Lecture notes
Einführung in die Logik 2024W

This is a summary of the material discussed in the lecture "Mathematische Logik". It is still a work in progress and there **may be mistakes** in this work. If you find any, feel free to let me know and I will correct them
The content of this script is intended for educational purposes only. It relies on the book [1] for which all rights belong to their respective owners and I do not claim ownership over this content. However the LaTeXcode I wrote is entirely my own work and is open source under the MIT-Licence. Dieses Skript ist noch nicht vollständig und wird regelmäßig aktualisiert.

Contents

List of Abbreviations

prop. - propositional 2
exp. - expression(s) 2
sent. - sentence(s) 3
seq. - sequence 3
TA - truth assignment 3
fla. - formula 3
TV - truth value 3
taut. - tautological 4
w/ - with 4
lp / rp - left / right parenthesis 5
i.e. - id est (that is) 9
MP - Modus Ponens 19
SUB - substitutable 20

CHAPTER 1

Propositional logic

Language

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: $A_1, A_2, A_3, \dots$ (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

expression

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

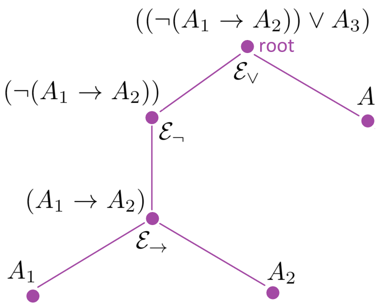
1. every prop. atom is a prop. sentence
2. if α, β are prop. sentences, then also $(\neg\alpha), (\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \rightarrow \beta), (\alpha \leftrightarrow \beta)$
3. nothing else (in particular $\varnothing$ is not a prop. fla.)

prop. fla.
prop. sentence

and call them **prop. sentences** or **prop. fla.** Equivalently stated every prop. sentence is built up by applying finitely many formula building operations on atoms and the prop. sent. returned from building operations.

$$\mathcal{E}_\neg, \mathcal{E}_\neg(\alpha) := (\neg\alpha) \text{ for any prop. fla. } \alpha \text{ and similarly for } \mathcal{E}_\wedge, \mathcal{E}_\vee \mathcal{E}_\rightarrow, \mathcal{E}_\leftrightarrow$$

This allows us to symbolize the **expression tree** (Here for example for $((\neg(A_1 \rightarrow A_2)) \vee A_3))$



We will return to these construction trees in 1.2, where we answer the question of what truth value a given prop. sentence might have.

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

construction sequence

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

Definition 1.4. Closedness of a set: Let S be a set. We say S is **closed** under an n-ary operational symbol f iff for all $s_1, s_2, \dots s_n \in S$ it holds $f(s_1, s_2, \dots s_n) \in S$

closure

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. $\langle \alpha_1, \dots \alpha_{n-1}, \alpha \rangle$ and $\alpha_1 \in S$. Let's assume that for $i \leq k < n$ each α_i 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 $\alpha_{k+1} \in S$

□

1.1 TRUTH ASSIGNMENTS

The interpretation of a prop. atom is either true or false, denoted by 0/1 or T/F or $\top/\bot$. A truth assigment is simply any map $\nu : S \mapsto \{0, 1\}$, where S is a map of propositional atoms. Our goal is going to be to extend any truth assigment v to a function $\overline{v} : \overline{S} \mapsto \{0, 1\}$, where $\overline{S}$ is the closure of S under the 5 fla. building operations.

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 $\nu : S \rightarrow \{0, 1\}$

Truth assigment
TA

We now want to extend ν to $\overline{\nu} : \overline{S} \rightarrow \{0, 1\}$, where $\overline{S}$ is the closure of S under the 5 fla. building operations such that for all propositional atoms $A \in S$ and propositional formulas α, β in $\overline{S}$

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. $\overline{\nu}(\alpha \rightarrow \beta) = \begin{cases} 1 & \text{iff } \overline{\nu}(\alpha) = 0 \text{ or } \overline{\nu}(\beta) = 1 \\ 0 & \text{otherwise} \end{cases}$
6. $\overline{\nu}(\alpha \leftrightarrow \beta) = \begin{cases} 1 & \text{iff } \overline{\nu}(\alpha) = \overline{\nu}(\beta) \\ 0 & \text{otherwise} \end{cases}$

We also want the extention to be unique, that is

Theorem 1.1. Unique readability: For all TA ν for a set S $\exists! \overline{\nu} : \overline{S} \rightarrow \{0, 1\}$ satisfying the above properties

We will prove this later

satisfy
satisfiable

Definition 1.6. Satisfaction: A TA ν satisfies a prop. sent. α if $\overline{\nu}(\alpha) = 1$ (that is, provided that everery atom of α is in the domain of ν). We call α satisfiable if there exists a TA that satisfies it.

taut. implication
$\models$

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 if for all TA that satisfy Σ, α is also satisfied and we write $\Sigma \models \alpha$. If $\Sigma = \{\beta\}$, we simply write $\beta \models \alpha$ If $\Sigma = \varnothing$ then α is called a **tautology** and we write $\models \alpha$ instead of $\varnothing \models \alpha$
α, β are called **tautologically equivalent** iff $\alpha \models \beta$ and $\beta \models \alpha$, we then write $\alpha \mathrel{=\!\mid\!\models} \beta$

Note: In other words, tautological implication $\Sigma \models \alpha$ means that you can not find a TA, that satisfy all members of Σ but not α. A tautology is satisfied by every TA. 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 $P = NP$

However we can find a way to reduce satisfiability of an infinite set Σ of prop. sent. to all finite subsets of Σ. There later will be a more elementary proof of the compactness theorem, this proof is not part of the exam.

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

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

then there is a TA satsfying every member of Σ.

Proof. using topology: We have our infinite set of prop. sent. which satisfies above condition. One way to look at TA is as a sequence of 0 and 1. Let $\mathcal{A} = \{A_0, A_1, \dots\}$ be the set of all prop. atoms. We are going to identify the truth assignments on $\mathcal{A}$ with elements in $\{0,1\}^{\mathcal{A}} := \{f : \mathcal{A} \to \{0,1\}\}$ (the set of all TAs) This is a topological space with product topology, on which the basic open sets (called cylinders) are: $U \subseteq \{0,1\}^{\mathcal{A}}$ is a cylinder, such that $p_n(U) = \{0,1\}$for all but finite many n, where p_n is the n-th projection. This means U is a cylinder if the truth values of its elements are at finitely many places fixed, and are arbitrary on everything else.

Note: These basic open sets are also closed. The open sets are unions of basic open sets. The idea is to use Tychonoffs Theorem which tells us that $\{0,1\}^{\mathcal{A}}$ is compact. i.e. the intersection of a family of closed subsets w/ the finite intersection property (FIP) is non-empty. Finite intersection property means the intersection of finitely many sets is non-empty.

For $\alpha \in \Sigma$ let $T_\alpha \subseteq \{0,1\}^{\mathcal{A}}$ be the set of TA that satisfy α. This T_α is a finite union of cylinders, hence T_α is closed. For the family $\{T_\alpha : \alpha \in \Sigma\}$ of closed sets we have (FIP). Tychonoff tells us, that $\bigcup_{\alpha \in \Sigma} T_\alpha \neq \varnothing$ so there is a TA satisfying Σ. ☐

For a list of tautologies: useful might be book p. 26-27

1.2 A PARSING ALGORITHM

To prove Theorem 1.1 We essentially need to show that we have enough parenthesis to make the reading of a prop. sent. unique. That is given a TA v there is at most one truth value we can assign to a prop. sent.

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.
$\mathcal{E}_\neg = (\neg\alpha) \dots$ ☐

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

Proof. Let $\alpha = \alpha_1\alpha_2\ldots\alpha_n$ be a prop. sent. By proper initial segment we understand $\beta = \alpha_1\ldots\alpha_i$ 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. Let PS = set of all prop. sent. and PF = set of prop. sent. s.t. no proper initial segment has # left parenthesis = # right parenthesis, we will prove that these sets are the same.
Let $\alpha \in PF$. By induction on the fla. building operations

- Atoms: since the empty sequence is not a 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 $\mathcal{E}_\neg$ preserves this property and under $\mathcal{E}_\wedge, \mathcal{E}_\vee, \mathcal{E}_\rightarrow, \mathcal{E}_\leftrightarrow$ this is also the case.

□

Parsing algorithm

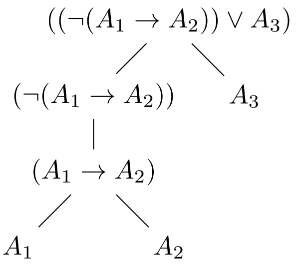
We now give a parsing algorithm procedure. For input we take some expression τ and the algorithm will determine if τ is a prop. sent. If so, it will generate a unique construction tree (in form of a rooted tree) for τ. (i.e. the construction tree gives us unique readability) That there is a unique way to perform the algorithm is implied by Lemma 1.2

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 "¬" 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 the five sentential connectives $\wedge, \vee, \rightarrow, \leftrightarrow$ 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 $:= \beta$, right $:= \beta'$) 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. *:* The parsing algorithm applied to $((\neg(A_1 \rightarrow A_2)) \vee A_3)$ returns the following construction tree.



Correctness of the parsing algorithm

- The algorithm always halts, because a child's label is shorter 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. cannot 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.

1.3 INDUCTION AND RECURSION

Generalization of induction principle:

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

$$f : U \times U \to U, \qquad g : U \to U$$

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

Definition 1.8. Closedness, Inductiveness: We say $\mathcal{S} \subseteq U$ is

closed
- **closed** under f and g iff for all $x, y \in \mathcal{S}$ it holds $f(x,y) \in \mathcal{S}$ and $g(x) \in \mathcal{S}$

inductive
- **inductive** if $B \subseteq \mathcal{S}$ and $\mathcal{S}$ is closed under $\mathcal{F}$

One way is from the top down

$$C^* := \bigcap_{\substack{B \subseteq S \\ \text{inductive}}} S$$

Another is from bottom up: We call $C_1 := B$,

$$C_i := C_{i-1} \cup \{f(x,y) : x, y \in C_{i-1}\} \cup \{g(x) : x \in C_{i-1}\}$$

and $C_* := \bigcup_{n \geq 1} C_n$ Exercise: show that $C^* = C_* =: C$.

Example 1.3. *:*

1. Let U be the set of all expressions, B the set of atoms and $\mathcal{F} = \{\mathcal{E}_\square : \square \in \{\neg, \wedge, \vee, \to, \leftrightarrow\}\}$ Then C would be the set of all propositional formulas.
2. Let U be $\mathbb{R}$, B the set containing 0 and $\mathcal{F} = \{S\}$, $S(x) = x + 1$ Then C would be the set of the natural numbers.

Induction principle

C generated from B by use of elements of $\mathcal{F}$ if $S \subseteq C$ such that $B \subseteq S$ and S is closed under all elements of $\mathcal{F}$, then $S = C$

Proof. $S \subseteq C$ is clear. S is inductive, so $C \subseteq S$. ☐

Question: under what conditions do we get "generalized unique readability?" The goal would be to define a function on C recursively i.e. to have rules for computing $\overline{h}(x)$ for $x \in B$ with some rules of computing $\overline{h}(f(x,y))$ and $\overline{h}(g(x))$ from $\overline{h}(x)$ and $\overline{h}(y)$.

Example 1.4. : Suppose that G is some additive group, generated from B (the set of generators), $h = B \to H$ where $(H, \cdot, ^{-1}, 1)$ a group. When is there an extention $\overline{h}$ of h s.th. $\overline{h} : G \to H$ is a grouphomomorphism.

- $\overline{h}(0) = 1$
- $\overline{h}(a + b) = \overline{h}(a) \cdot \overline{h}(b)$
- $\overline{h}(-a) = \overline{h}(a)^{-1}$

This is not always possible. **Note:** that it is possible if G is generated freely by the elements of B and the set of atoms is independent (one element of B cannot be generated in finitely many steps by other elements of B).

Definition 1.9. `Freely generated set:` C is **freely generated** from B by f, g if

freely generated

- C is generated from B by f, g
- $f|_{C^2}$ and $g|_C$ are such that
  1. $f|_{C^2}$ and $g|_C$ are one-to-one (injective)
  2. ran$(f|_{C^2})$ and ran$(g|_C)$ and B are p.w. disjoint

Theorem 1.3. `Recursion Theorem:` $C \subseteq U$ freely generated from B by f, g and V a set and $h : B \to V$, $F : V^2 \to V$, $G : V \to V$ Then $\exists! \overline{h} : C \to V$ s.that

- for all a in B it holds $\overline{h}(a) = h(a)$
- for all x, y in C it holds
  1. $\overline{h}(f(x, y)) = F(\overline{h}(x), \overline{h}(y))$
  2. $\overline{h}(g(x)) = G(\overline{h}(x))$

Note: if given conditions are satisfied then h extends uniquely to a homomorphism

$$(C, f, g) \to (V, F, G)$$

Before we proof the recursion theorem, we will show how unique readability easily follows from it.
Note: Recusion Theorem implies unique readability for propositional formulas. What we need to check is that the Assumptions of recursion theorem are satisfied.

Claim: The formula building operations are one-to-one.
proof of claim. $\mathcal{F}_\vee$ is one to one, suppose $(\alpha \vee \beta) = (\delta \vee \gamma)$ then $\alpha \vee \beta) = \delta \vee \gamma)$ And α, δ are prop. formulas, so they equal to each other (else one is an initial segment of the other, hence not a prop. fla.) By the same argument we get β is equal to γ. ☒

Claim: Disjointment of ranges
proof of claim.
- if $(\alpha \vee \beta) = A$ then A starts with (which can not be the case
- if $(\alpha \vee \beta) = (\gamma \to \delta)$ then by the same argument α is γ but $\vee$ and $\to$ are diffrent
- if $(\alpha \vee \beta) = (\neg\gamma)$, then $\alpha \vee \beta) = \neg\gamma)$, so α would start with a $\neq$, -no

For all other connectives the proof is similar. ☒

Proof of the Rec Thm.

$v : C \rightharpoonup V$ is called acceptable if $\forall x, y \in C$

acceptable

1. if $x \in B \cap dom(v)$ then $v(x) = h(x)$
2. if $f(x, y) \in dom(v)$ then $x, y \in dom(v)$ and similarly for g
   - $v(f(x, y)) = F(v(x), v(y))$
   - $v(g(x)) = G(v(x))$

And when $U = \{\Gamma_v : v\text{acceptable}\}$, we define $\overline{h} :=$ function w/ graph $\bigcup \Gamma_v$

Claim 1: $\overline{h}$ is a function.
proof of claim.

$$S := \{x \in C : \exists \text{at most one } y \text{ with } (x,y) \in \bigcup \mathit{\Gamma}_v\}$$

We want $S = C$, we have $S \subseteq C$, it is enough to show that S is inductive.

- $x \in B \cap \operatorname{dom}(v)$ for some v acceptable.
  then $v(x) = h(x)$ by 1. also $x \notin \operatorname{ran}(f|_{C^2})$ and $x \notin \operatorname{ran}(g|_C)$
- $x, y \in \mathcal{S}$ We want $f(x,y), g(x) \in S$
  there are v_1, v_2 acceptable s.t. $f(x,y) \in \operatorname{dom}(v_1) \cap \operatorname{dom}(v_2)$

⊠

Claim 2: $\overline{h}$ is acceptable.
proof of claim. $\overline{h} : C \rightharpoonup V$ by definition. if $x \in B \cap \operatorname{dom} \overline{h}$ then there is a v acceptable, s.t. $x \in dom(v)$ then $\overline{h}(x) = v(x) = h(x)$ if $f(x,y) \in \operatorname{dom} \overline{h}$ then $f(x,y) \in \operatorname{dom}(v)$ form some v acceptable. Hence $x, y \in \operatorname{dom}(v)$ and therefore $x, y \in \operatorname{dom}(\overline{h})$ and we have

$$\overline{h}(f(x,y)) = v(f(x,y)) = F(v(x), v(y)) = F(\overline{h}(x), \overline{h}(y))$$

⊠

Claim 3: The domain of $\overline{h}$ equals C.
proof of claim. it is enough to show that the domain of $\overline{h}$ is inductive. $B \subseteq \operatorname{dom}(\overline{h})$ bc. $B \subseteq \operatorname{dom}(h)$ where h is acceptable. Now we need to show closure under f, g. suppose $x', y' \in \operatorname{dom}(\overline{h})$ then $x' \in \operatorname{dom}(v_1)$ for some acceptable v_i lets assume $f(x', y') \notin \operatorname{dom}(\overline{h})$ then we extend $\overline{h}$ to a function with the same graph as $\overline{h}$. Then $\mathit{\Gamma} \cup \{(f(x', y'), F(\overline{x'}, \overline{y'}))\}$ is the graph of an acceptable function. ⊠

Claim 4:$\overline{h}$ is uniquely constructed
proof of claim. Suppose both $\overline{h}, \overline{\overline{h}}$ work, we schow that $S = \{x \in C : \overline{h}(x) = \overline{\overline{h}}(x)\}$ is the whole set C. it is enough to show that S is inductive. Let $x \in B$ then $\overline{h}(x) = h(x) = \overline{\overline{h}}(x)$. Then for $x, y \in S$

$$\overline{h}(f(x,y)) = F(\overline{h}(x), \overline{h}(y)) = F(\overline{\overline{h}}(x), \overline{\overline{h}}(y)) = \overline{\overline{h}}(f(x,y))$$

$$\overline{h}(g(x)) = G(\overline{h}(x)) = G(\overline{\overline{h}}(x)) = \overline{\overline{h}}(g(x))$$

and $f(x,y), g(x) \in S$, therefore S is inductive. ⊠

1.4 SENTENTIAL CONNECTIVES

tautological equivalence
$\sim$
$\models\dashv$
$\dashv\models$

Definition 1.10. `Tautological equivalence relation:` For α, β prop. sent. we define $\alpha \sim \beta$ iff $\alpha \dashv\models \beta$ (alternative notation: $\models\dashv$). This defines an equivalent relation.

Example 1.5. : $A \to B \dashv\models \neg A \vee B$

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 $(A_1 \vee \neg A_2)$ then $B_\alpha : \{0,1\}^2 \to \{0,1\}$ and asign its values corresponding a truth value of α. That is for any TA $v : \{A_1, A_2\} \to \{0,1\}$ we define $B_\alpha(v(A_1), v(A_2)) = \overline{v}(\alpha)$

Theorem 1.4. : 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. $\alpha \dashv\models \beta$ 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 $B_\alpha(x) = 1$

n-ary boolean func.

Theorem 1.5. `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 $A_1 \wedge \neg A_1$.
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 $1 \leq i \leq k$
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).

DNF
Disjunctive normal form

Corollary 1.5. Every prop. sent. is tautologically equivalent to a sentence in DNF

Corollary 1.5. $\{\neg, \wedge, \vee\}$ is a complete set of logical connectives, i.e. every prop. sent. is tautologically equivalent to a sentence built up from atoms and $\neg, \wedge, \vee$.

complete

Theorem 1.6. : 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 $\alpha \wedge \beta \dashv\models \neg(\neg\alpha \vee \neg\beta)$ We prove this by induction on the number of disjuctions of a prop. sent. α in DNF. Suppose the statement is true for $k \leq n$. For $n+1$ and $\alpha = \bigvee_{j=1}^{n+1} \gamma_j$ there exists an $\alpha' \dashv\models \bigvee_{j=1}^{n} \gamma_j$ and

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

□

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

Example 1.6. $\{\rightarrow, \wedge\}$ `is not complete.:` Let $\alpha \in PS$ built up from only $\rightarrow, \wedge$ 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 $\{\rightarrow, \wedge\}$ is not complete bc. $\neg A$ is not tautological equivalent to a sent. built up from $\rightarrow, \wedge$

Proof. Let $C := \{\alpha \in PS$ built up from $\rightarrow, \wedge$ 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 $\rightarrow, \wedge$ 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) $A_1 \wedge \dots \wedge A_n \models \alpha \rightarrow \beta$
  (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: $\{\wedge, \vee, \rightarrow, \leftrightarrow\}$ 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 $\{\neg, \rightarrow\}$ and $\{\rightarrow, \perp\}$ are both complete, it holds $\neg A \dashv\models A \rightarrow \perp$

Definition 1.11. **`Satisfiability:`** A set of prop. sent. Σ is called **satisfiable** if there exists a TA that satisfies every member of Σ.

satisfiable

1.5 COMPACTNESS THEOREM

Theorem 1.7. Compactness Theorem: Σ is satisfiable iff every finite subset $\Sigma_0 \subseteq \Sigma$ is satisfiable. (i.e. Σ is finitely satisfiable)

finitely satisfiable

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 $\alpha_1, \alpha_2, \dots$ 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 $A_1, A_2, \dots$ 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) $\{A_1, A_2 \dots\} \subseteq S$ 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 $\epsilon_\rightarrow$: Let $\varphi_1, \varphi_2 \in S$ similiarly

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

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

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

Corollary 1.7. 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 1.7 and Corollary 1.7 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: for every $n > 0$ we have a set of n-ary predicates P
   - constant symbols: Some set of constants (could also be $\varnothing$)
   - function symbols: for every $n > 0$ we have a set of n-ary function symbols

logical symbols

parameters

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
- In the book [1] they assume that some n-place predicate symbol is present for some n.

Example 2.1. *:*

- $\mathcal{L}_{\text{set}} = \{\in\}$, $=$ included and the binary predicate symbol $\in$ "element in"
- $\mathcal{L}_{\text{arith}} = \{<, 0, S, E, +, \cdot\}$

  $=$ included

  $<$ is a binary rel. symbol

  0 is a constant

  S is a unary function symbol

  E exponentiation function symbol

  $+, \cdot$ binary function symbols

- $\mathcal{L}_{\text{ring}} = \{=, +, \cdot, -, 0, 1\}$

  $=$ included

  $0, 1$ are constants

  $-$ is a unary function symbol (additive inverse)

  $+, \cdot$ binary function symbols

2.1 FORMULAS

Definition 2.2. Expression: An **expression** is any finite sequence of symbols. There exist two kinds of expressions that makes sense "grammatically"

Terms:
- points to an object
- they are built up from variables and constants using function symbols

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 =, if included

term

Definition 2.3. Term Building Operations: For every $n > 0$ and for every n-place function symbol f let $\mathcal{F}_f$ be an n-place term building operation, that is $\mathcal{F}_f(t_1, \dots t_n) := ft_1, \dots t_n$ (polish notation for $f(t_1, \dots t_n)$). The Set of terms we then define as the set of expressions that are built up from variables and constants by applying the term 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

$$= t_1 t_2 \text{ or } Pt_1, \dots t_n, \text{ where } t_1, \dots t_n \text{ are terms and } P \text{ is an } n\text{-ary predicate symbol}$$

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.

formula

Definition 2.5. Formulas: We define $\varepsilon_\neg$, $\varepsilon_\to, Q_i$ to be the fla. building operations, defined as follows $\varepsilon_\neg(\alpha) := (\neg\alpha)$, $\varepsilon_\to := (\alpha \to \beta)$ 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.

Example 2.4. : *cont.* $\forall v_1 (= Sv_1 0)$ is a formula we get by applying Q_1 on the atomic formula $= Sv_1 0$.

Free variables

∃ quantifier

bounded variable

Example 2.5. : We introduce the $\exists$ quantifier by defining $\exists y \alpha$ means $\neg\forall y \neg\alpha$.
"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" $x \neq 0 \to \exists y S(y) = x$. x occurs bounded in the first formula, for the latter x occures free in the fla.

If you have an expression without free variables, it is either true or false, on the other hand if a variable occurs free in a formula, the truth value of it depends on the variable itself.

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 $\varphi = (\neg\alpha)$ then x occurs free in φ iff x occurs free in α
3. If $\varphi = (\alpha \to \beta)$ then x occurs free in φ iff x occurs free in α or β
4. If $\varphi = \forall v_i \alpha$ then x occurs free in φ iff x occurs free in α and $x \neg v_i$

sentence

A formula α is called a sentence, if no variable occurs free in α

Note: The above definition makes sense thanks to the recursion theorem. define the function h on the set of atoms: $h(\alpha) =$ the set of var occ in fla α, which is the set of all variables v_i that occur free in α. we now want to extend h to $\overline{h}$, which is the set of all formulas.

- $\overline{h}(\neg\alpha) = \overline{h}(\alpha)$
- $\overline{h}(\alpha \to \beta) = \overline{h}(\alpha) \cup \overline{h}(\beta)$
- $\overline{h}(Q_i(\alpha)) = \overline{h}(\alpha)\backslash\{v_i\}$

We say x occurs free in α iff $x \in \overline{h}(\alpha)$.
Note: We will now use $\neg, \wedge, \vee, \to, \leftrightarrow, \exists v_i$ (all can be expressed in terms of $\neg, \to, Q_i$.) We will sometimes drop the $(,)$ and not always be using polish notation.

2.2 SEMANTICS OF FIRST ORDER LOGIC

The equivalent scheme to our TA in predicate logic. The meaning of formulas is given by *structures*, which also determine the scope of the quantifier $\forall$, the meaning of all parameters.

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 parameters 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 interpretation
- 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

$$f^{\mathcal{A}} : A^n \to A$$

Note: $A \neq \varnothing$, and all functions $f^{\mathcal{A}}$ are total.

Example 2.6. : Let $\mathcal{L} = \{\in\}$ where $\in$ is a binary relation " An example of an $\mathcal{L}$ structure is $(\mathbb{N}, \in^{\mathbb{N}})$ where $\in^{\mathbb{N}} = \{(x, y) \in \mathbb{N}^2 : x < y\}$

Definition 2.8. Assigment: Let φ be a $\mathcal{L}$-fla. and $\mathcal{A}$ a $\mathcal{L}$-structure. Let V be the set of all variables in $\mathcal{L}$ and $s : V \to A$ an assignment. We define the extention $\overline{s}$ of s to the set of all $\mathcal{L}$-terms by

assigment

- if $x \in V$ then $\overline{s}(x) := s(x)$
- for $c \in \mathcal{L}$ a constant symbol, then $\overline{s}(c) := c^{\mathcal{A}}$
- for $t_1, \dots t_n$ $\mathcal{L}$-terms and $f \in \mathcal{L}$ an n-ary function symbol, then

$$\overline{s}(ft_1 \dots t_n) := f^{\mathcal{A}}(\overline{s}(t_1), \dots \overline{s}(t_n))$$

Note: in the previous definition point 3. for $n = 1$ yields a commutative diagram.

Theorem 2.1. : For any given assignment s there exists a unique extention $\overline{s}$ as in the previous definition.

Proof. will follow from recursion theorem and unique decomposition of terms. ☐

Definition of truth

$\models_{\mathcal{A}}$ **Definition 2.9. Satisfy:** We define '$\mathcal{A}$ satisfies φ with s' and write $\mathcal{A} \models \varphi\,[s]$ or $\models_{\mathcal{A}} \varphi\,[s]$ inductively over the complexity of the formula φ

1. if φ is atomic:
   - $\mathcal{A} \models = t_1, t_2\,[s]$ iff $\overline{s}(t_1) = \overline{s}(t_2)$
   - $\mathcal{A} \models Pt_1, \dots t_n\,[s]$ iff $(\overline{s}(t_1), \dots \overline{s}(t_2)) \in P^{\mathcal{A}}$
2. suppose $\mathcal{A} \models \varphi\,[s]$ and $\mathcal{A} \models \psi\,[s]$ are defined, then
   - $\mathcal{A} \models \neg\varphi\,[s]$ iff $\mathcal{A} \not\models \varphi\,[s]$
   - $\mathcal{A} \models \varphi \to \psi\,[s]$ iff $\mathcal{A} \models \psi\,[s]$ or $\mathcal{A} \not\models \varphi\,[s]$
   - $\mathcal{A} \models \forall x\varphi\,[s]$ iff for all $a \in A$ $\mathcal{A} \models \varphi[s(x|a)]$ where

$$s(x|a)(v) = \begin{cases} s(v) \text{ if } v \neq x \\ a \text{ if } v = x \end{cases}$$

Example 2.7. : $\mathcal{L} = \{\forall, \leq, S, 0\}$ a $\mathcal{L}$-structure then could be $\mathcal{N} = (\mathbb{N}, \leq^{\mathcal{N}}, S^{\mathcal{N}}, 0^{\mathcal{N}})$ together with an assignment $s : v_n \mapsto n - 1$ then:

- $s(v_1) = 0$
- $\overline{s}(0) = 0^{\mathcal{N}}$ (a constant is always mapped to its realisation, the interpretation of constant 0 in the structure $\mathcal{N}$)
- $\overline{s}(Sv_1) = S^{\mathcal{N}}(\overline{s}(v_1)) = S^{\mathcal{N}}(0) = 1$

- $\mathcal{N} \models \forall v_1(S(v_1) \neq v_1)\,[s]$
  iff for all $a \in \mathbb{N}$ we have that $\mathcal{N} \models (S(v_1) \neq v_1)[s(v_1|a)]$
  iff ...
  iff for all $a \in \mathbb{N}$ we have $S^{\mathcal{A}}(a) \neq a$, which is true in our structure of the natural numbers.
- Is it true in $\mathcal{N}$ that $\mathcal{N} \models S(0) \leq S(v_1)\,[s]$? Yes because

$$\begin{aligned} &\mathcal{N} \models S(0) \leq S(v_1)\,[s] \\ &\text{iff } 1 \leq 1 \end{aligned}$$

Note: To know wheter $\mathcal{A} \models \varphi\,[s]$ it suffices to know where s maps the variables that are free in φ

Theorem 2.2. : Suppose $s_1, s_2 : V \to A$ agree on all variables that occur free in φ then

$$\mathcal{A} \models \varphi\,[s_1] \text{ iff } \mathcal{A} \models \varphi\,[s_2]$$

Proof. By complexity of φ

1. if φ is $Pt_1, \dots t_n$ note: any var that occur in φ occur free in φ, so s_1, s_2 agree on all variables that occur in the terms $t_1, \dots t_n$.
   So we Claim: for t a term, s_1, s_2 assignments that agree on all variables of t then $\overline{s}_1(t) = \overline{s}_2(t)$
   proof of claim. By complexity of t
   - $t = v_m$ then $\overline{s}_1(t) = s_1(v_m) = s_2(v_m) = \overline{s}_2(t)$
   - $t = c$ then $\overline{s}_1(t) = c^{\mathcal{A}} = \overline{s}_2(t)$
   - $t = ft_1 \dots t_n$ inductively, assume $\overline{s}_1(t_i) = \overline{s}_2(t_i)$ for all $1 \leq i \leq n$ then TODO

   ☒
2. if φ is $= t_1, t_2$ is similar
3. if φ is $\neg\alpha$ then $\mathcal{A} \models \neg\alpha\,[s_1]$ iff $\mathcal{A} \vDash \alpha\,[s_1]$iff $\mathcal{A} \vDash \alpha\,[s_2]$ iff $\mathcal{A} \models \neg\alpha\,[s_1]$

4. if φ is $\alpha \to \beta$ then $\mathcal{A} \models \alpha \to \beta\,[s_1]$ iff .. or .. iff for s2 iff ... or ..
5. if φ is $\forall x\alpha$ then the assumption is that s_1, s_2 .. the free variables of α are the free variables of φ exept for x. but because $s_1(x|a) = s_2(x|a)$ they both agree on all free variables of α.

$$\begin{aligned}\mathcal{A} \models \forall x\varphi\,[s_1] &\,\text{iff forall}\, a \in A \mathcal{A} \models \varphi\,[s_1(x|a)]\\ &\,\text{iff forall}\, a \in A \mathcal{A} \models \varphi\,[s_2(x|a)]\\ &\,\text{iff}\, \mathcal{A} \models \forall x\varphi\,[s_2]\end{aligned}$$

□

Notation: $\mathcal{A} \models \varphi$TODO means that all free variables of φ are among $v_1, \dots v_n$ and $\mathcal{A} \models \varphi\,[s]$ whenever $s(v_i) = a_i$ for all $1 \leq i \leq n$.

Corollary 2.2. If σ is a sentence then $\mathcal{A} \models \sigma\,[s]$ for all $s : V \to A$ or $\mathcal{A} \vDash \sigma\,[s]$ for all $s : V \to A$.

Notation: $\mathcal{A} \models \sigma$ ”σ is true in $\mathcal{A}$, $\mathcal{A}$ is a model of σ or σ holds in $\mathcal{A}$.

Note: If σ is a sentence then we can not have $\mathcal{A} \models \sigma$ and $\mathcal{A} \vDash \sigma$ because $A \neq \varnothing$.

Definition 2.10. **`Model`:** $\mathcal{A}$ is a model of a set of sentences Σ iff for every sentence $\sigma \in \Sigma$ it holds $\mathcal{A} \models \sigma$

Example 2.8. *:* $\mathcal{L} = \{0, 1, +, -, \cdot\}$ A realisation could be $\mathcal{R} = (\mathbb{R}, 0, 1, +, -, \cdot)$ or $\mathcal{C} = (\mathbb{C}, 0, 1, +, -, \cdot)$ then the sentence $\sigma :\quad \exists x(x \cdot x = -1)$ then $\mathcal{R} \vDash \sigma$ but $\mathcal{C} \models \sigma$

Note: $\wedge, \vee, \leftrightarrow, \exists$ work as expected. That is $\mathcal{A} \models (\alpha \wedge \beta)\,[s]$ iff $\mathcal{A} \models \alpha\,[s]$ and $\mathcal{A} \models \beta\,[s]$ $\mathcal{A} \models (\alpha \vee \beta)\,[s]$ iff $\mathcal{A} \models \alpha\,[s]$ or $\mathcal{A} \models \beta\,[s]$ $\mathcal{A} \models \exists x\alpha\,[s]$ iff $\mathcal{A} \models \neg\forall x\neg\alpha\,[s]$
iff $\mathcal{A} \vDash \forall x\neg\alpha\,[s]$
iff it is not true that forall $a \in A$ $\mathcal{A} \models \neg\alpha[s(x|a)]$
iff there is $a \in A$ such that $\mathcal{A} \models \alpha[s(x|a)]$

2.3 LOGICAL IMPLICATION

Let Γ be a set of $\mathcal{L}$-formulas, φ a $\mathcal{L}$-formula.

Definition 2.11. **`Logical implication`:** $\Gamma \models \varphi$ ”Γ logically implies φ” if for every L-structure $\mathcal{A}$ and for every $s : V \to A$

if $\mathcal{A} \models \gamma\,[s]$for every $\gamma \in \Gamma$ then $\mathcal{A} \models \varphi\,[s]$

Definition 2.12. **`Logical equivalence`:** φ, ψ are called logically equivalent if $\varphi \models \psi$ and $\psi \models \varphi$.

Definition 2.13. **`Valid`:** φ is called valid iff $\models \varphi$ i.e. $\varnothing \models \varphi$ i.e. for every $\mathcal{L}$-structure $\mathcal{A}$ and every $s : V \to A$ it is $\mathcal{A} \models \varphi\,[s]$

Example 2.9. *:*

1. $\forall x_1 Px_1 \models Px_2$
   Suppose $\mathcal{A} \models \forall x_1 Px_1\,[s]$. then for all $a \in A$ it is $\mathcal{A} \models Px_1[s(x_1|a)]$ in particular, $a \in P^{\mathcal{A}}$ for $a = s(x_2)$
2. $\forall Px_2 \vDash \forall x_1 Px_1$
   We need a counterexample to $\forall Px_2 \models \forall x_1 Px_1$. Let $A = \{a_1, a_2\}$ $s(x_2) = a_1$ and $P^{\mathcal{A}} = \{a_1\}$ then $\mathcal{A} \models Px_2\,[s]$.
3. Is the following valid? $\models \exists x(Px \to \forall y Py)$ yes
4. $\Gamma, \alpha \models \varphi$ iff $\Gamma \models \alpha \to \varphi$. (on next problem set, quite impoirtant)

2.4 DEFINABILITY IN A STRUCTURE

Definition 2.14. definability in a structure: We say that a general n-ary relation P on A (we will just call it P, it does not have to be in the language) is definable in $\mathcal{A}$, if there is a $\mathcal{L}$-formula φ with free variables among $\{v_1, \ldots, v_n\}$ such that

$$P = \{(a_1, \ldots a_n) : \mathcal{A} \models \varphi\, [\![a_1, \ldots a_n]\!]\}$$

We also say that φ defines P in the structure $\mathcal{A}$.

Example 2.10. *:*

1. $x = x$ would define the entire universe.
2. $\neg x = x$ would define the empty set.

Example 2.11. *:*

1. TODO
2. $\mathcal{R} = (\mathbb{R}, 0, 1, +, -, \cdot)$ Q: is $[0, \infty)$ definable in $\mathcal{R}$ Yes because $\exists y(y \cdot y = x)$ Indeed we can even define the $\leq$ relation on $\mathbb{R}^2$ by $x \leq z :\Leftrightarrow \exists y(x + y \cdot y = z)$

Definition 2.15. definability of classes of structures: Let Σ be a set of sentences. τ a sentence. We will say that the class of models of Σ is the class $\operatorname{Mod}\Sigma = \{\mathcal{A} : \mathcal{A} \models \Sigma\}$. Let K be a class of structures. We are going to call K an elementary class (EC) if there is a single sentence τ such that $K = \operatorname{Mod}\tau$. K is called an elementary class in the wider sence (EC_Δ) if there is a set of sentences Σ such that $K = \operatorname{Mod}\Sigma$

Example 2.12. *:* $\mathcal{L} = \{0, 1, +, \cdot\}$ τ is a sentence that expresses the field axioms (the unary inverse functions are not in our language but are definable.) $\operatorname{Mod}\tau$ is the class of all the fields, which is EC. the class of all fields of characteristic 0. Let $\sigma_p : \neg(1 + \cdots + 1 = 0)$ then $\Sigma = \{\tau\} \cup \{\sigma_p : p \in \mathbb{P}\}$ yields $\operatorname{Mod}\Sigma$ is the class of fields with characteristic 0, therefore EC_Δ, we will later see that it is not EC.

Example 2.13. *:* Let E be a binary relation, $\mathcal{L} = \{E\}$ then a graph is a realisation $\mathcal{G} = (V, E^{\mathcal{G}})$ such that $v \neq \varnothing$, $E^{\mathcal{G}}$ is irreflexive and symmetric. By definition the universe is not empty, we still have to check irreflexive and symmetric.

- irreflexive: $\forall x(\neg xEx)$
- symmetric: $\forall x \forall y(xEy \to yEx)$

We take τ to be $\forall x \forall y((\neg xEx) \wedge (xEy \to yEx))$ Then $\operatorname{Mod}\tau$ is the class of all graphs and is EC Note: the class of all finite graphs is neither EC nor EC_Δ. proof later.

We want to have some notion that tells us when two graphs are the same or at least similar.

2.5 HOMOMORPHISMS OF STRUCTURES

Definition 2.16. Homomorphism: Suppose that $\mathcal{A}, \mathcal{B}$ are two $\mathcal{L}$-structures. then a Homomorphism of $\mathcal{A}$ into $\mathcal{B}$ is a map $h : A \to B$ that satisfy the below conditions

- for every n-ary predicate $P \in \mathcal{L}$ it is $(a_1, \ldots a_n) \in P^{\mathcal{A}}$ iff $(h(a_1), \ldots h(a_n)) \in P^{\mathcal{B}}$ (this def. a strong Homomorphism, other textbooks maybe only reqire $\to$ direction)
- for every n-ary function $f \in \mathcal{L}$ and for all $\underline{a} = (a_1, \ldots a_n) \in A^n$ it holds $h(f^{\mathcal{A}}(\underline{a})) = f^{\mathcal{B}}(h(a_1), \ldots h(a_n))$
- for every constant symbol $c \in \mathcal{L}$ it is $h(c^{\mathcal{A}}) = c^{\mathcal{B}}$ (could also skip this if we consider constants as 0-ary functions)

Note: Intuativly a Homomorphism of $\mathcal{A}$ into $\mathcal{B}$ is a map $A \to B$ that preserve all funciton and relation symbols in some sense, (imp: not the definable relations)

Definition 2.17. Isomorphism:

- $h : A \to B$ is called isomorphism of $\mathcal{A}$ into $\mathcal{B}$ if h is a Homomorphism and injective (in other textbooks: an isomorphic embedding of $\mathcal{A}$ into $\mathcal{B}$)
- $h : A \to B$ is called isomorphism of $\mathcal{A}$ onto $\mathcal{B}$ if h is a Homomorphism and bijective $A \to B$
- $\mathcal{A}$ and $\mathcal{B}$ are called isomorphic if there is an isomorphism of $\mathcal{A}$ onto $\mathcal{B}$

isomorphic

Note:

Example 2.14. *:* $\mathcal{L} = \{+, \cdot\}$ $\mathcal{N} = (\mathbb{N}, +^{\mathbb{N}}, \cdot^{\mathbb{N}})$ and $\mathcal{B} = (B, +^{\mathcal{B}}, \cdot^{\mathcal{B}})$ where $B = \{0, 1\}$ and $\begin{array}{c|cc} +^{\mathcal{B}} & e & 0 \\ \hline e & e & 0 \\ 0 & 0 & e \end{array}$ $\begin{array}{c|cc} \cdot^{\mathcal{B}} & e & 0 \\ \hline e & e & e \\ 0 & e & 0 \end{array}$ let $h : \mathbb{N} \to B$ a Homomorphism? $h(n) = \begin{cases} e & \text{if } n \text{ is even} \\ 0 & \text{else} \end{cases}$ need at first that $h(m + n) = h(m) +^{\mathcal{B}} h(n)$ and $h(m \cdot n) = h(m) \cdot^{\mathcal{B}} h(n)$. it is indeed a Homomorphism.

Definition 2.18. Substructure: Suppose we have two $\mathcal{L}$ structures and $A \subseteq B$ hen $\mathcal{A}$ is a substructure of $\mathcal{B}$ (notation: $\mathcal{A} \subseteq \mathcal{B}$ or we might say $\mathcal{B}$ is an extention of $\mathcal{A}$) if

- for every n-ary relation $P^{\mathcal{A}} = P^{\mathcal{B}}|_A$
- for every n-ary function $f^{\mathcal{A}} = f^{\mathcal{B}}|_A$
- for every constant symbol c in $\mathcal{L}$ it is $c^{\mathcal{A}} = c^{\mathcal{B}}$

Example 2.15. *:* $\mathcal{L} = \{\leq\}$ then $\mathcal{N} = (\mathbb{N}, \leq)$ and $\mathcal{P} = (\mathbb{N}^+, \leq^{\mathcal{P}})$ where $\leq^{\mathcal{P}}$ is the restriction of $\leq$ to the positive natual numbers. $\mathcal{P} \subseteq \mathcal{N}$ and there exists a isomorphic embedding $id : \mathbb{N}^+ \to \mathbb{N}$ from $\mathcal{P}$ into $\mathcal{N}$ They are even isomorphic ($h : \mathbb{N} \to \mathbb{N}^+, h(n) = n+1$) so in fact $\mathcal{P} \cong \mathcal{N}$.

Example 2.16. *:* $(\mathbb{Q}, +) \subseteq (\mathbb{C}, +)$

Note: If $\mathcal{A} \subseteq \mathcal{B}$ then in particular $\mathcal{A}$ is closed under all constant and functions in $\mathcal{B}$ So suppose that $\mathcal{B}$ is a substructure and $A \subseteq B$ and $A \neq \varnothing$ and A is closed under $f^{\mathcal{B}}$, $c^{\mathcal{B}}$ Can then A be made into a substructure $\mathcal{A}$ of $\mathcal{B}$. $f^{\mathcal{A}}$ would be the restriction of $f^{\mathcal{B}}$ to A^n, constants $c^{\mathcal{A}} = c^{\mathcal{B}}$ and if $P \in \mathcal{L}$ is an n-ary predicate then $P^{\mathcal{A}}$ should be $P^{\mathcal{B}} \cap A^n$. If $\mathcal{L}$ has no const. or fuction symbols then any subset can be made into a substructure of a structure on $\mathcal{L}$.

Our next question will be: what is the relation of the above notions with truth and satisfiability The answer will be given by the so called Homomorphism theorem.

Theorem 2.3. Homomorphism theorem: h homomorphism of $\mathcal{A}$ into $\mathcal{B}$, $s : V \to A$ then

1. for all terms t it is $h(\overline{s}(t)) = \overline{(h \circ s)}(t)$
2. φ a fla. that is quantifier free and does not include $=$ then $\mathcal{A} \models \varphi\,[s]$ iff $\mathcal{B} \models \varphi\,[h \circ s]$
3. if h is additionally injective then we can drop the requirement " no =".
4. if h is homomorphism of $\mathcal{A}$ onto $\mathcal{B}$ then we can drop the requirement "q.f." in (b)

Proof. 1. problem set

2.
   - $\varphi : Pt$ then $\mathcal{A} \models Pt\,[s]$ iff $\overline{s}(t) \in P^{\mathcal{A}}$ iff $h(\overline{s}(t)) \in P^{\mathcal{B}}$ iff $\overline{(h \circ s)}(t) \in P^{\mathcal{B}}$ iff $\mathcal{B} \models Pt\,[h \circ s]$
   - $\varphi : \neg\psi$ $\mathcal{A} \models \neg\psi\,[s]$ iff $\mathcal{A} \nvDash \psi\,[s]$ iff $\mathcal{A} \nvDash \psi\,[s]$ iff
   - $\varphi : \psi \to \alpha$
3. $\mathcal{A} \models = t_1 t_2\,[s]$ iff $\overline{s}(t_1) = \overline{s}(t_2)$ iff $h(\overline{s}(t_1)) = h(\overline{s}(t_2))$ iff (by (a))$\overline{(h \circ s)}(t_1) = \overline{(h \circ s)}(t_2)$ iff $\mathcal{B} \models = t_1 t_2\,[h \circ s]$

4. φ $\forall s : V \to A$ $\mathcal{A} \models \varphi\,[s]$ iff $\mathcal{B} \models \varphi\,[h \circ s]$, want $\mathcal{A} \models \forall x \varphi\,[s]$ iff $\mathcal{B} \models \forall x \varphi\,[h \circ s]$ 1. $\mathcal{B} \models \forall x \varphi\,[(h \circ s)]$ iff for all $s : V \to A$, $a \in A$ (req. surjectivity) it is $\mathcal{B} \models \varphi\,[(h \circ s)(x|h(a))]$ iff $\mathcal{B} \models \varphi\,[h \circ (s(x|a))]$ iff (inductive assumption)$\mathcal{A} \models \varphi\,[s(x|a)]$ because a was arbitrary it is $\mathcal{A} \models \forall x \varphi\,[s]$ 2. Suppose $\mathcal{B} \vDash \forall x \varphi\,[(h \circ s)]$ then there exists a $b \in B$ such that $\mathcal{B} \models \neg\varphi\,[(h \circ s)(x|b)]$ by surjectivity we can find $a \in A$ such that $h(a) = b$ and it is $\mathcal{B} \models \neg\varphi\,[(h \circ s)(x|h(a))]$ By the inductive assumption $\mathcal{A} \models \neg\varphi\,[s(x|a)]$ and $\mathcal{A} \vDash \forall x \varphi\,[s]$ ☐

Note: $\mathcal{A} \cong \mathcal{B}$ then $\mathcal{A}$ and $\mathcal{B}$ satisfy exactly the same sentences.

Definition 2.19. elementarily equivalent: $\mathcal{A}$ and $\mathcal{B}$ are called elementarily equivalent ($\mathcal{A} \equiv \mathcal{B}$) if $\mathcal{A}$ and $\mathcal{B}$ satisfy the same sentences.

Note: If $\mathcal{A} \cong \mathcal{B}$ implies $\mathcal{A} \equiv \mathcal{B}$ The converse is not true. For instance DLO (dence linear order) w/o endpoints is complete, so two structures on DLO are equivalent $(\mathbb{Q}, <) \equiv (\mathbb{R}, <)$ but they are not isomorphic because the universes have diffrent cardinality.

Example 2.17. : $\mathcal{N} = (\mathbb{N}, \leq)$ and $\mathcal{P} = (\mathbb{N}^{>0}, \leq)$ $h : n \mapsto n - 1 : \mathcal{P} \to \mathcal{N}$ isom. so in part $\mathcal{N} \equiv \mathcal{P}$. but $id : \mathcal{P} \to \mathcal{N}$ is only isom embedding, so for example $\forall y(x \neq yx \leq y)$ $\mathcal{P} \models \alpha\,[1]$ but $\mathcal{N} \nvDash \alpha\,[1]$ but $\mathcal{N} \models \alpha\,[h(1)]$

Definition 2.20. Automorphism: An automorphism is an isomorphism of the form $h : A \to A$ from $\mathcal{A}$ onto $\mathcal{A}$

Note: Every structure has a trivial automorphism $id : A \to A$

Definition 2.21. Rigid: If the only automorphism on $\mathcal{A}$ is the trivial automorphism, then $\mathcal{A}$ is called rigid.

Example 2.18. : If every element is definable then the structure is rigid. For example $(\mathbb{N}, 0, S)$ and $(\mathbb{N}, <)$ every element is definable, therefore the structures are rigid.

Corollary 2.3. Let h be atutom of $\mathcal{A}$, $R \subseteq A^n$ definable in $\mathcal{A}$ then $\forall a \in A^n a \in R \,\text{iff}\, (h(a_1), \dots h(a_n)) \in R$ Suppose φ defines R in $\mathcal{A}$ we want $\mathcal{A} \models \varphi\,[a]$ iff $\mathcal{A} \models \varphi\,[h(a_1), \dots h(a_n)]$ which is true by the homom. thm.
Note: Corol can be used to show that some $R \subseteq A^n$ is not definable in $\mathcal{A}$

Example 2.19. : $\mathcal{R} = (\mathbb{R}, <)$ then $\mathbb{N}$ is not definable in $\mathcal{R}$. What do automorphisms of $\mathcal{R}$ look right? $h : \mathbb{R} \to \mathbb{R}$ is a bijection and $x < y$ iff $h(x) < h(y)$ so h is strictly increasing. for example $x \mapsto x + \frac{1}{2}$ or $x \mapsto x^3$.

2.6 UNIQUE READABILITY FOR TERMS

Definition 2.22. : We define K on symbols from which terms are built up (variables, constants, function symbols). $K(s) = 1 - n$ where s is a symbol and n is the number of terms that need to follow s in order to obtain a term. $K(x) = 1 = K(c)$ and $K(f) = 1 - n$ where f is an n-ary funciton symbol We now extend K to the set of all expressions which are built up from above symbols (variables, constants, function symbols): $K(s_1, \dots s_n) = K(s_1) + \dots + K(s_n)$ (unique because no symbol is a finite sequence of other symbols)

Lemma 2.1. : t a term then $K(t) = 1$

Proof. $K(x) = 1 = K(c)$ and $K(ft_1, \dots t_n) = 1 - n + n = 1$ ☐

Definition 2.23. : A terminal segment of string of symbols $(s_1, \dots s_n)$ is $(s_k, s_{k+1}, \dots s_n)$ for some $1 \leq k \leq n$.

Lemma 2.2. : Any terminal segment of terms is a concatenation of one or more terms.

Proof. True for variables and constants. $ft_1 \dots t_n$ the only non trivial case is $t'_k t_{k+1} \dots t_m$ where t_k is $t''_k t'_k$ ☐

Corollary 2.3. If t_1 is a proper initial segment of a term t then its $K(t_1) < 1$. proof: let t be $t_1 t_2$ where t_1 is a proper initial segment then $K(t) = 1$ and $K(t_2) \geq 1$ therefore $K(t_1) \leq 0$

Unique readability for terms

The set of terms is freely generated from the set of variables(Var), the set of constant symbols (Const) by the term building operations $\mathcal{F}_f$ for the function symbols f.

Proof.
- disjointment of ranges: Let f and g be two distinct funciton symbols then $\operatorname{ran}\mathcal{F}_f \cap \operatorname{ran}\mathcal{F}_g = \varnothing$ $\operatorname{ran}\mathcal{F}_f \cap Var = \varnothing$ $\operatorname{ran}\mathcal{F}_f \cap Const = \varnothing$
- $\mathcal{F}_f|_{\text{terms}}$ are 1-1: assume $ft_1 \dots t_n = ft'_1 \dots t'_n$ and assume $t_1 \neq t'_1$ then one is an initial segment of the other. Then its K-value has to be less than 1 so it is not a term. $t_1 = t'_1 \dots t_n = t'_n$.

☐

Definition 2.24. : Extend K as follows: $K(() = -1$ $K()) = 1$ $K(\forall) = 1$ $K(\neg) = 0$ $K(\to) = -1$ $K(P) = 1 - n$ for an n-ary rel. symb. P. $K(=) = -1$. Extend K to the set of all expressions by $K(s_1, \dots s_n) = K(s_1) + K(s_n)$ The idea is that K tells us the number of symbols that at least need to follow to obtain a formula.

Lemma 2.3. : for every formula φ it is $K(\varphi) = 1$

Proof. induction on φ ☐

Lemma 2.4. : for every proper initial segment α' of a fla. α we have $K(\alpha') < 1$

Corollary 2.3. No proper initial segment of a fla. is a fla.

The set of flas. is freely generated from the set of atomic flas. by operations $\mathcal{E}_\neg, \mathcal{E}_\to, Q_i$

Proof.
- $\mathcal{E}_\neg, Q_i$ are one to one
- $\mathcal{E}_\to|_{\text{Flas.}}$ then itemwise and use of prev. lemmas
- p.w. disjointness of ranges

☐

2.7 A PARSING ALGORITHM FOR FIRST ORDER LOGIC

2.8 DEDUCTIONS (FORMAL PROOFS)

Definition 2.25. Modus Ponens: We will use one rule of interference, Modus Ponens(MP). Our notation will be:

$$\frac{\alpha, \alpha \to \beta}{\beta} \qquad \text{MP}$$

And it reads as follwos: "If α and $\alpha \to \beta$ then β." This rule is the formalisation of the rather informal statement: "If we know a statement α is true, and this statement implies another statement β, then β must also be true."

Definition 2.26. Deduction: A formal proof (decuction) of a fla φ from a set of formulas Σ is a finite sequence of formulas $(\alpha_0, \alpha_1, \dots, \alpha_n)$ such that $\alpha_n = \varphi$ and for every $i < n$ α_i is either a logical axiom or $\alpha_i \in \Sigma$ or α_i is obtained from α_k and α_l where $0 \leq k, l < i$ by the use of MP, in particular $\alpha_k = \beta \to \alpha_i$ and $\alpha_l = \beta$. If a deduction of φ from Σ exists, we say "φ is deducible from Σ" or "φ is a theorem of Σ".

Note: Deductions are not unique. However we do have an induction principle: If a set of formulas contains all logical axioms and all of Σ and is closed under MP, then it contains all theorems of Σ.

Logical axioms

Definition 2.27. Generalization: ψ is a generalization of φ if $\psi = \forall x_{i_1} \dots \forall x_{i_k} \varphi$

Definition 2.28. Logical axioms: Let x, y be variables and α, β formulas. then the logical axioms are generalizations of the following formulas:

1. tautologies
2. $\forall x \alpha \to \alpha^x_t$ where t is substitutable for x in α
3. $\forall x(\alpha \to \beta) \to (\forall x \alpha \to \forall x \beta)$
4. $\alpha \to \forall x \alpha$ where x does not occur free in α

if our language contains $=$ then

1. $x = x$
2. $x = y \to (\alpha \to \alpha')$ where α' is obtained from α by replacing some of the occurences of x with y.

Ad axiom group (2), Substitution:

Definition 2.29. Substitution: Let α, β be formulas, x a variable and t a term then α^x_t is expression obtained from α by substituting t for x We define substitution inductive as follows:

1. if α is atomic then $\alpha =$ expression obtained from α by replacing all x's by t's
2. $(\neg \alpha)^x_t = \neg(\alpha^x_t)$
3. $(\alpha \to \beta)^x_t = (\alpha^x_t) \to (\beta^x_t)$
4. $(\forall y \alpha)^x_t = \begin{cases} \forall y(\alpha^x_t) & \text{iff } x \neq y \\ \forall x \alpha & \text{iff } x = y \end{cases}$

Example 2.20. *:*

- $\alpha^x_x = \alpha$
- Let $\alpha = \neg \forall y x = y$ what is $\forall x \alpha \to \alpha^x_z$?

  $$\forall x \neg \forall y x = y \rightsquigarrow \neg \forall y z = y$$

  What is $\forall x \alpha \to \alpha^x_y$ $\forall x \neg \forall y x = y$ is true in all structures with a universe A with $|A| \geq 2$.

  $$\forall x \neg \forall y x = y \rightsquigarrow \neg \forall y y = y$$

  and $\neg \forall y y = y$ is an antitautology (it is always false).

-

So we have to define substitutable

Definition 2.30. substitutable: Let x be a variable, t a term. Then t is substitutable for x in α if

1. α atomic then t is SUB for x in α
2. then t is SUB for x in $\neg \alpha$ iff then t is SUB for x in α
3. then t is SUB for x in $\alpha \to \beta$ iff then t is SUB for x in α and β
4. then t is SUB for x in $\forall y \alpha$ iff either
    - x does not occur free in $\forall y \alpha$ or
    - y does not occur in t and t is SUB for x in α

Example 2.21. *:* For instance the following is a logical axiom.

$$\forall x_3(\forall x_1(Ax_1 \to \forall x_2 Ax_2) \to (Ax_2 \to \forall x_2 Ax_2))$$

It is a generalization of $\forall x_1(Ax_1 \to \forall x_2 Ax_2) \to (Ax_2 \to \forall x_2 Ax_2)$ which is by point two a substitution with $\alpha = Ax_1 \to \forall x_2 Ax_2$. Then $\alpha^{x_1}_{x_2} = Ax_2 \to \forall x_2 Ax_2$ And x_2 is indeed substitutable for x_1 in α because it does not get bounded.

$$\forall x_1(\forall x_2 Bx_1x_2 \to \forall x_2 Bx_2x_2)$$

is a generalization of point (2), but x_2 is not substitutable for x_1 in $\alpha =$, therefore it is not a logical axiom.

Ad (1): tautologies

Definition 2.31. **`Tautologies of first order language:`** Tautologies are the formulas obtained from tautologies of propositional logic by replacing all propositional atoms by formulas of first order logic.

An alternative definition is: Divide all formulas of first order logic into two groups:

1. atomic formulas and generalizations of first order formulas (these are called prime formulas)
2. all other formulas i.e. of the form $\neg\alpha$ and $\alpha \to \beta$ (non-prime formulas)

So any first order formula is built up from the prime formulas using finitly many times the formula building operations. $\mathcal{E}_\neg$ $\mathcal{E}_\to$ We have unique readability because the set of formulas is freely generated.

Example 2.22. *:*

$$\neg(\forall y(Px \to Py)) \to (Px \to \forall y \neg Py)$$

is built up from $\neg(\forall y(Px \to Py))$ and $Px \to \forall y\neg Py$. which itself $\forall y(Px \to Py)$ and Px and $\forall y \neg Py$ where they are all prime formulas.

Example 2.23. *:* Is the following a tautology?

$$(\forall y(\neg Py) \to \neg Px) \to (Px \to \neg\forall y\neg Py)$$

We construct the construction tree into prime formulas and then assign truth values to them and evalue the truth value of the whole formula. It is indeed a tautology.

Note:

- $\forall x(Px \to Px)$ is a prime formula which corresponds to a propositional atom, and therefore not a tautology. But it is a generalization of a tautology and therefore by (1) a logical axiom.
- $\forall x Px \to Px$ is not a tautology but is a logical axiom by group (2).

Note: $\Gamma \models_{\text{taut}} \varphi$ from propositional logic can be translated to first order logic.

Lemma 2.5. **:** If $\Gamma \models_{\text{taut}} \varphi$ then $\Gamma \models \varphi$

Proof. Problem set. ☐

Note: The converse fails. For instance $\forall x Px \models Pc$. However Pc is a diffrent propositional atom then $\forall x Px$ they have no connection between them when viewed in propositional logic.

We will prove $\Gamma \models \varphi$ iff $\Gamma \vdash \varphi$ (the first direction is completeness and the converse soundness.)

Theorem 2.4. **:** $\Gamma \vdash \varphi$ iff $\Gamma \cup \Lambda \models_{\text{taut}} \varphi$

Proof.
- Let $\Gamma \vdash \varphi$ and v be a truth assigment that satiesfies every element in $\Gamma \cup \Lambda$. Induction on deduction of φ from Γ.
  - if $\varphi \in \Gamma \cup \Lambda$ then we are done

  - if φ is obtained from α, $\alpha \to \varphi$ by MP then v satiesfies α and $\alpha \to \varphi$
    $\{\alpha, \alpha \to \varphi\} \models_{\text{taut}} \varphi$

- Assume $\Gamma \cup \Lambda \models_{\text{taut}} \varphi$. Then by the compactness theorem for propositional logic there are $\gamma_1 \dots, \gamma_n \in \Gamma$ and $\lambda_1, \dots \lambda_m \in \Lambda$ such that

$$\gamma_1 \to \gamma_2 \to \cdots \to \gamma_n \to \lambda_1 \to \cdots \to \lambda_m$$

  is a tautology (always grouped to the left) because $\Gamma \cup \{\alpha\} \models_{\text{taut}} \beta$ iff $\Gamma \models_{\text{taut}} (\alpha \to \beta)$

□

2.9 GENERALIZATION AND DEDUCTION THEOREM

Note: Intuativly if Γ does not assume anything about x and Γ proves φ then Γ proves $\forall x\varphi$

Theorem 2.5. Generalization theorem: If $\Gamma \vdash \varphi$ and x does not occur free in Γ, then $\Gamma \vdash \forall x\varphi$

Proof. We use axiom group 4, $\alpha \to \forall x\alpha$ if x is not occuring free in α. Since x does not occur free in $\sigma \in \Gamma$, if $\varphi \in \operatorname{Thm}\Gamma$ then $\forall x\varphi \in \operatorname{Thm}\Gamma$. Induction principle: S the set of flas. If $\Lambda \cup \Gamma \subseteq S$ and S is closed under MP then S contains Thm(Γ). It is enough to show that $\{\varphi : \Gamma \vdash \forall x\varphi\}$ contains $\Gamma \cup \Lambda$. and is closed under MP.

1. if φ is a logical axiom then $\forall x\varphi$ is a generalization and therefore also a logical axiom, so $\Gamma \vdash \forall x\varphi$
2. Lets assume $\varphi in \Gamma$. then x does not occur free in any element of Γ, then $\varphi \to \forall x\varphi$ is a logical axiom and $\Gamma \vdash \forall x\varphi$ by MP.
3. Closedness under MP. suppose φ is obtained from ψ, $\psi \to \varphi$ by MP. Then by induction hyphothesis $\Gamma \vdash \forall x\psi$ and $\Gamma \vdash \forall x(\psi \to \varphi)$ Then $\forall x(\psi \to \varphi) \to (\forall x\psi \to \forall x\varphi)$ is a logical axiom in group 3. Then by MP $\Gamma \vdash \forall x\psi \to \forall x\varphi$
   By MP again $\Gamma \vdash \forall x\varphi$

□

Note: Suppose x has free occurence in Γ for example $Px \not\models \forall xPx$ so we can not have $Px \vdash \forall xPx$ (want $\models$ iff $\vdash$)
Note: Proof of Generalization theorem can be used to obtain a deduction of $\forall x\varphi$ from Γ from a deduction of φ from Γ.

Lemma 2.6. Rule T: If $\Gamma \vdash \alpha_1$, $\Gamma \vdash \alpha_2$,$\dots \Gamma \vdash \alpha_n$ and $\{alpha_1, \alpha_2, \dots \alpha_n\} \models_{\text{taut}} \beta$ then $\Gamma \vdash \beta$.

Proof. $\alpha_1 \to \alpha_2 \to \cdots \to \alpha_n \to \beta$ is a logical axiom because it is a tautology. Apply MP n-times. □

Theorem 2.6. Deduction theorem: If $\Gamma \cup \{\gamma\} \vdash \varphi$ then $\Gamma \vdash (\gamma \to \varphi)$

Proof. Assume $\Gamma \cup \{\gamma\} \vdash \varphi$. $\Gamma \cup \{\gamma\} \vdash \varphi$ iff $\Gamma \cup \{\gamma\} \cup \Lambda \models_{\text{taut}} \varphi$
iff $\Gamma \cup \Lambda \models_{\text{taut}} \gamma \to \varphi$ (exercise sheet 1, ex 7)
iff $\Gamma \vdash (\gamma \to \varphi)$ □

Note: Deduction theorem is an equivalence. $\Gamma \vdash \gamma \to \varphi$ then $\Gamma \cup \{\gamma\} \vdash \gamma$. the statement follows by MP.

Corollary 2.6. (Contraposition): If $\Gamma \cup \{\varphi\} \vdash \neg\psi$ then $\Gamma \cup \{\psi\} \vdash \neg\varphi$

Proof. Suppose $\Gamma \cup \{\varphi\} \vdash \neg\psi$ then by deduction theorem $\Gamma \vdash \varphi \to \neg\psi$ We observe that $\{\varphi \to \neg\psi\} \models_{\text{taut}} \psi \to \neg\varphi$.

By rule T: $\Gamma \vdash \psi \to \neg\varphi$ and by the converse of the deduction theorem, by MP we have $\Gamma \cup \{\psi\} \vdash \neg\varphi$ □

Definition 2.32. Inconsistence: A set of flas. Γ is called inconsistent, if for some (euqivalent to all) fla. β it is $\beta, \neg\beta \in \operatorname{Thm}\Gamma$.

Note: If Γ is inconsistent, then for $\alpha \in \operatorname{Thm}\Gamma$. Then $(\beta \to (\neg\beta \to \alpha))$ is a tautology. Use β from definition of inconsistence and use MP twice.

Corollary 2.6. (Reductio ad absurdum): If $\Gamma; \varphi$ inconsistent, then $\Gamma \vdash \neg\varphi$.

Proof. Suppose that $\Gamma; \varphi$ is inconsistent. then for any β $\Gamma; \varphi \vdash \beta$ and $\Gamma; \varphi \vdash \neg\beta$ By the deduction theorem $\Gamma \vdash \varphi \to \beta$ and $\Gamma \vdash \varphi \to \neg\beta$, therefore $\{\varphi \to \beta, \varphi \to \neg\beta\} \models_{\text{taut}} \neg\varphi$ By Rule T: $\Gamma \vdash \neg\varphi$. ☐

Note: strategies for finding deductions can be found in the textbook [1].

Theorem 2.7. Generalization on constants: Suppose $\Gamma \vdash \varphi$ and c is a constant symbol that does not occur in Γ. Then there is a variable y (y does not occur in φ) s.th. $\Gamma \vdash \forall y(\varphi)^c_y$. and moreover also there is a deduction of $\forall y(\varphi)^c_y$ in which c does not occur.

Proof. We will take a deduction $\langle \alpha_1, \dots \alpha_n \rangle$ of φ from Γ. Pick the variable y as the first variable in any α_i for each i. **Claim:** $\langle (\alpha_1)^c_y, \dots (\alpha_n)^c_y \rangle$ is a deduction of $(\varphi)^c_y$ from Γ.
proof of claim. We need to verify that every member $(\alpha)^c_y$ is actually provable from Γ.

- if $\alpha_k \in \Gamma$ then c does not occur in α_k then $(\alpha)^c_y = \alpha_k$
- if $\alpha_k \in \Lambda$ then $(\alpha_k)^c_y$ is also a logical axiom.
- lets say α_k was obtained by α_i, $\alpha_i \to \alpha_k$ $i < k$ by MP. Now take $(\alpha_i \to \alpha_k)^c_y = (\alpha_i)^c_y \to (\alpha_k)^c_y$. (induction hyphothesis) $(\alpha_k)^c_y$ is obtained from $(\alpha_i)^c_y$ nad $(\alpha_i \to \alpha_k)^c_y$. by MP.

☒

Because formal proofs are finite, there is a $\Gamma_0 \subseteq \Gamma$ finite such that Γ_0 consists of the elements of Γ used in our deduction $\langle (\alpha_1)^c_y, \dots (\alpha_n)^c_y \rangle$ (is therefore deduction of $(\varphi)^c_y$ from Γ_0). And because we assumed that y does not occur in Γ_0, so we can use the generalization theorem on $\Gamma_0 \vdash (\varphi)^c_y$ and yield $\Gamma_0 \vdash \forall y(\varphi)^c_y$ ☐

Alphpabetic Variants

We will formalize and proof the statement "You can always rename your bound variables". Why is that impoirtant? Suppose we want to proof that it is provable that $\forall x \forall y P(x, y) \to \forall y P(y, y)$ If we want to use a logical axiom of group 2, we would need to check if y is actually SUB for x. We obviously do not have that because y would get bounded. $\vdash \forall x \forall y P(x, y) \to \forall x \forall z P(x, z) \vdash \forall x \forall z P(x, z) \to \forall y P(y, y)$

Theorem 2.8. Existence of alphabetic variants: Let φ be a fla., x a variable, t a term. Then there exists a fla. φ' such that φ differs from φ only in the choice of names of the bound variables. And

1. $\varphi' \vdash \varphi$ as well as $\varphi \vdash \varphi'$
2. t is SUB for x in φ'

Proof. Define φ' inductively on complexity of φ.

- if φ is atomic, then $\varphi' = \varphi$
- $(\neg\varphi)' = \neg\varphi'$
    1. $\varphi' \vdash \varphi$ and $\varphi \vdash \varphi'$, we want: $\neg\varphi' \vdash \neg\varphi$ as well as $\neg\varphi \vdash \neg\varphi'$ Ok by Contraposition.
    2. ok by definition of SUB
- $(\varphi \to \psi)' = \varphi' \to \psi'$
    1. By assumption: We want $(\varphi \to \psi) \vdash (\varphi \to \psi)'$, it is enough to show $\varphi \to \psi; \varphi' \vdash \psi'$ We have
    $$\varphi \to \psi; \varphi' \vdash \varphi$$
    $$\varphi \to \psi; \varphi' \vdash \psi$$
    2. ok by definition of SUB
- $(\forall y \varphi)'$

Case 1: No occurence of y in t. or $x = y$ (that is, t is substitutable for x in φ). We define $(\forall y\varphi)' = \forall y\varphi'$. All we need to check is part (a). We have that $\forall y\varphi \vdash \varphi$ because $\forall y\varphi \to \varphi$ is an axiom group 2. So $\forall y\varphi \vdash \varphi'$ and therefore by the generalization theorem $\forall x\varphi \vdash \forall y\varphi'$

Case 2: If y does occur in t and $x \neq y$. let z be the variable that is the first variable that does not occur in φ', x, t then set $(\forall y\varphi)' = \forall z(\varphi')^y_z$

2. want t SUB for x in $(\forall y\varphi)'$
   z does not occur in t (choice of z) t is SUB for x in φ'. (ind assumption) Then t is SUB für x in $\forall z(\varphi')^y_z$ iff t is SUB for x in $(\varphi')^y_z$ because $x \neq z$.
1. $\varphi \vdash \varphi'$ (by ind. assumption) Then $\forall y\varphi \vdash \forall y\varphi'$, because

$$\vdash \forall y(\varphi \to \varphi') \to (\forall y\varphi \to \forall y\varphi')(\text{axiom of group 3})$$

then

$$\forall y(\varphi \to \varphi')\text{gen thm}$$

and by MP:

$$\forall y\varphi \to \forall y\varphi'$$

We have $\forall y\varphi' \vdash (\varphi')^y_z$ (axiom of group 2, z does not occur in φ') By Gen Thm. $\forall y\varphi' \vdash \forall z(\varphi')^y_z$ Then
Want $\forall z(\varphi')^y_z \vdash \forall y\varphi$
$\forall z(\varphi')^y_z \vdash ((\varphi')^y_z)^z_y$ (ax of group 2.), y is SUB for z in $(\varphi')^y_z$ bc. φ' does not contain z so all occurences of z in $(\varphi')^y_z$ are free. (we substituted z for free occ of y.) (Re-replacement lemma $((\varphi')^y_z)^z_y = \varphi'$, see problem set.) So we have $\forall z(\varphi')^y_z \vdash \varphi$ We also know that $\varphi' \vdash \varphi$ by the inductive hyphothesis. So $\forall z(\varphi')^y_z \vdash \varphi$ So $\forall z(\varphi')^y_z \vdash \forall y\varphi$ (Gen Thm.)

□

Note: φ' constructed in proof is also called an alphabetic variant of φ
if our language contains equality:

1. $\vdash \forall x x = x$ (ax 5.)
2. $\vdash \forall x\forall y(x = y \to y = x)$ p.122
3. $\vdash \forall x\forall y\forall z(x = y \to (y = z \to x = z))$ (Exercise 11. in [1])
4. $\vdash \forall x_1\forall x_2\forall y_1\forall y_2(x_1 = y_1 \to (x_2 = y_2 \to (Px_1x_2 \to Py_1y_2)))$, similarly for any n-ary predicate. p.128
5. $\vdash \forall x_1\forall x_2\forall y_1\forall y_2(x_1 = y_1 \to (x_2 = y_2 \to (fx_1x_2 = fy_1y_2)))$, similarly for n-ary formula symbol, p.122

2.10 SOUNDNESS AND COMPLETENESS

In first order logic it holds:

- soundness: If $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$
- completeness: If $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$

For the proof of soundness we will have to show that all our axioms are valid. For this we will need the following two lemmas.

Lemma 2.7. `pre-substitution lemma:` Let be a map TODO

Lemma 2.8. `Substitution lemma:` If t SUB x in φ then $\mathcal{A} \models \varphi^x_t\,[s]$ iff $\mathcal{A} \models \varphi\,[s(x|\overline{s}(t))]$

Proof. 1. φ atomic: use pre-substitution lemma.

2. φ is of the form $\neg\psi$ or $\psi \to \theta$ - use induction
3. φ is of the form $\forall y\psi$ and x does not occur free in φ
   $\varphi^x_t = \varphi$ wts. $\mathcal{A} \models \varphi^x_t\,[s]$ iff $\mathcal{A} \models \varphi\,[s(x|\overline{s}(t))]$
   By Theorem 2.2, this is indeed the case, so the lemma holds.

4. φ is $\forall y\psi$ where x occurs free in φ and t is SUB for x in φ. Then it must be: y does not occur in t and t is SUB for x in ψ.
   then $\overline{s}(t) = \overline{s(y|a)}(t)$ for every $a \in A$. Moreover we also have, that $\varphi_t^x = \forall y\psi_t^x$ bc. $x \neq y$
   Then $\mathcal{A} \models \varphi_t^x\ [s]$ iff $\mathcal{A} \models \forall y\psi_t^x\ [s]$
   iff $\mathcal{A} \models \psi_t^x\ [s(y|a)]$ and for all $a \in A$.
   iff $\mathcal{A} \models \psi\ [s(y|a)(x|\overline{s(y|a)}(t))]$ (inductive assumption) and for all $a \in A$
   By above: iff $\mathcal{A} \models \psi\ [s(y|a)(x|\overline{s}(t))]$ for all $a \in A$
   iff $\mathcal{A} \models \forall y\psi\ [s(x|\overline{s}(t))]$

$\square$

Theorem 2.9. : If $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$

Proof. Proof by induction on φ. We have to show:

1. that every logical axiom is valid
2. logical implication is preserved by MP

2. Assume 1. we have to show that if $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$
   - $\varphi \in \Lambda$ by 1.
   - $\varphi \in \Gamma$ then $\Gamma \models \varphi$
   - φ follows by MP from $\psi, \psi \to \varphi$ then by assumption $\Gamma \models \psi$ and $\Gamma \models \psi \to \varphi$ Therefore $\Gamma \models \varphi$

1. Exercise 6 in section 2.2 consists in showing that if a logical axiom is valid, then also its generalization. So generalizations of valid formulas are valid, we therefore may only consider logical axioms that are not generalizations of another logical axiom.

Ax of 1. exercise 3, section 2.3
   3. exercise 3, section 2.2
   4. exercise 4, section 2.2
   5. $x = x$: $\mathcal{A} \models x = x[s]$ because $s(x) = s(x)$
   6. $x = y \to (\alpha \to \alpha')$ where α is atomic fla, and α' is obtained from α by remplacing some occurances of x's with y's. By the deduction theorem, is enough to show that the set of formulas $\{x = y, \alpha\} \models \alpha'$. Let $\mathcal{A}$ be a structure, s an assigment such that $\mathcal{A} \models x = y[s]$
      Claim: for every term t if t' is obtained from t by replacing some x's by y's, then $\overline{s}(t) = \overline{s}(t')$.
      proof of claim. Induction on terms. ☒
      - α of the form $t_1 = t_2$ then α' is $t_1' = t_2'$, use prev. claim.
      - α of the form $Pt_1 \dots t_n$ similar
   2. wts. $\forall x\varphi \to \varphi_t^x$ is valid, where t is SUB for x in φ.
      simple case: $\forall xPx \to Pt$ is valid. Let $\mathcal{A} \models \forall xPx[s]$ then $\mathcal{A} \models \forall xPx[s(x|a)]$ for every $a \in A$. so i.p. for $a = \overline{s}(t)$ this means $\overline{s}(t) \in P^{\mathcal{A}}$ that is $\mathcal{A} \models Pt$. In more generality we will need the substitution lemma: We have $\mathcal{A} \models \forall x\varphi\ [s]$
      this is equivalent to $\forall a \in A$ we have $\mathcal{A} \models \varphi[s(x|a)]$ then in particular $\mathcal{A} \models \varphi[s(x|\overline{s}(t))]$ and by the substitution lemma we have the equivalence to $\mathcal{A} \models \varphi_t^x\ [s]$

$\square$

Corollary 2.9. $\vdash \varphi \leftrightarrow \psi$ then φ, ψ are logically equivalent.

Corollary 2.9. φ' an alphabetic variant of φ then φ, φ' are logically equivalent.

Definition 2.33. : A set of formulas Γ is called satisfiable, whenever there is a structure $\mathcal{A}$ with an assigment into A that for all $\sigma \in \Gamma$ $\mathcal{A} \models \sigma\ [s]$

Corollary 2.9. If Γ is satisfiable then Γ is consistent
Note: This corollary is equivalent to Soundness (Exercise)

completeness

Theorem 2.10. Completness Theorem: $\Gamma \models \varphi \implies \Gamma \vdash \varphi$

Theorem 2.11. Completness Theorem': Every consistent set of formulas is satisfiable.

Note:

- The completeness Theorem is equivalent to completeness theorem'
- The completeness Theorem holds for language of any cardinality.
- We will assume for simplicity that the Language is countable.

Proof. Let Γ be a consisten t set of flas in some language $\mathcal{L}$ The idea of the proof:

- 1.-3. build a new set of formulas Δ
  - $\Gamma \subseteq \Delta$
  - Δ consistent and maximal
  - For every fla φ and every variable x there is constant c $\neg\forall x\varphi \to \neg\varphi^x_c \in \Delta$
- 4. Build $\mathcal{A}$ by A is the set of terms (in expanded language) such that Every fla in Δ w/o. equality $(=)$ is satisfiable in $\mathcal{A}$
- accomodate $=$

1. Add a countable infinite set of new constant symbols to the language $\mathcal{L}$ and call it $\mathcal{L}'$
   Claim: Γ is a consistent set of flas. in $\mathcal{L}'$.
   proof of claim. Why? If not, then $\Gamma \vdash \beta\wedge\neg\beta$ where deduction is in $\mathcal{L}'$ and there occurs finitly many new constant symbols in this deduction. By generalization on constants the new constants in the proof can be replaced by new variables. We get a deduction in the old language $\mathcal{L}$ and that contradicts the assumption that Γ is consistent. ☒

2. Want to add for every formula φ and every variable x $\neg\forall x\varphi \to \varphi^x_c$
   and need to stay consistent. Fix enumeration of pairs (φ, x) where φ' is a $\mathcal{L}'$-fla., x variable.

$$\theta_1 := \forall x_1\varphi_1 \to \neg{\varphi_1}^{x_1}_{c_1}$$

   where c_1 is the first new constant that does not occur in φ_1 $\vdots$

$$\theta_n := \forall x_n\varphi_n \to \neg{\varphi_n}^{x_n}_{c_n}$$

   where c_n is the first new constant that does not occur in φ_n and does not occur in θ_k for $k < n$.
   $\Theta = \{\theta_1, \dots\}$
   Claim: $\Gamma \cup \Theta$ is consistent.
   proof of claim. Suppose it is not. Then let m be minimal such that $\Gamma \cup \{\theta_1 \dots \theta_{m+1}\} \vdash \beta \wedge \neg\beta$. Then by (Raa) $\Gamma \cup \{\theta_1 \dots \theta_m\} \vdash \neg\theta_{m+1}$ θ_{m+1} is of the form

$$\forall x_m\varphi_m \to \neg{\varphi_n}^{x_m}_{c_m}$$

   then by (Rule T)

$$\Gamma \cup \{\theta_1 \dots \theta_m\} \vdash \neg\forall x\varphi$$

   and

$$\Gamma \cup \{\theta_1 \dots \theta_m\} \vdash \varphi^x_c$$

   (star..TODO)
   star: $\Gamma \cup \{\theta_1, \dots \theta_m\} \vdash \forall x\varphi$ By generalization on constants: $\Gamma \cup \{\theta_1, \dots \theta_m\} \vdash \forall x(\varphi^x_c)^c_x$ since c does not occur on the left. also $(\varphi^x_c)^c_x = \varphi$ bc c does not occur in φ. Now we have

$$\Gamma \cup \{\theta_1 \dots \theta_m\} \vdash \neg\forall x\varphi$$

   and

$$\Gamma \cup \{\theta_1, \dots \theta_m\} \vdash \forall x(\varphi^x_c)^c_x$$

   which is a contradiction to minimality of m or the consistentness of Γ. ☒

3. Extend $\Gamma \cup \Theta$ to maximal consistent set. Λ is the set of logical axioms in $\mathcal{L}'$ we know that $\Gamma \cup \Theta$ is consistent. so we know that there is no β
$$\Gamma \cup \Theta \cup \Lambda \models_{\text{taut}} \beta \wedge \neg\beta$$
So we find v a truth assigment on prime flas. that satisfies $\Gamma \cup \Theta \cup \Lambda$ and we are going to use this truth assigment to find the maximal set
$$\Delta := \{\varphi : \overline{v}(\varphi) = 1\}$$
Then for every φ either $\varphi \in \Delta$ or $\neg\varphi \in \Delta$ so we have maximality and we also have consistency bc. $\Delta \vdash \varphi$ then $\Delta \models_{\text{taut}} \varphi$ because $\Lambda \subseteq \Delta$ and that means $\overline{v}(\varphi) = 1$ so $\varphi \in \Delta$. So we have that Δ is consistent. and we say that Δ is deductively closed i.e. $\Delta \vdash \varphi$ then $\varphi \in \Delta$.

4. Construction of an $\mathcal{L}'$ structure $\mathcal{A}$ from Δ. We will firstly replace $=$ with E bin. predicate symbol. $A =$ set of all $\mathcal{L}'$-terms
$E^{\mathcal{A}}$ def. by $uE^{\mathcal{A}}t$ iff $u = t \in \Delta$
$f^{\mathbf{A}}$ def by $f^{\mathbf{A}}(t_1, \dots t_n) = ft_1 \dots t_n$
$c^{\mathcal{A}} := c$
$P^{\mathcal{A}}$ then $P^{\mathcal{A}}t_1, \dots t_n$ iff $Pt_1 \dots t_n \in \Delta$ We take the assignment $s : Var \to A$ by $s(x) = x$
Claim 1: $\overline{s}(t) = t$ for every term t **Claim 2:** for every φ let φ^* be obtained from φ by replacing each $=$ with E then $\mathcal{A} \models \varphi^* \, [s]$ iff $\varphi \in \Delta$

proof of claim.
   - φ atomic then φ is Pt
$$\mathcal{A} \models \varphi^* \, [s] \text{ iff } \mathcal{A} \models Pt \, [s] \text{ iff } \overline{s}(t) \in P^{\mathcal{A}} \text{ iff } t \in P^{\mathcal{A}}$$
φ is uEt then
$$\mathcal{A} \models \varphi^* \, [s] \text{ iff } \mathcal{A} \models uEt \, [s] \text{ iff } \overline{s}(u) E \overline{s}(t) \text{ iff } u = t \in \Delta$$
   - $\neg\varphi$
$$\mathcal{A} \models \neg\varphi^* \, [s] \text{ iff } \mathcal{A} \vDash \varphi \, [s] \text{ iff } \varphi \notin \Delta \text{ iff } \neg\varphi \in \Delta$$
   - $\varphi \to \psi$
$$\mathcal{A} \models \varphi^* \to \psi^*[s] \text{ iff } \mathcal{A} \vDash \varphi^*[s] \text{ or } \mathcal{A} \models \psi^*[s] \text{ iff } \mathcal{A} \models \neg\varphi^*[s] \text{ or } \mathcal{A} \models \psi^*[s] \text{ iff } \neg\varphi \in \Delta \text{ or } \psi \in \Delta \text{ iff} (\varphi \to \psi) \in \Delta$$
   - $\forall x\varphi$ wts. $\mathcal{A} \models \forall x\varphi^* \, [s]$ iff $\forall x\varphi \in \Delta$ Suppose $\mathcal{A} \models \forall x\varphi^* \, [s]$ then $\mathcal{A} \models \varphi^* \, [s(x|c)]$ where c is such that $\neg\forall x\varphi \to \neg\varphi^x_c \in \Delta$ Provided that we have substitutability we have by substitution lemma we know $\mathcal{A} \models (\varphi^x_c)^* \, [s]$ By the inductive hyphothesis $\varphi^x_c \in \Delta$ and $\neg\varphi^x_c \notin \Delta$ so we do not have $\neg\forall x\varphi \notin \Delta$ and by maximality of Δ we have $\forall x\varphi \in \Delta$.
     Suppose $\mathcal{A} \vDash \forall x\varphi^* \, [s]$ then $\mathcal{A} \vDash \varphi^* \, [s(x|t)]$ for some t. By the substitution lemma (providet that t is SUB for x in φ) we can replace x by t in the formula.
     $\mathbf{A} \vDash (\varphi^x_t)^* \, [s]$ by the inductive hyphothesis $\varphi^x_t \notin \Delta$ then $\forall x\varphi \notin \Delta$ becasue Δ is deductively closed. If t is not SUB for x in φ, we know that there exists a logically equivalent alphabetic variant φ' of φ such that t is SUB for x in φ'.

   ⊠

   So at this point we have: If $\mathcal{L}$ does not contain $=$ then take $\mathcal{A}$ reduction to $\mathcal{L}$ and $\mathcal{A}$ w/s satisfies Δ.

5. Define $\mathcal{A}/E$ and assigment
Claim: $E^{\mathcal{A}}$ is a congruence on the structure $\mathcal{A}$ compatible with the predicates and formulas.
   - $E^{\mathcal{A}}$ is equivalence relation
   - $P^{\mathcal{A}}$ compatible w/ $E^{\mathcal{A}}$ i.e. $P^{\mathcal{A}}t_1, \dots t_n$ iff $P^{\mathcal{A}}s_1, \dots s_n$ whenever $t_i E^{\mathcal{A}} s_i$ for all $1 \le i \le n$.
   - $f^{\mathcal{A}}$ compatible w/ $E^{\mathcal{A}}$ i.e. $f^{\mathcal{A}}(\underline{t}) E^{\mathcal{A}} f^{\mathcal{A}}(\underline{s})$ iff $\underline{t} E^{\mathcal{A}} \underline{s}$

Definition 2.34. **:** $\mathcal{A}/_E$ is the structure w/ universe $A/_E$ and $([t_1], \dots [t_n]) \in P^{\mathcal{A}/_E}$ iff $(t_1, \dots t_n) \in P^{\mathcal{A}}$ $f^{\mathcal{A}/_E}([t_1], \dots [t_n]) = [f^{\mathcal{A}}(t_1, \dots t_n)]$ Let $h : A \to A/_E : t \mapsto [t]$ quotient map. note h is surjective. $E^{A/_E}$ realized by equality on $A/_E$: $[t]E^{\mathcal{A}/_E}[s]$ iff $tE^{\mathcal{A}}s$ iff $[t] = [s]$

Claim: $\mathcal{A}/_E$ satisfies Δ w/ $h \circ s$.

proof of claim. Let $\varphi \in \Delta$, $\mathcal{A} \models \varphi^*[s]$ by (4) Want to show $\mathcal{A} \models \varphi^*[s]$ iff $\mathcal{A}/_E \models \varphi^*[h \circ s]$ by Homomorphism Thm.(φ^* has no occurence of $=$, surjectivity) realisation of E in $A/_E$ is the equality in $A/_E$. Take the reduct of $\mathcal{A}/_E$ to $\mathcal{L}$. ☒

□

Exam exam

Corollary 2.11. compactness statements

1. $\Gamma \models \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ s.th. $\Gamma_0 \models \varphi$
2. every finitly satisfiable set of formulas is satisfiable

Proof.

1. $\Gamma \models \varphi$ then by completeness $\Gamma \vdash \varphi$ where the deduction uses only formulas from some $\Gamma_0 \subseteq \Gamma$ finite. By soundness, $\Gamma_0 \models \varphi$
2. Γ finitly satisfiable. Suppose Γ is not satisfiable then by completeness Γ is not consistent. So there has to be some $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \beta \wedge (\neg\beta)$ so Γ is not finitly satisfiable (by soundness).

□

2.10.1 Sizes of models

Let Γ be a consistent set of formulas. Is it possible to

Example 2.24. **:**

1. For each $n \in \mathbb{N}$ there is Γ such that all models of Γ that have size n.
2. DLO (Dense liniear order) w/o endpoints: no finite models

Lemma 2.9. **:** Γ such that all models are finite. Then there has to be $m \in \mathbb{N}$ such that $|\mathcal{A}| \leq m$ for every model $\mathcal{A} \in \operatorname{Mod}\Gamma$

Proof. □

Suppose Γ has models of arbitrarily large finite size.
Idea: Expand language by new constant symbols $c_0, c_1, \dots$.

$$\theta_1 := \ c_0 \neq c_1$$

$$\theta_2 := \ c_0 \neq c_1 \wedge c_1 \neq c_2 \wedge c_0 \neq c_2$$

$\vdots$

$$\theta_n := \bigwedge_{i,j=0 i\neq j}^{n} c_i \neq c_j$$

$$\Theta := \{\theta_1, \theta_2 \dots\}$$

$\Gamma \cup \theta$ finitely satisfiable. By the compactness theorem there exists a $\Theta_0 \subseteq \Theta$ finite TODO. then there is a maximal element θ_k By the compactness theorem reduct to language of Γ is an infinite models of Γ which is a contradiction to all models of Γ are finite.
Note**:** There is no sentence in the language of groups / rings /…that would be satisfied in all finite groups/rings/…and not satisfied in all infinite groups/rings/…

Recall the proof of completeness theorem. $\mathcal{L}$, $|\mathcal{L} = \aleph_0$ Γ consistent set of $\mathcal{L}$-formulas. $\mathcal{A}/_E$ countable.

2.10.2 Completness for uncountable languages

Use (AC) in the form of Zorn's Lemma and Zermelo's Theorem

Theorem 2.12. `Zorn's Lemma:` P partially ordered set such that every chain has an upperbound in P then P contains a maximal element.

Theorem 2.13. `Zermelos's Theorem:` Every set can be well-ordered. That is linearly ordered such that every non-empty set has a smallest element.

ω is the first infinite ordinal. then it is also a cardinal and is called $\aleph_0$

$$A_0 = \{(\varphi_\alpha, x_\alpha) : \alpha < \lambda\}$$

$\vdots$

$$|\mathcal{A}/_E| \leq \lambda$$

2.10.3 Löwenheim-Skolem-Theorem

Theorem 2.14. : Supposee Γ is a set of $\mathcal{L}$-formulas. $|\mathcal{L}| = \lambda$ and lets assume Γ is satisfiable in some infinite structure.
Then for every cardinal $\kappa \geq \lambda$, Γ is satisfiable in a structure of cardinality κ.

Proof. add κ many new constants to the language $\mathcal{L}$.
$\mathcal{L}' = \mathcal{L} \cup \{c_\alpha : \alpha < \kappa\}$
$\Sigma = \{c_\alpha \neq c_\beta : \alpha \leq \beta,\ \alpha, \beta \leq \kappa\}$
Then $\Gamma \cup \Sigma$ is finitly satisfiable in $\mathcal{L}'$. This is because Γ is satisfiable in some infinite structure. By compactness $\Gamma \cup \Sigma$ is satisfiable. We have $\mathcal{A} \models \Gamma \cup \Sigma$ then $|\mathcal{A}| \geq \kappa$.

By the proof of completeness theorem, $\Gamma \cup \Sigma$ has a model of size $\leq \kappa$. Hence it is exactly of size κ. Take reduct TODO ☐

Example 2.25. *:* Language of ZFC $\mathcal{L} = \{\in\}$ is countable. Löwenheim-Skolem guaranties that ZFC has a countable model. called skolems paradox. ZFC knows that there are uncountable sets. explanation: some bijections are missing

Example 2.26. *:*

1. $\overline{\mathbb{R}}$ real field. Thm($\overline{\mathbb{R}}$) has a countable model. $\mathbb{R}_{\text{alg}}$
2. $\mathcal{N} = (\mathbb{N}, 0, S, +, \cdot)$
   Claim: there exists a countable structure $\mathcal{M}$ such that $\mathcal{N} \equiv \mathcal{M}$ but $\mathcal{N} \not\cong \mathcal{M}$ One way is to add new constant c to language $\Sigma = \{0 < c, S0 < c, \dots\}$ is fin satisfiable. So $\Sigma \cup Th(\mathcal{N})$ is fin satisfiable by compactness it is satisfiable

   Take the reduct to original language. $\mathcal{M}$. and $\mathcal{M}$ not isomorphic to $\mathcal{N}$, bc A bijection of $M \to \mathbb{N}$ would have to map c somewhere but for every $S^k 0 < c$ for every k wont be preserved by any map.

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, $\mathrm{Prop}(A)$ the set of prop. generated by A. Further let $\Sigma \subseteq \mathrm{Prop}(A)$ and p, q, r range over $\mathrm{Prop}(A)$.
We say p is Σ-equivalent to q iff $\Sigma \models_{\text{taut}} p \leftrightarrow q$ Σ-Equivalence is an equivalent relation on $\mathrm{Prop}(A)$ and $\mathrm{Prop}(A)/\Sigma$ 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 $X_1, \ldots X_m \subseteq S$ and $\mathcal{A}$ a boolean algebra on S generated by $\{X_1, \ldots X_m\}$. Then $\mathcal{A}$ is finite and isomorphic to $\mathcal{P}(\{1, 2, \ldots n\})$ 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 $\quad a \leq b \text{ and } b \in I \implies a \in I \quad$ 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.

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

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 $\forall z (z \in x \rightarrow z \in y)$ 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

Bibliography

[1] Herbert B Enderton and Herbert Enderton. *A Mathematical Introduction to Logic*. eng. United States: Elsevier Science and Technology, 2001. ISBN: 0122384520.