delta$
   (c) (Satisfiability): Δ is finitely satisfiable. For every finite subset there exists a Δ_n which is a superset.

2. Let ν be a TA for the prop. atoms $A_1, A_2, \ldots$ such that $\nu(A) = 1$ iff $A \in \Delta$
   Claim: For every prop. sent. φ it holds $\overline{\nu}(\varphi) = 1$ iff $\varphi \in \Delta$.
   proof of claim. Let $S = \{\varphi \in PS \text{ s.t. } \overline{\nu}(\varphi) = 1 \text{ iff } \varphi \in \Delta\}$.

   - $PS \supseteq S$ is clear.
   - $PS \subseteq S$
     (a) $\{A_1, A_2 \ldots\} \subseteq S$ by definition of ν
     (b) closure under $\epsilon_\neg$: Let $\varphi \in S$ then we get by maximality and satisfiability of Δ:

$$\begin{aligned} & \overline{\nu}(\neg\varphi) = 1 \\ \text{iff} \quad & \overline{\nu}(\varphi) = 0 \\ \text{iff} \quad & \varphi \notin \Delta \\ \text{iff} \quad & (\neg\varphi) \in \Delta \end{aligned}$$

     closure under $\epsilon_\to$: Let $\varphi_1, \varphi_2 \in S$ similiarly

$$\begin{aligned} & \overline{\nu}(\varphi_1 \to \varphi_2) = 0 \\ \text{iff} \quad & \overline{\nu}(\varphi_1) = 1 \text{ and } \overline{\nu}(\varphi_2) = 0 \\ \text{iff} \quad & \varphi_1 \in \Delta \text{ and } \varphi_2 \notin \Delta \\ \text{iff} \quad & (\varphi_1 \to \varphi_2) \notin \Delta \end{aligned}$$

     The closure under the other fla. building operations are similar. ⊠

   By this claim $\overline{\nu}$ satisfies Σ. □

Corollary 1.6. If $\Sigma \models \tau$ then there exists a finite subset $\Sigma' \subseteq \Sigma$ s.t. $\Sigma' \models \tau$

Proof. Recall: $\Sigma \models \tau$ iff $\Sigma \cup \{\neg\tau\}$ is not satisfiable. Suppose $\Sigma \models \tau$ but no finite subset does.
Then $\forall \Sigma' \subseteq \Sigma$ finite $\Sigma' \cup \{\neg\tau\}$ is satisfiable. By the compactness theorem $\Sigma \cup \{\neg\tau\}$ is satisfiable which is a contradiction to $\Sigma \models \tau$. □

Note: Theorem 1.6 and Corollary 1.6 are equivalent.

Chapter 2

Predicate - / first order logic

Definition 2.1. ***A First order Language:*** consists of infinetely many distinct symbols such that no symbol is a proper initial segment of another symbol and the symbols are divided into 2 groups:

1. logical symbols (These elements have a fixed meaning and the equivalence symbol $=$ is optional)

   $(,), \neg, \rightarrow, v_1, v_2, \ldots, =$

2. parameters
   - quantifier symbol: $\forall$ (the range is subject of interpretation)
   - predicate symbols: $\forall n > 0$ we have a set of n-ary predicates
   - constant symbols: Some set of constants (could be $\varnothing$)
   - function symbols: $\forall n > 0$ we have a set of n-ary function symbols

Note:

- We could drop constants and instead introduce 0-ary function symbols
- to specify language we need to specify the parameters and say if $=$ is included

Example 2.1. *:*

- $\mathcal{L}_{\text{set}} = \{\in\}, \quad =$ included
- $\mathcal{L}_{\text{arith}} = \{<, 0, S, E, +, \cdot\}$

$$\begin{aligned} = &\text{ included} \\ < &\text{ is a binary rel. symbol} \\ 0 &\text{ is a constant} \\ S &\text{ is a unary function symbol} \\ E &\text{ exponentiation TODO} \\ +, \cdot &\text{ binary function symbols} \end{aligned}$$

2.1 Formulas

Definition 2.2. ***Expression:*** An **expression** is any finite sequence of symbols. There exist two kinds of expressions

Terms:
- the names of objects
- they are built up from variables and constants (by use of polish notation)

Formulas:
- They express assertions about objects,
- they are built up from atomic formulas
- atomic formulas these are built up from terms using predicate symbols and $=$

Definition 2.3. ***Building Operations:*** $\forall n > 0$ and for every n-place function symbol f let $\mathcal{F}_f$ be an n-place term building operation, that is $\mathcal{F}_f(\alpha_1, \ldots \alpha_n) := f(\alpha_1, \ldots \alpha_n)$ The Set of Terms we then define as the set of expressions that are built up from variables and constants by applying the building operations finitely many times.

Example 2.2. : Let $\mathcal{L} = \mathcal{L}_{arith}$ then the set of terms will contain 0, v_{42}, $S0$, $SSS0$, Sv_1, $+SOv_1$

Definition 2.4. *Atomic formula:* Any expression of the form

$$t_1 = t_2 \text{ of } P(t_1, \dots t_n), \text{ where } t_1, \dots t_n \text{ are terms and } P \text{ is an } n\text{-ary predicate symbol}$$

Note: Atomic formulas are not defined inductively.

Example 2.3. : *cont.* $= v_1 v_{42}$, $< S0SS0$ are atomic formulas, but $\neg = v_1 v_{42}$ is not.

Definition 2.5. *Formulas:* Let $\varepsilon_\neg$, $\varepsilon_\to, Q_i$ be fla. building operations $\varepsilon_\neg(\alpha) = (\neg\alpha)$, $\varepsilon_\to = (\alpha \to \beta)$ and $Q_i(\gamma) = \forall v_i \gamma$ The set of formulas is the set of expressions built up from atomic formulas by applying the fla. building operations finitely many times.

Free variables

Example 2.4. : "Every non-zero natual number is a succsesor" $\forall x(x \neq 0 \to \exists y S(y) = x)$ is different then "if a number is not 0, then it is a succsesor" $x \neq 0 \to \exists y S(y) = x$. In the latter, x occures free in the fla.

Definition 2.6. *Free variables:* Let x be a variable. x occurs **free in** ϕ is defined inductively as follows:

1. If ϕ is an atomic fla. then x occurs **free** in ϕ iff x occurs in ϕ
2. If $\phi = (\neg\alpha)$ then x occurs free in ϕ iff x occurs free in α
3. If $\phi = (\alpha \to \beta)$ then x occurs free in ϕ iff x occurs free in α or β
4. If $\phi = \forall v_i \alpha$ then x occurs free in ϕ iff x occurs free in α and $x \neg v_i$

TODO

2.2 Semantics of first order logic

Definition 2.7. *structure:* A **structure** $\mathcal{A}$ for a first order language $\mathcal{L}$ is a non-empty set set A called **universe** or **underlying set of** $\mathcal{A}$ together with an interpretation of each parameter of $\mathcal{L}$ i.e.

- $\forall$ ranges over the universe A
- for an n-ary pred. symbol $P \in \mathcal{L}$ its interpretation $P\mathcal{A}$ is a subset of A^n
- for a constant $c \in \mathcal{L}$ its interpretation $c\mathcal{A}$ is an element of A
- for an n-ary function symbol $f \in \mathcal{L}$ its interpretation $f^{\mathcal{A}}$ is a total function $f^{\mathcal{A}} : A^n \to A$

Example 2.5. : Let $\mathcal{L} = \{\in\}$ where $\in$ is a binary relation " An example of an $\mathcal{L}$ structure is $(\mathbb{N}, \in^{\mathbb{N}})$ where $\in^{\mathbb{N}} = \{(x, y) \in \mathbb{N}^2 : x < y\}$

2.3 Logical implication

2.4 Definability in a structure

2.5 Homomorphisms of structures

2.6 A parsing algorithm for first order logic

2.7 Unique readability for terms

2.8 Deductions (formal proofs)

2.9 Generalization and deduction theorem

TODO evt noch sectioons

Chapter 3

Boolean Algebra

Definition 3.1. ***Boolean Algebra:*** A **boolean algebra** is a set B with

- distinguished elements $0, 1$ (called zero and unit of B)
- a unary operation $'$ on B (called **complementation**)
- two binary operations $\vee$ called **join** and $\wedge$ called **meet** s.t. for all $x, y, z \in B$
  1. $x \vee 0 = x \qquad x \wedge 1 = x$
  2. $x \vee x' = 1 \qquad x \wedge x' = 0$
  3. $x \vee y = y \vee x \qquad x \wedge y = y \wedge x$
  4. $(x \vee y) \vee z = x \vee (y \vee z) \qquad (x \wedge y) \wedge z = x \wedge (y \wedge z)$
  5. $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \qquad x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$

Example 3.1. ***:*** Let S be a set, $B := \mathcal{P}(S)$ the power set of S, $0 := \varnothing$ and $1 := S$,

$$' : \mathcal{P}(S) \to \mathcal{P}(S), x' := S \backslash x \qquad x \vee y := x \cup y, \quad x \wedge y := x \cap y \text{ for } x, y \in \mathcal{P}(S)$$

Lemma 3.1. ***:*** Let $(B, ', \vee, \wedge, 0, 1)$ be a boolean algebra. Then it holds

a) $0' = 1$, $1' = 0$

b) $x \vee x = x$, $x \wedge x = x$

c) $(x')' = x$

d) $(x \vee y)' = x' \wedge y'$, $(x \wedge y)' = x' \vee y'$

e) $x \vee y = y$ iff $x \wedge y = x$

Lemma 3.2. ***:***

a) $x \leq y :\Leftrightarrow x \vee y = y$ defines a partial ordering on B (inclusion) and it holds

b) $x \vee y$ is the least upper bound of $\{x, y\}$ in B
$x \wedge y$ is the greatest lower bound of $\{x, y\}$ in B

c) $0 \leq x \leq 1$ for all $x \in B$

Note: A boolean algebra is a complemented distributive lattice.

Definition 3.2. ***Opposite of boolean algebra:*** Let $(B, ', \vee, \wedge, 0, 1)$ be a boolean algebra. The boolean algebra B^{op} is defined by

$$B^{\text{op}} := B, \quad 0^{\text{op}} := 1, \quad 1^{\text{op}} := 0, \quad ' \text{ stayes the same as for} B, \quad \vee^{\text{op}} := \wedge, \quad \wedge^{\text{op}} := \vee$$

Note: $(B^{\text{op}})^{\text{op}} = B$

Definition 3.3. ***Subalgebra:*** A **subalgebra** of B is a subset $A \subseteq B$ s.t. $0, 1 \in A$ and A is closed under $', \wedge, \vee$. The subalgebra generated by $P \subseteq B$ is defined to be the smallest subalgebra containing P. Equivalently it is the intersection of all Subalgebras of B that contain P.

Example 3.2. ***Power set algebra:*** Let S be a set then $\mathcal{P}(S)$ defines a boolean algebra on S. $B := \{x \in \mathcal{P}(S) : x \text{is finite or cofinite}\}$ is a subalgebra of $\mathcal{P}(S)$ w/ set of generators $\{\{s\} : s \in S\}$

Note: We will prove the Tarski-Stone Theorem: every boolean algebra is isomorphic to an algebra on a set.

Example 3.3. ***Lindenbaum Algebra of*** Σ***:*** Let A be a set of prop. atoms, Prop(A) the set of prop. generated by A. Further let $\Sigma \subseteq \text{Prop}(A)$ and p, q, r range over Prop(A).
We say p is Σ-equivalent to q iff $\Sigma \models_{\text{taut}} p \leftrightarrow q$ Σ-Equivalence is an equivalent relation on Prop(A) and Prop$(A)/\Sigma$ is a boolean algebra with

$$0 := \bot/\Sigma, \quad 1 := \top/\Sigma, \quad (p/\Sigma)' := (\neg p)/\Sigma, \quad (p/\Sigma \vee q/\Sigma) := (p \vee q)/\Sigma, \quad (p/\Sigma \wedge q/\Sigma) := (p \wedge q)/\Sigma$$

a set of generators is $\{a/\Sigma : a \in A\}$

Definition 3.4. ***Homomorphisms of boolean algebras:*** Let B, C be boolean algebras. A map $\phi : B \to C$ is a (homo)morphism of boolean algebras iff $\forall x, y \in B$ it holds

- $\phi(0_B) = 0_C$
- $\phi(x') = \phi(x)'$
- $\phi(x \vee y) = \phi(x) \vee \phi(y)$
- $\phi(x \wedge y) = \phi(x) \wedge \phi(y)$

If $\phi : B \to C$ is bijective too , we call ϕ an isomorphism and $\phi^{-1} : C \to B$ is also a morphism of boolean algebras.

Note: $\phi(B)$ is subalgebra of C

Example 3.4. ***:*** Let S, T be sets then a function $f : S \to T$ induces a morphism of boolean algebras $\mathcal{P}(T) \to \mathcal{P}(S) : y \mapsto f^{-1}(y)$ If $S \subseteq T$ and f the inclusion map $S \hookrightarrow T$ then we get a boolean algebra morphism $Y \to Y \cap S$.
• $id_B : B \to B$ • $x \mapsto x' : B \to B^{\text{op}}$ are both isomorphism

Note: A boolean algebra morphism $\phi : B \to C$ is injective iff $\ker f = 0_B$

Lemma 3.3. ***:*** Let $X_1, \dots X_m \subseteq S$ and $\mathcal{A}$ a boolean algebra on S generated by $\{X_1, \dots X_m\}$. Then $\mathcal{A}$ is finite and isomorphic to $\mathcal{P}(\{1, 2, \dots n\})$ for some $n \leq 2^m$.

Proof. TODO ☐

Definition 3.5. ***Trivial algebras:***

- B is trivial if $|B| = 1$ (equivalently $0 = 1 \in B$) according to Lemma 3.3 B is isomorphic to $\mathcal{P}(\varnothing)$
- If $|S| = 1$ then $|\mathcal{P}(S)| = 2$ TODO

Definition 3.6. ***Ideal:*** An ideal of B is a subset of $I \subseteq B$ s.t.

(I1) $0 \in I$

(I2) $\forall a, b \in B$ it holds $\quad a \leq b$ and $b \in I \implies a \in I \quad$ and $\quad a, b \in I \implies a \vee b \in I$

Example 3.5. ***:*** $F_{\text{in}} = \{F \subseteq S : F \text{ finite}\}$ is ideal in $\mathcal{P}(S)$.

Note: If I is an ideal of B then $I \vee b := \{x \in B : x = a \vee b \text{ for some } a \in I\}$ is the smallest ideal w/ respect of $\subseteq$ of B that contains $I \cup \{b\}$.

Example 3.6. ***:***

- For a boolean algebra morphism $\phi : B \to C$ the kernel $\ker(\phi)$ is an ideal in B.
- If I is an ideal in B then $a =_I b :\Leftrightarrow a \vee x = b \vee x$ for some $x \in I$ defines an equivalent relation and $B/_{=_I}$ is a boolean algebra w/

  $$0 := 0/_{=_I} \quad 1 := 1/_{=_I} \quad (a/_{=_I})' := a'/_{=_I} \quad a/_{=_I} \vee b/_{=_I} := (a \vee b)/_{=_I} \quad a/_{=_I} \wedge b/_{=_I} := (a \wedge b)/_{=_I}$$

  Then $\phi : B \to B/_{=_I} : b \mapsto b/_{=_I}$ is a boolean algebra morphism w/ $\ker(\phi) = I$

Chapter 4

Set Theory

Example 4.1. ***Russel's paradox:*** Let $A = \{a : a \notin a\}$. If any collection of elements is a set, then A would be a set. Question: is $A \in A$? if yes, then $A \notin A$, if not then $A \in A$

Trying to resolve this, we will introduce the ZFC (Zermelo-Frankel axioms w/ choice) System. Let $\mathcal{L} = \{\in\}$ be a Language of first order, where $\in \ldots$ binary relation "beeing element of" For $(\mathcal{U}, \in)$ If $(\mathcal{U}, \in) \models$ ZFC, then the elements of the universe $\mathcal{U}$ are called sets.

TODO

4.1 Axioms of ZFC

Definition 4.1. ***Axiom of extensionality:***

$$\forall x \forall y (x = y \leftrightarrow \forall u (u \in x \leftrightarrow u \in y))$$

Definition 4.2. ***Pairing Axiom:*** for any two sets a, b one can form a set whose elements are precicely a, b

$$\forall x \forall y \exists z (u \in z \leftrightarrow (u = x \vee u = y))$$

Our notation will be $z = \{x, y\}$

Note: $\{x, y\}$ is unique by Definition 4.1

Lemma 4.1. ***:*** Let x, y be sets. We define $(x, y) := \{\{x\}, \{x, y\}\}$. Then it holds $(x, y) = (a, b)$ iff $x = a$ and $y = b$

Proof.

- if $x = y$, then $(x, y) = \{\{x\}\}$ therefore $a = b$ and by Definition 4.1 it holds $x = a$.
- if $x \neq y$, then $\{\{x\}, \{x, y\}\} = \{\{a\}, \{a, b\}\}$ iff $\{x\} = \{a\}$ and $\{x, y\} = \{a, b\}$. That is, iff $x = a$ and $y = b$. □

TODO oredered n-tuples

Definition 4.3. ***Union Axiom:*** For every set x there is a set z consisting of all elements of the elements of x.

$$\forall x \exists z \forall y (y \in z \leftrightarrow (\exists uu \in x \wedge y \in u))$$

We call z the union of x, notation: $\bigcup_x := z$

Definition 4.4. ***Power set Axiom:*** Let $x \subseteq y$ be the abbreviation for $\forall z (z \in x \to z \in y)$ The **Powerset Axiom** states, that for every set x there exists a set z consisting of all subsetes $y \subseteq x$ that are themselve sets.

$$\forall x \exists z \forall y (y \in z \leftrightarrow y \subseteq x)$$

Notation: $\mathcal{P}(x) := z$.

TODO class relations

Definition 4.5. ***Axiom of replacement / substitution:*** Let $\varphi(x, y, \underline{a})$ a $\mathcal{L}$-fla., w/ free variables among x, y and set-parameters $\underline{a}$. Suppose φ defines a class function on $\mathcal{U}$, than the followoing is an axiom:

$$\forall u \exists z \forall y (y \in z \leftrightarrow \exists x (x \in u \wedge \varphi(x, y, \underline{a})))$$

i.e. the image of a set under a class function is a set.

Definition 4.6. ***Axiom scheme of comprehension:*** TODO

List of definitions