Advanced Logic for Computer Science

Master 1st Year, 1st Semester 2020/2021

Laurențiu Leuștean

Web page: http://cs.unibuc.ro/~lleustean/

FIRST-ORDER LOGIC

First-order languages

Definition 1.1

A *first-order language* $\mathcal{L}$ *consists of:*

- *a countable set* $V = \{v_n \mid n \in \mathbb{N}\}$ *of variables;*
- *the connectives* $\neg$ *and* $\rightarrow$*;*
- *parantheses:* (,)*;*
- *the equality symbol* $=$*;*
- *the universal quantifier* $\forall$*;*
- *a set* $\mathcal{R}$ *of* *relation symbols;*
- *a set* $\mathcal{F}$ *of* *function symbols;*
- *a set* $\mathcal{C}$ *of* *constant symbols;*
- *an* *arity* *function* $\mathrm{ari} : \mathcal{F} \cup \mathcal{R} \rightarrow \mathbb{N}^*$*.*

- $\mathcal{L}$ is uniquely determined by the quadruple $\tau := (\mathcal{R}, \mathcal{F}, \mathcal{C}, \mathrm{ari})$.
- τ is called the signature of $\mathcal{L}$ or the similaritaty type of $\mathcal{L}$.

First-order languages

Let $\mathcal{L}$ be a first-order language.

- The set $Sym_{\mathcal{L}}$ of symbols of $\mathcal{L}$ is

$$Sym_{\mathcal{L}} := V \cup \{\neg, \rightarrow, (,), =, \forall\} \cup \mathcal{R} \cup \mathcal{F} \cup \mathcal{C}$$

- The elements of $\mathcal{R} \cup \mathcal{F} \cup \mathcal{C}$ are called non-logical symbols.
- The elements of $V \cup \{\neg, \rightarrow, (,), =, \forall\}$ are called logical symbols.

- We denote variables by $x, y, z, v, \ldots$, relation symbols by $P, Q, R \ldots$, function symbols by $f, g, h, \ldots$ and constant symbols by $c, d, e, \ldots$.

- For every $m \in \mathbb{N}^*$ we denote:

$\mathcal{F}_m$:= the set of function symbols of arity m;

$\mathcal{R}_m$:= the set of relation symbols of arity m.

Definition 1.2

The set $Expr_{\mathcal{L}}$ *of* *expressions* *of* $\mathcal{L}$ *is the set of all finite sequences of symbols of* $\mathcal{L}$.

Definition 1.3

The length of an expression θ *is the number of symbols of* θ.

Definition 1.4

Let $\theta = \theta_0\theta_1 \ldots \theta_{k-1}$ *be an expression of* $\mathcal{L}$, *where* $\theta_i \in Sym_{\mathcal{L}}$ *for all* $i = 0, \ldots, k-1$.

- *If* $0 \leq i \leq j \leq k-1$, *then the expression* $\theta_i \ldots \theta_j$ *is called the* (i, j)-*subexpression of* θ.
- *We say that an expression* ψ *appears in* θ *if there exists* $0 \leq i \leq j \leq k-1$ *such that* ψ *is the* (i, j)-*subexpression of* θ.
- *We denote by* $Var(\theta)$ *the set of variables appearing in* θ.

Definition 1.5

The terms of $\mathcal{L}$ are the expressions inductively defined as follows:

(T0) *Every variable is a term.*

(T1) *Every constant symbol is a term.*

(T2) *If $m \geq 1$, $f \in \mathcal{F}_m$ and $t_1, \ldots, t_m$ are terms, then $ft_1 \ldots t_m$ is a term.*

(T3) *Only the expressions obtained by applying rules (T0), (T1), (T2) are terms.*

Notations:

- The set of terms is denoted by $Term_{\mathcal{L}}$.
- Terms are denoted by $t, s, t_1, t_2, s_1, s_2, \ldots$.
- $Var(t)$ is the set of variables that appear in the term t.

Definition 1.6

A term t is called closed if $Var(t) = \emptyset$.

Terms

Proposition 1.7 (Induction on terms)

Let Γ be a set of expressions satisfying the following properties:

- *Γ contains the variables and the constant symbols.*
- *If $m \geq 1$, $f \in \mathcal{F}_m$ and $t_1, \ldots, t_m \in \Gamma$, then $ft_1 \ldots t_m \in \Gamma$.*

Then $Trm_{\mathcal{L}} \subseteq \Gamma$.

It is used to prove that all terms have a propriety $\mathcal{P}$: we define Γ as the set of all expressions satisfying $\mathcal{P}$ and apply induction on terms to obtain that $Trm_{\mathcal{L}} \subseteq \Gamma$.

Formulas

Definition 1.8

The atomic formulas of $\mathcal{L}$ are the expressions having one of the following forms:

- $(s = t)$, where s, t are terms;
- $(Rt_1 \dots t_m)$, where $R \in \mathcal{R}_m$ and $t_1, \dots, t_m$ are terms.

Definition 1.9

The formulas of $\mathcal{L}$ are the expressions inductively defined as follows:

(F0) Every atomic formula is a formula.

(F1) If φ is a formula, then $(\neg\varphi)$ is a formula.

(F2) If φ and ψ are formulas, then $(\varphi \rightarrow \psi)$ is a formula.

(F3) If φ is a formula, then $(\forall x \varphi)$ is a formula for every variable x.

(F4) Only the expressions obtained by applying rules (F0), (F1), (F2), (F3) are formulas.

Formulas

Notations

- The set of formulas is denoted by $Form_{\mathcal{L}}$.
- Formulas are denote by $\varphi, \psi, \chi, \dots$.
- $Var(\varphi)$ is the set of variables that appear in the formula φ.

Unique readability

If φ is a formula, then exactly one of the following hold:

- $\varphi = (s = t)$, where s, t are terms;
- $\varphi = (Rt_1 \dots t_m)$, where $R \in \mathcal{R}_m$ and $t_1, \dots, t_m$ are terms;
- $\varphi = (\neg\psi)$, where ψ is a formula;
- $\varphi = (\psi \to \chi)$, where ψ, χ are formulas;
- $\varphi = (\forall x \psi)$, where x is a variable and ψ is a formula.

Furthermore, φ can be written in a unique way in one of these forms.

Formulas

Proposition 1.10 (Induction on formulas)

Let Γ be a set of expressions satisfying the following properties:

- *Γ contains all atomic formulas.*
- *Γ is closed to $\neg, \rightarrow$ and $\forall x$ (for any variable x), that is: if $\varphi, \psi \in \Gamma$, then $(\neg\varphi), (\varphi \rightarrow \psi), (\forall x \varphi) \in \Gamma$.*

Then $Form_{\mathcal{L}} \subseteq \Gamma$.

It is used to prove that all formulas have a propriety $\mathcal{P}$: we define Γ as the set of all expressions satisfying $\mathcal{P}$ and apply induction on formulas to obtain that $Form_{\mathcal{L}} \subseteq \Gamma$.

Formulas

Derived connectives

Connectives $\vee$, $\wedge$, $\leftrightarrow$ and the existential quantifier $\exists$ are introduced by the following abbreviations:

$$
\begin{array}{lcl}
\varphi \vee \psi & := & ((\neg\varphi) \to \psi) \\
\varphi \wedge \psi & := & \neg(\varphi \to (\neg\psi))) \\
\varphi \leftrightarrow \psi & := & ((\varphi \to \psi) \wedge (\psi \to \varphi)) \\
\exists x\varphi & := & (\neg\forall x(\neg\varphi)).
\end{array}
$$

Conventions and notations

- Usually the external parantheses are omitted, we write them only when necessary. We write $s = t$, $Rt_1 \dots t_m$, $ft_1 \dots t_m$, $\neg\varphi$, $\varphi \to \psi$, $\forall x \varphi$. On the other hand, we write $(\varphi \to \psi) \to \chi$.
- To reduce the use of parentheses, we assume that
    - $\neg$ has higher precedence than $\to, \wedge, \vee, \leftrightarrow$;
    - $\wedge, \vee$ have higher precedence than $\to, \leftrightarrow$;
    - quantifiers $\forall$, $\exists$ have higher precedence than the other connectives. Thus, $\forall x \varphi \to \psi$ is $(\forall x \varphi) \to \psi$ and not $\forall x (\varphi \to \psi)$.
- Hence, the formula $(((\varphi \to (\psi \vee \chi)) \wedge ((\neg\psi) \leftrightarrow (\psi \vee \chi)))$ is written as $(\varphi \to \psi \vee \chi) \wedge (\neg\psi \leftrightarrow \psi \vee \chi)$.

Conventions and notations

- We write sometimes $f(t_1, \ldots, t_m)$ instead of $ft_1 \ldots t_m$ and $R(t_1, \ldots, t_m)$ instead of $Rt_1 \ldots t_m$.
- Function/relation symbols of arity 1 are called unary.
  Function/relation symbols of arity 2 are called binary.
- If f is a binary function symbol, we write $t_1 f t_2$ instead of $f t_1 t_2$.
- If R is a binary relation symbol, we write $t_1 R t_2$ instead of $R t_1 t_2$.

We identify often a language $\mathcal{L}$ with the set of its non-logical symbols and write $\mathcal{L} = (\mathcal{R}, \mathcal{F}, \mathcal{C})$.

Definition 1.11

An $\mathcal{L}$-structure is a quadruple

$$\mathcal{A} = (A, \mathcal{F}^{\mathcal{A}}, \mathcal{R}^{\mathcal{A}}, \mathcal{C}^{\mathcal{A}}),$$

where

- A *is a nonempty set;*
- $\mathcal{F}^{\mathcal{A}} = \{f^{\mathcal{A}} \mid f \in \mathcal{F}\}$ *is a set of functions on* A*; if* f *has arity* m*, then* $f^{\mathcal{A}} : A^m \to A$*;*
- $\mathcal{R}^{\mathcal{A}} = \{R^{\mathcal{A}} \mid R \in \mathcal{R}\}$ *is a set of relations on* A*; if* R *has arity* m*, then* $R^{\mathcal{A}} \subseteq A^m$*;*
- $\mathcal{C}^{\mathcal{A}} = \{c^{\mathcal{A}} \in A \mid c \in \mathcal{C}\}$.
- A *is called the universe of the structure* $\mathcal{A}$. *Notation:* $A = |\mathcal{A}|$
- $f^{\mathcal{A}}$ *(* $R^{\mathcal{A}}$, $c^{\mathcal{A}}$*, respectively) is called the interpretation of* f *(*R*,* c*, respectively) in* $\mathcal{A}$.

Examples - The language of equality $\mathcal{L}_=$

$\mathcal{L}_= = (\mathcal{R}, \mathcal{F}, \mathcal{C})$, where

- $\mathcal{R} = \mathcal{F} = \mathcal{C} = \emptyset$
- this language is proper for expressing the properties of equality.
- $\mathcal{L}_=$-structures are the nonempty sets.

Examples of formulas:

- equality is symmetric:

$$\forall x \forall y (x = y \rightarrow y = x)$$

- the universe has at least three elements:

$$\exists x \exists y \exists z (\neg(x = y) \wedge \neg(y = z) \wedge \neg(z = x))$$

Examples - The language of arithmetics $\mathcal{L}_{ar}$

$\mathcal{L}_{ar} = (\mathcal{R}, \mathcal{F}, \mathcal{C})$, where

- $\mathcal{R} = \{\dot{<}\}$; $\dot{<}$ is a binary relation symbol;
- $\mathcal{F} = \{\dot{+}, \dot{\times}, \dot{S}\}$; $\dot{+}, \dot{\times}$ are binary function symbols and $\dot{S}$ is a unary function symbol;
- $\mathcal{C} = \{\dot{0}\}$.

We write $\mathcal{L}_{ar} = (\dot{<}; \dot{+}, \dot{\times}, \dot{S}; \dot{0})$ or $\mathcal{L}_{ar} = (\dot{<}, \dot{+}, \dot{\times}, \dot{S}, \dot{0})$.

The natural example of $\mathcal{L}_{ar}$-structure:

$$\mathcal{N} := (\mathbb{N}, <, +, \cdot, S, 0),$$

where $S : \mathbb{N} \to \mathbb{N}, S(m) = m + 1$ is the successor function. Thus,

$$\dot{<}^{\mathcal{N}} =<,\ \dot{+}^{\mathcal{N}} = +,\ \dot{\times}^{\mathcal{N}} = \cdot,\ \dot{S}^{\mathcal{N}} = S,\ \dot{0}^{\mathcal{N}} = 0.$$

- Another example of $\mathcal{L}_{ar}$-structure: $\mathcal{A} = (\{0, 1\}, <, \vee, \wedge, \neg, 1)$.

Examples - The language with a binary relation symbol

$\mathcal{L}_R = (\mathcal{R}, \mathcal{F}, \mathcal{C})$, where

- $\mathcal{R} = \{R\}$; R is a binary relation symbol.
- $\mathcal{F} = \mathcal{C} = \emptyset$
- $\mathcal{L}$-structures are nonempty sets together with a binary relation.

- If we are interested in partially ordered sets $(A, \leq)$, we use the symbol $\dot{\leq}$ instead of R and we denote the language by $\mathcal{L}_{\leq}$.
- If we are interested in strictly ordered sets $(A, <)$, we use the symbol $\dot{<}$ instead of R and we denote the language by $\mathcal{L}_{<}$.
- If we are interested in graphs $G = (V, E)$, we use the symbol $\dot{E}$ instead of R and we denote the language by $\mathcal{L}_{Graf}$.
- If we are interested in structures $(A, \in)$, we use the symbol $\dot{\in}$ instead of R and we denote the language by $\mathcal{L}_{\in}$.

SEMANTICS

Interpretation

Let $\mathcal{L}$ be a first-order language and $\mathcal{A}$ be an $\mathcal{L}$-structure.

Definition 1.12

An $\mathcal{A}$-assignment or $\mathcal{A}$-evaluation is a function $e : V \to A$.

When the $\mathcal{L}$-structure $\mathcal{A}$ is clear from the context, we also write simply e is an assignment.

In the following, $e : V \to A$ is an $\mathcal{A}$-assignment.

Definition 1.13 (Interpretation of terms)

The interpretation $t^{\mathcal{A}}(e) \in A$ of a term t under the $\mathcal{A}$-assignment *e is defined by induction on terms :*

- *if $t = x \in V$, then $t^{\mathcal{A}}(e) := e(x)$;*
- *if $t = c \in \mathcal{C}$, then $t^{\mathcal{A}}(e) := c^{\mathcal{A}}$;*
- *if $t = ft_1 \dots t_m$, then $t^{\mathcal{A}}(e) := f^{\mathcal{A}}(t_1^{\mathcal{A}}(e), \dots, t_m^{\mathcal{A}}(e))$.*

Interpretation of formulas

The interpretation

$$\varphi^{\mathcal{A}}(e) \in \{0, 1\}$$

of a *formula* φ *under the* $\mathcal{A}$*-assignment* e is defined by induction on formulas.

$$(s = t)^{\mathcal{A}}(e) = \begin{cases} 1 & \text{if } s^{\mathcal{A}}(e) = t^{\mathcal{A}}(e) \\ 0 & \text{otherwise.} \end{cases}$$

$$(Rt_1 \ldots t_m)^{\mathcal{A}}(e) = \begin{cases} 1 & \text{if } R^{\mathcal{A}}(t_1^{\mathcal{A}}(e), \ldots, t_m^{\mathcal{A}}(e)) \\ 0 & \text{otherwise.} \end{cases}$$

Negation and implication

- $(\neg\varphi)^{\mathcal{A}}(e) = 1 - \varphi^{\mathcal{A}}(e)$;
- $(\varphi \to \psi)^{\mathcal{A}}(e) = \varphi^{\mathcal{A}}(e) \to \psi^{\mathcal{A}}(e)$, where,

$\to: \{0,1\} \times \{0,1\} \to \{0,1\}$,

p	q	$p \to q$
0	0	1
0	1	1
1	0	0
1	1	1

Hence,

- $(\neg\varphi)^{\mathcal{A}}(e) = 1$ iff $\varphi^{\mathcal{A}}(e) = 0$.
- $(\varphi \to \psi)^{\mathcal{A}}(e) = 1$ iff $\big(\ \varphi^{\mathcal{A}}(e) = 0$ or $\psi^{\mathcal{A}}(e) = 1\big)$.

Interpretation of formulas

Notation

For any variable $x \in V$ and any $a \in A$, we define a new $\mathcal{A}$-assignment $e_{x \leftarrow a} : V \to A$ by

$$e_{x \leftarrow a}(v) = \begin{cases} e(v) & \text{if } v \neq x \\ a & \text{if } v = x. \end{cases}$$

Universal quantifier

$$(\forall x \varphi)^{\mathcal{A}}(e) \quad = \quad \begin{cases} 1 & \text{if } \varphi^{\mathcal{A}}(e_{x \leftarrow a}) = 1 \text{ for all } a \in A \\ 0 & \text{otherwise.} \end{cases}$$

Satisfaction relation

Let $\mathcal{A}$ be an $\mathcal{L}$-structure and $e : V \to A$ be an $\mathcal{A}$-assignment.

Definition 1.14

Let φ be a formula. We say that:

- *e satisfies φ in $\mathcal{A}$ if $\varphi^{\mathcal{A}}(e) = 1$. Notation: $\mathcal{A} \vDash \varphi[e]$.*
- *e does not satisfy φ in $\mathcal{A}$ if $\varphi^{\mathcal{A}}(e) = 0$. Notation: $\mathcal{A} \nvDash \varphi[e]$.*

Corollary 1.15

For all formulas φ, ψ and any variable x,

(i) $\mathcal{A} \vDash \neg\varphi[e]$ *iff* $\mathcal{A} \nvDash \varphi[e]$.

(ii) $\mathcal{A} \vDash (\varphi \to \psi)[e]$ *iff* $\mathcal{A} \vDash \varphi[e]$ *implies* $\mathcal{A} \vDash \psi[e]$
iff $\mathcal{A} \nvDash \varphi[e]$ *or* $\mathcal{A} \vDash \psi[e]$.

(iii) $\mathcal{A} \vDash (\forall x\varphi)[e]$ *iff for all* $a \in A$, $\mathcal{A} \vDash \varphi[e_{x\leftarrow a}]$.

Proof: Easy exercise.

Satisfaction relation

$\vee, \wedge, \leftrightarrow$: $\{0,1\} \times \{0,1\} \to \{0,1\}$,

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

p	q	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

Let φ, ψ be formulas and x a variable.

Proposition 1.16

(i) $(\varphi \vee \psi)^{\mathcal{A}}(e) = \varphi^{\mathcal{A}}(e) \vee \psi^{\mathcal{A}}(e)$;

(ii) $(\varphi \wedge \psi)^{\mathcal{A}}(e) = \varphi^{\mathcal{A}}(e) \wedge \psi^{\mathcal{A}}(e)$;

(iii) $(\varphi \leftrightarrow \psi)^{\mathcal{A}}(e) = \varphi^{\mathcal{A}}(e) \leftrightarrow \psi^{\mathcal{A}}(e)$;

(iv) $$(\exists x\varphi)^{\mathcal{A}}(e) = \begin{cases} 1 & \text{if there exists } a \in A \text{ s.t. } \varphi^{\mathcal{A}}(e_{x\leftarrow a}) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Proof: We prove, as an example, (iv).

Satisfaction relation

$$
\begin{aligned}
(\exists x\varphi)^{\mathcal{A}}(e) = 1 \quad &\iff \quad (\neg\forall x\neg\varphi)^{\mathcal{A}}(e) = 1 \iff (\forall x\neg\varphi)^{\mathcal{A}}(e) = 0 \\
&\iff \quad \text{there exists } a \in A \text{ s.t. } (\neg\varphi)^{\mathcal{A}}(e_{x\leftarrow a}) = 0 \\
&\iff \quad \text{there exists } a \in A \text{ s.t. } \varphi^{\mathcal{A}}(e_{x\leftarrow a}) = 1.
\end{aligned}
$$

Corollary 1.17

(i) $\mathcal{A} \models (\varphi \wedge \psi)[e]$ *iff* $\mathcal{A} \models \varphi[e]$ *and* $\mathcal{A} \models \psi[e]$.

(ii) $\mathcal{A} \models (\varphi \vee \psi)[e]$ *iff* $\mathcal{A} \models \varphi[e]$ *or* $\mathcal{A} \models \psi[e]$.

(iii) $\mathcal{A} \models (\varphi \leftrightarrow \psi)[e]$ *iff* $\mathcal{A} \models \varphi[e]$ *iff* $\mathcal{A} \models \psi[e]$.

(iv) $\mathcal{A} \models (\exists x\varphi)[e]$ *iff there exists* $a \in A$ *s.t.* $\mathcal{A} \models \varphi[e_{x\leftarrow a}]$.

Let φ be a formula of $\mathcal{L}$.

Definition 1.18

We say that φ is satisfiable if there exists an $\mathcal{L}$-structure $\mathcal{A}$ and an $\mathcal{A}$-assignment e such that

$$\mathcal{A} \vDash \varphi[e].$$

We also say that $(\mathcal{A}, e)$ is a model of φ.

Attention! It is possible that both φ and $\neg\varphi$ are satisfiable
Example: $\varphi := x = y$ in $\mathcal{L}_=$.

Semantics

Let φ be a formula of $\mathcal{L}$.

Definition 1.19

We say that φ is true in an $\mathcal{L}$-structure $\mathcal{A}$ if for all $\mathcal{A}$-assignments e,

$$\mathcal{A} \vDash \varphi[e].$$

We also say that $\mathcal{A}$ satisfies φ or that $\mathcal{A}$ is a model of φ.

Notation: $\mathcal{A} \vDash \varphi$

Definition 1.20

We say that φ is universally true or logically valid or, simply, valid if for all $\mathcal{L}$-structures $\mathcal{A}$,

$$\mathcal{A} \vDash \varphi.$$

Notation: $\vDash \varphi$

Semantics

Let φ, ψ be formulas of $\mathcal{L}$.

Definition 1.21

φ and ψ are logically equivalent or, simply, equivalent if for all $\mathcal{L}$-structures $\mathcal{A}$ and all $\mathcal{A}$-assignments e,

$$\mathcal{A} \vDash \varphi[e] \textit{ iff } \mathcal{A} \vDash \psi[e].$$

Notation: $\varphi \mathrel{\text{⫤}}\!\vDash \psi$

Definition 1.22

ψ is a logical consequence of φ if for all $\mathcal{L}$-structures $\mathcal{A}$ and all $\mathcal{A}$-assignments e,

$$\mathcal{A} \vDash \varphi[e] \quad \textit{implies} \quad \mathcal{A} \vDash \psi[e].$$

Notation: $\varphi \vDash \psi$

Remark

(i) $\varphi \vDash \psi$ iff $\vDash \varphi \to \psi$.

(ii) $\varphi \mathrel{\text{⫤}}\!\vDash \psi$ iff ($\psi \vDash \varphi$ and $\varphi \vDash \psi$) iff $\vDash \psi \leftrightarrow \varphi$.

Logical consequences and equivalences

For all formulas φ, ψ and all variables x, y,

$$\begin{align}
\neg\exists x\varphi &\;\;\models\mid\;\; \forall x\neg\varphi \tag{1}\\
\neg\forall x\varphi &\;\;\models\mid\;\; \exists x\neg\varphi \tag{2}\\
\forall x(\varphi \wedge \psi) &\;\;\models\mid\;\; \forall x\varphi \wedge \forall x\psi \tag{3}\\
\forall x\varphi \vee \forall x\psi &\;\;\models\;\; \forall x(\varphi \vee \psi) \tag{4}\\
\exists x(\varphi \wedge \psi) &\;\;\models\;\; \exists x\varphi \wedge \exists x\psi \tag{5}\\
\exists x(\varphi \vee \psi) &\;\;\models\mid\;\; \exists x\varphi \vee \exists x\psi \tag{6}\\
\forall x(\varphi \to \psi) &\;\;\models\;\; \forall x\varphi \to \forall x\psi \tag{7}\\
\forall x(\varphi \to \psi) &\;\;\models\;\; \exists x\varphi \to \exists x\psi \tag{8}\\
\forall x\varphi &\;\;\models\;\; \exists x\varphi \tag{9}
\end{align}$$

$$\begin{aligned}
\varphi &\models \exists x \varphi && (10)\\
\forall x \varphi &\models \varphi && (11)\\
\forall x \forall y \varphi &\;\models\!\!\!\dashv\; \forall y \forall x \varphi && (12)\\
\exists x \exists y \varphi &\;\models\!\!\!\dashv\; \exists y \exists x \varphi && (13)\\
\exists y \forall x \varphi &\models \forall x \exists y \varphi. && (14)
\end{aligned}$$

Proof: Exercise.

Equality

Proposition 1.23

For all terms s, t, u,

(i) $\models t = t$;

(ii) $\models s = t \to t = s$;

(iii) $\models s = t \wedge t = u \to s = u$.

Proposition 1.24

For all $m \geq 1$, $f \in \mathcal{F}_m, R \in \mathcal{R}_m$ and all terms $t_i, u_i, i = 1, \ldots, m$,

$$\models (t_1 = u_1) \wedge \ldots \wedge (t_m = u_m) \to ft_1 \ldots t_m = fu_1 \ldots u_m$$

$$\models (t_1 = u_1) \wedge \ldots \wedge (t_m = u_m) \to (Rt_1 \ldots t_m \leftrightarrow Ru_1 \ldots u_m)$$

Free and bound variables

Definition 1.25

Let $\varphi = \varphi_0\varphi_1 \ldots \varphi_{n-1}$ be a formula of $\mathcal{L}$ and x be a variable.

- *We say that x occurs bound on position k in φ if $x = \varphi_k$ and there exists $0 \le i \le k \le j \le n-1$ such that the (i,j)-subexpression of φ has the form $\forall x\psi$.*
- *We say that x occurs free on position k in φ if $x = \varphi_k$, but x does not occur bound on position k in φ.*
- *x is a bound variable of φ if there exists k such that x occurs bound on position k in φ.*
- *x is a free variable of φ if there exists k such that x occurs free on position k in φ.*

Example

Let $\varphi = \forall x(x = y) \to x = z$. Free variables: x, y, z. Bound variables: x.

Notation: $FV(\varphi) :=$ the set of free variables of φ.

Alternative definition

The set $FV(\varphi)$ of free variables of a formula φ can be also defined by induction on formulas:

$$\begin{array}{lcll} FV(\varphi) & = & Var(\varphi), & \text{if } \varphi \text{ is an atomic formula} \\ FV(\neg\varphi) & = & FV(\varphi) & \\ FV(\varphi \to \psi) & = & FV(\varphi) \cup FV(\psi) & \\ FV(\forall x\varphi) & = & FV(\varphi) \setminus \{x\}. & \end{array}$$

Notation: $\varphi(x_1, \ldots, x_n)$ if $FV(\varphi) \subseteq \{x_1, \ldots, x_n\}$.

Proposition 1.26

For any $\mathcal{L}$-structure $\mathcal{A}$ and any $\mathcal{A}$-assignments e_1, e_2,

(i) for any term t,

$$\text{if } e_1(v) = e_2(v) \text{ for all variables } v \in Var(t), \text{ then } t^{\mathcal{A}}(e_1) = t^{\mathcal{A}}(e_2).$$

(ii) for any formula φ,

$$\text{if } e_1(v) = e_2(v) \text{ for all variables } v \in FV(\varphi), \text{ then } \mathcal{A} \vDash \varphi[e_1] \text{ iff } \mathcal{A} \vDash \varphi[e_2].$$

Proposition 1.27

For all formulas φ, ψ and any variable $x \notin FV(\varphi)$,

$$\varphi \;\; \models\!\!\!\!\dashv \;\; \exists x \varphi \tag{15}$$

$$\varphi \;\; \models\!\!\!\!\dashv \;\; \forall x \varphi \tag{16}$$

$$\forall x(\varphi \wedge \psi) \;\; \models\!\!\!\!\dashv \;\; \varphi \wedge \forall x \psi \tag{17}$$

$$\forall x(\varphi \vee \psi) \;\; \models\!\!\!\!\dashv \;\; \varphi \vee \forall x \psi \tag{18}$$

$$\exists x(\varphi \wedge \psi) \;\; \models\!\!\!\!\dashv \;\; \varphi \wedge \exists x \psi \tag{19}$$

$$\exists x(\varphi \vee \psi) \;\; \models\!\!\!\!\dashv \;\; \varphi \vee \exists x \psi \tag{20}$$

$$\forall x(\varphi \to \psi) \;\; \models\!\!\!\!\dashv \;\; \varphi \to \forall x \psi \tag{21}$$

$$\exists x(\varphi \to \psi) \;\; \models\!\!\!\!\dashv \;\; \varphi \to \exists x \psi \tag{22}$$

$$\forall x(\psi \to \varphi) \;\; \models\!\!\!\!\dashv \;\; \exists x \psi \to \varphi \tag{23}$$

$$\exists x(\psi \to \varphi) \;\; \models\!\!\!\!\dashv \;\; \forall x \psi \to \varphi \tag{24}$$

Proof: Exercise.

Definition 1.28

A formula φ is called a sentence if $FV(\varphi) = \emptyset$, that is φ does not have free variables.
Notation: $Sent_{\mathcal{L}} :=$ the set of sentences of $\mathcal{L}$.

Proposition 1.29

Let φ be a sentence. For all $\mathcal{A}$-assignments e_1, e_2,

$$\mathcal{A} \vDash \varphi[e_1] \iff \mathcal{A} \vDash \varphi[e_2]$$

Proof: It is an immediate consequence of Proposition 1.26.(ii) and of the fact that $FV(\varphi) = \emptyset$. $\square$

Definition 1.30

Let φ be a sentence. An $\mathcal{L}$-structure $\mathcal{A}$ is a model of φ if $\mathcal{A} \vDash \varphi[e]$ for an (any) $\mathcal{A}$-assignment e. Notation: $\mathcal{A} \vDash \varphi$

Sets of formulas

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

Definition 1.31

We say that Γ is satisfiable if there exists an $\mathcal{L}$-structure $\mathcal{A}$ and an $\mathcal{A}$-assignment e such that

$$\mathcal{A} \models \gamma[e] \text{ for all } \gamma \in \Gamma.$$

$(\mathcal{A}, e)$ is called a model of Γ.

Definition 1.32

We say that φ is a logical consequence of Γ if for all $\mathcal{L}$-structures $\mathcal{A}$ and all $\mathcal{A}$-assignments $e : V \to A$,

$$(\mathcal{A}, e) \text{ model of } \Gamma \implies (\mathcal{A}, e) \text{ model of } \varphi.$$

Notation: $\Gamma \models \varphi$

Sets of sentences

Let φ be a sentence and Γ be a set of sentences of $\mathcal{L}$.

Definition 1.33

We say that Γ is satisfiable if there exists an $\mathcal{L}$-structure $\mathcal{A}$ such that

$$\mathcal{A} \models \gamma \text{ for all } \gamma \in \Gamma.$$

$\mathcal{A}$ is called a model of Γ. Notation: $\mathcal{A} \models \Gamma$

Definition 1.34

We say that φ is a logical consequence of Γ if for all $\mathcal{L}$-structures $\mathcal{A}$,

$$\mathcal{A} \models \Gamma \implies \mathcal{A} \models \varphi.$$

Notation: $\Gamma \models \varphi$

Tautologies

The notions of tautology and tautological consequence from propositional logic can also be applied to a first-order language $\mathcal{L}$. Intuitively, a tautology is a formula which is "true" based only on the interpretations of the connectives $\neg, \to$.

Definition 1.35

An $\mathcal{L}$-truth assignment is a function $F : Form_{\mathcal{L}} \to \{0,1\}$ satisfying, for all formulas φ, ψ,

- $F(\neg\varphi) = 1 - F(\varphi)$;
- $F(\varphi \to \psi) = F(\varphi) \to F(\psi)$.

Proposition 1.36

For any $\mathcal{L}$-structure $\mathcal{A}$ and any $\mathcal{A}$-assignment e, the function

$$V_{e,\mathcal{A}} : Form_{\mathcal{L}} \to \{0,1\}, \quad V_{e,\mathcal{A}}(\varphi) = \varphi^{\mathcal{A}}(e)$$

is an $\mathcal{L}$-truth assignment.

Proof: Easy exercise.

Tautologies

Definition 1.37

Let φ be a formula and Γ be a set of formulas.

- φ *is a* *tautology* *if* $F(\varphi) = 1$ *for any* $\mathcal{L}$*-truth assignment* F.
- φ *is a* *tautological consequence* *of* Γ *if for any* $\mathcal{L}$*-truth assignment* F,

$$F(\gamma) = 1 \text{ for all } \gamma \in \Gamma \quad \Rightarrow \quad F(\varphi) = 1.$$

Examples of tautologies: $\varphi \to (\psi \to \varphi)$, $(\varphi \to \psi) \leftrightarrow (\neg\psi \to \neg\varphi)$, etc..

Tautologies

Proposition 1.38

Let φ be a formula and Γ be a set of formulas.

(i) If φ is a tautology, then φ is valid.

(ii) If φ is a tautological consequence of Γ, then $\Gamma \vDash \varphi$.

Proof:

(i) Let $\mathcal{A}$ be an $\mathcal{L}$-structure and e an $\mathcal{A}$-assignment. Since φ is a tautology and $V_{e,\mathcal{A}}$ is an $\mathcal{L}$-truth assignment, it follows that $\varphi^{\mathcal{A}}(e) = V_{e,\mathcal{A}}(\varphi) = 1$, that is $\mathcal{A} \vDash \varphi[e]$.

(ii) Let $(\mathcal{A}, e)$ be a model of Γ. Then $\gamma^{\mathcal{A}}(e) = 1$, so $V_{e,\mathcal{A}}(\gamma) = 1$ for all $\gamma \in \Gamma$. Since φ is a tautological consequence of Γ, it follows that $V_{e,\mathcal{A}}(\varphi) = 1$, hence $\varphi^{\mathcal{A}}(e) = 1$, that is $\mathcal{A} \vDash \varphi[e]$.

$\square$

Example

$x = x$ is valid, but $x = x$ is not a tautology.

Substitution

Let x be a variable of $\mathcal{L}$ and u be a term of $\mathcal{L}$.

Definition 1.39

For any term t of $\mathcal{L}$, we define

$$t_x(u) \quad := \quad \textit{the expression obtained from } t \textit{ by replacing all occurences of } x \textit{ with } u.$$

Proposition 1.40

For any term t of $\mathcal{L}$, $t_x(u)$ is a term of $\mathcal{L}$.

Substitution

- We would like to define, similarly, $\varphi_x(u)$ as the expression obtained from φ by replacing all free occurences of x in φ with u.
- We expect that the following natural properties of substitution are true:

$$\vDash \forall x\varphi \to \varphi_x(u) \quad \text{and} \quad \vDash \varphi_x(u) \to \exists x\varphi.$$

As the following example shows, there are problems with this definition.

Let $\varphi := \exists y \neg(x = y)$ and $u := y$. Then $\varphi_x(u) = \exists y \neg(y = y)$.
Avem

- For any $\mathcal{L}$-structure $\mathcal{A}$ with $|A| \geq 2$, $\mathcal{A} \vDash \forall x\varphi$.
- $\varphi_x(u)$ is not satisfiable.

Substitution

Let x be a variable, u a term and φ a formula.

Definition 1.41

We say that x is free for u in φ or that u is substitutable for x in φ if for any variable y that occurs in u, no subformula of φ of the form $\forall y \psi$ contains free occurences of x.

Remark

x is free for u in φ in any of the following cases:

- u does not contain variables;
- φ does not contain variables that occur in u;
- no variable from u occurs bound in φ;
- x does not occur in φ;
- φ does not contain free occurences of x.

Substitution

Let x be a variable, u a term and φ be a formula such that x is free for u in φ.

Definition 1.42

$$\varphi_x(u) \quad := \quad \textit{the expression obtained from } \varphi \textit{ by replacing all free occurences of } x \textit{ in } \varphi \textit{ with } u.$$

We say that $\varphi_x(u)$ is a free substitution.

Proposition 1.43

$\varphi_x(u)$ is a formula of $\mathcal{L}$.

Proof: Exercise.

Free substitution rules out the problems mentioned above, it behaves as expected.

Substitution

Let $\mathcal{A}$ be an $\mathcal{L}$-structure and e be an $\mathcal{A}$-assignment.

Lemma 1.44

Let x be a variable, u a term and $a = u^{\mathcal{A}}(e)$.

(i) For any term t, $(t_x(u))^{\mathcal{A}}(e) = t^{\mathcal{A}}(e_{x \leftarrow a})$.

(ii) For any formula φ, if x is free for u in φ, then

$$\mathcal{A} \vDash \varphi_x(u)[e] \iff \mathcal{A} \vDash \varphi[e_{x \leftarrow a}].$$

The idea of the lemma is simple: modifying an assignment e to evaluate x to a is equivalent to replacing x with a term u whose value under e is a.

Proposition 1.45

Let φ be a formula and x be a variable.

(i) For any term u substitutable for x in φ,

$$\models \forall x\varphi \to \varphi_x(u) \quad \text{and} \quad \models \varphi_x(u) \to \exists x\varphi.$$

(ii) $\models \forall x\varphi \to \varphi$ and $\models \varphi \to \exists x\varphi$.

(iii) For any constant symbol c,

$$\models \forall x\varphi \to \varphi_x(c) \text{ and } \models \varphi_x(c) \to \exists x\varphi.$$

Proof:

(i) Let $\mathcal{A}$ and $e : V \to A$. Then $\mathcal{A} \models \forall x\varphi[e] \iff$ for any $a \in A$, $\mathcal{A} \models \varphi[e_{x\leftarrow a}] \implies$ for $a = u^{\mathcal{A}}(e)$, $\mathcal{A} \models \varphi[e_{x\leftarrow a}] \iff$ $\mathcal{A} \models \varphi_x(u)[e]$ (by Lemma 1.44.(ii)). The second assertion follows by applying the first one to $\neg\varphi$.

(ii) Apply (i) with $u := x$.

(iii) Apply (i) with $u := c$. $\square$

In general, if x and y are variables, φ and $\varphi_x(y)$ are not logically equivalent: let $\mathcal{L}_{ar}$, $\mathcal{N}$ and $e : V \to \mathbb{N}$ such that $e(x) = 3$, $e(y) = 5$, $e(z) = 4$. Then

$$\mathcal{N} \models (x \dot{<} z)[e], \text{ but } \mathcal{N} \not\models (x \dot{<} z)_x(y)[e].$$

However, bound variables can be substituted, with the condition to avoid conflicts.

Proposition 1.46

For any formula φ, distinct variables x and y such that $y \notin FV(\varphi)$ and y is substitutable for x in φ,

$$\exists x\varphi \mathrel{\models\!\!\!\dashv} \exists y\varphi_x(y) \quad \text{and} \quad \forall x\varphi \mathrel{\models\!\!\!\dashv} \forall y\varphi_x(y).$$

In particular, this holds if y is a new variable, that does not occur in φ.

We use Proposition 1.46 as follows: if $\varphi_x(u)$ is not a free substitution (that is x is not free for u in φ), then we replace φ with a logically equivalent formula φ' such that $\varphi'_x(u)$ is a free substitution .

Definition 1.47

For any formula φ and any variables $y_1, \dots, y_k$, the $y_1, \dots, y_k$-free variant φ' of φ is inductively defined as follows:

- *if φ is an atomic formula, then φ' is φ;*
- *if $\varphi = \neg\psi$, then φ' is $\neg\psi'$;*
- *if $\varphi = \psi \to \chi$, then φ' is $\psi' \to \chi'$;*
- *if $\varphi = \forall z \psi$, then*

  $$\varphi' = \begin{cases} \forall w \psi'_z(w) & \text{if } z \in \{y_1, \dots, y_k\} \\ \forall z \psi' & \text{altfel;} \end{cases}$$

  where w is the first variable in the sequence $v_0, v_1, \dots,$ which does not occur in ψ' and is not among $y_1, \dots, y_k$.

Definition 1.48

φ' is a variant of φ if it is the $y_1, \ldots, y_k$-free variant of φ for some variables $y_1, \ldots, y_k$.

Proposition 1.49

(i) For any formulas φ and φ', if φ' is a variant of φ, then $\varphi \mathrel{\text{⫤}\!\vDash} \varphi'$;

(ii) For any formula φ and any term u, if the variables of u are among $y_1, \ldots, y_k$ and φ' is the $y_1, \ldots, y_k$-free variant of φ, then $\varphi'_x(u)$ is a free substitution.

Prenex normal form

Definition 1.50

A formula that does not contain quantifiers is called quantifier-free.

Definition 1.51

A formula φ is in prenex normal form if

$$\varphi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \, \psi,$$

where $n \in \mathbb{N}$, $Q_1, \dots, Q_n \in \{\forall, \exists\}$, $x_1, \dots, x_n$ are variables and ψ is a quantifier-free formula. $Q_1 x_1 Q_2 x_2 \dots Q_n x_n$ is the prefix of φ and ψ is called the matrix of φ.

Any quantifier-free formula is in prenex normal form, as one can take $n = 0$ in the above definition.

Prenex normal form

Examples of formulas in prenex normal form:

- universal formulas: $\varphi = \forall x_1 \forall x_2 \ldots \forall x_n \, \psi$, where ψ is quantifier-free
- existential formulas: $\varphi = \exists x_1 \exists x_2 \ldots \exists x_n \, \psi$, where ψ is quantifier-free
- $\forall\exists$-formulas: $\varphi = \forall x_1 \forall x_2 \ldots \forall x_n \exists y_1 \exists y_2 \ldots \exists y_k \psi$, where ψ is quantifier-free
- $\forall\exists\forall$-formulas: $\varphi = \forall x_1 \ldots \forall x_n \exists y_1 \ldots \exists y_k \forall z_1 \ldots \forall z_p \psi$, where ψ is quantifier-free

Theorem 1.52 (Prenex normal form theorem)

For any formula φ there exists a formula φ^* in prenex normal form such that $\varphi \mathrel{\text{⊨⫤}} \varphi^*$ and $FV(\varphi) = FV(\varphi^*)$. φ^* is called a prenex normal form of φ.

Prenex normal form

Let $\mathcal{L}$ be a first-order language containing

- two unary relation symbols R, S and two binary relation symbols P, Q;
- a unary function symbol f and a binary function symbol g;
- two constant symbols c, d.

Example

Find a prenex normal form of the formula

$$\varphi := \exists y(g(y,z) = c) \wedge \neg\exists x(f(x) = d)$$

We have that

$$\begin{aligned}
\varphi \quad &\mathrel{\text{⫤⊨}} \quad \exists y\big(g(y,z) = c \wedge \neg\exists x(f(x) = d)\big) \\
&\mathrel{\text{⫤⊨}} \quad \exists y\big(g(y,z) = c \wedge \forall x\neg(f(x) = d)\big) \\
&\mathrel{\text{⫤⊨}} \quad \exists y\forall x\big(g(y,z) = c \wedge \neg(f(x) = d)\big)
\end{aligned}$$

Thus, $\varphi^* = \exists y\forall x\big(g(y,z) = c \wedge \neg(f(x) = d)\big)$ is a prenex normal form of φ.

Example

Find a prenex normal form of the formula

$$\varphi := \neg\forall y(S(y) \to \exists z R(z)) \wedge \forall x(\forall y P(x,y) \to f(x) = d).$$

$$\begin{aligned}
\varphi \quad &\mathrel{\vDash\!\!\!\Dashv} \quad \exists y \neg(S(y) \to \exists z R(z)) \wedge \forall x(\forall y P(x,y) \to f(x) = d) \\
&\mathrel{\vDash\!\!\!\Dashv} \quad \exists y \neg\exists z(S(y) \to R(z)) \wedge \forall x(\forall y P(x,y) \to f(x) = d) \\
&\mathrel{\vDash\!\!\!\Dashv} \quad \exists y \neg\exists z(S(y) \to R(z)) \wedge \forall x \exists y(P(x,y) \to f(x) = d) \\
&\mathrel{\vDash\!\!\!\Dashv} \quad \exists y \forall z \neg(S(y) \to R(z)) \wedge \forall x \exists y(P(x,y) \to f(x) = d) \\
&\mathrel{\vDash\!\!\!\Dashv} \quad \exists y \forall z \Big(\neg(S(y) \to R(z)) \wedge \forall x \exists y(P(x,y) \to f(x) = d)\Big) \\
&\mathrel{\vDash\!\!\!\Dashv} \quad \exists y \forall z \forall x \Big(\neg(S(y) \to R(z)) \wedge \exists y(P(x,y) \to f(x) = d)\Big) \\
&\mathrel{\vDash\!\!\!\Dashv} \quad \exists y \forall z \forall x \Big(\neg(S(y) \to R(z)) \wedge \exists v(P(x,v) \to f(x) = d)\Big) \\
&\mathrel{\vDash\!\!\!\Dashv} \quad \exists y \forall z \forall x \exists v \Big(\neg(S(y) \to R(z)) \wedge (P(x,v) \to f(x) = d)\Big)
\end{aligned}$$

Theories

Notation: For any set Γ of sentences, denote

$$Mod(\Gamma) := \text{the class of models of } \Gamma.$$

We write $Mod(\varphi_1, \ldots, \varphi_n)$ instead of $Mod(\{\varphi_1, \ldots, \varphi_n\})$.

Lemma 1.53

For any sets Γ, Δ of sentences and any sentence ψ,

(i) $\Gamma \models \psi \iff Mod(\Gamma) \subseteq Mod(\psi)$.

(ii) $\Gamma \subseteq \Delta \implies Mod(\Delta) \subseteq Mod(\Gamma)$.

(iii) Γ *is satisfiable* $\iff Mod(\Gamma) \neq \emptyset$.

Proof: Easy exercise.

Theories

Definition 1.54

A theory *is a set T of sentences of $\mathcal{L}$ that is closed under logical consequence, that is:*

$$\textit{for any sentence } \varphi, \quad T \models \varphi \implies \varphi \in T.$$

Definition 1.55

For any set Γ of sentences, the theory generated by *Γ is the set*

$$\begin{aligned} Th(\Gamma) &:= \{\varphi \mid \varphi \textit{ is a sentence and } \Gamma \models \varphi\} \\ &= \{\varphi \mid \varphi \textit{ is a sentence and } Mod(\Gamma) \subseteq Mod(\varphi)\}. \end{aligned}$$

Proposition 1.56

(i) $\Gamma \subseteq Th(\Gamma)$.

(ii) $Mod(\Gamma) = Mod(Th(\Gamma))$.

(iii) $Th(\Gamma)$ *is a theory.*

(iv) $Th(\Gamma)$ *is the smallest theory* T *with* $\Gamma \subseteq T$.

Proof:

(i) For any $\varphi \in \Gamma$, we have that $\Gamma \models \varphi$, so $\varphi \in Th(\Gamma)$.

(ii) "$\supseteq$" By (i) and Lemma 1.53.(ii).
"$\subseteq$" By the definition of $Th(\Gamma)$.

(iii) For any sentence φ, we have that
$Th(\Gamma) \models \varphi \iff Mod(Th(\Gamma)) \subseteq Mod(\varphi)$
$\iff Mod(\Gamma) \subseteq Mod(\varphi)$ (by (ii)) $\iff \varphi \in Th(\Gamma)$.

(iv) Let T be a theory that contains Γ and $\varphi \in Th(\Gamma)$. Since $Mod(\Gamma) \subseteq Mod(\varphi)$ and $Mod(T) \subseteq Mod(\Gamma)$, we get that $Mod(T) \subseteq Mod(\varphi)$, hence $T \models \varphi$. Since T is a theory, we obtain that $\varphi \in T$. Thus, $Th(\Gamma) \subseteq T$. $\square$

Proposition 1.57

For any sets Γ, Δ *of sentences,*

(i) $\Gamma \subseteq \Delta \implies Th(\Gamma) \subseteq Th(\Delta)$.

(ii) Γ *is a theory* $\iff \Gamma = Th(\Gamma)$.

(iii) $Th(\emptyset) = \{\varphi \mid \varphi$ *is a valid sentence*$\}$ *is included in any theory.*

Proof: Easy exercise.

- A theory expressed as $Th(\Gamma)$ is called an axiomatic theory or a theory presented axiomatically. Γ is called a set of axioms for $Th(\Gamma)$.
- Any theory can be trivially presented axiomatically, but we are interested on sets of axioms that satisfy some "nice" conditions.

Definition 1.58

A theory T is finitely axiomatizable if $T = Th(\Gamma)$ for a finite set Γ of sentences.

Definition 1.59

A class $\mathcal{K}$ of $\mathcal{L}$-structures is axiomatizable if $\mathcal{K} = Mod(\Gamma)$ for a set Γ of sentences. We also say that Γ axiomatizes $\mathcal{K}$.

Definition 1.60

A class $\mathcal{K}$ of $\mathcal{L}$-structures is finitely axiomatizable if $\mathcal{K} = Mod(\Gamma)$ for a finite set Γ of sentences.

Example - The theory of equivalence relations

- $\mathcal{L}_{\dot{\equiv}} = (\dot{\equiv}, \emptyset, \emptyset) = (\dot{\equiv})$
- $\mathcal{L}_{\dot{\equiv}}$-structures are $\mathcal{A} = (A, \equiv)$, $\equiv$ is a binary relation.

Consider the following sentences:

$$\begin{aligned}
(REFL) &:= \forall x (x \dot{\equiv} x) \\
(SIM) &:= \forall x \forall y (x \dot{\equiv} y \rightarrow y \dot{\equiv} x) \\
(TRANZ) &:= \forall x \forall y \forall z (x \dot{\equiv} y \wedge y \dot{\equiv} z \rightarrow x \dot{\equiv} z)
\end{aligned}$$

Definition

The theory of equivalence relations is

$$T := Th((REFL), (SIM), (TRANZ)).$$

- T is finitely axiomatizable.

Example - The theory of equivalence relations

- Let $\mathcal{K}$ be the class of structures $(A, \equiv)$, where $\equiv$ is an equivalence relation on A.
- $\mathcal{K} = Mod(T)$, hence T axiomatizes $\mathcal{K}$.
- We also say that T axiomatizes the class of equivalence relations.

- If we add the axiom:

$$\forall x \exists y \big(\neg(x = y) \land x \dot{\equiv} y \land \forall z (z \dot{\equiv} x \to (z = x \lor z = y))\big),$$

we obtain the theory of equivalence relations with the property that any equivalence class has exactly two elements.

Example - Graph theory

A graph is a pair $G = (V, E)$ of sets such that E is a set of subsets with 2 elements of V. The elements of V are called vertices and the elements of E are called edges.

- $\mathcal{L}_{Graf} = (\dot{E}, \emptyset, \emptyset) = (\dot{E})$
- $\mathcal{L}_{Graf}$-structures are $\mathcal{A} = (A, E)$, where E is a binary relation.

Consider the following sentences:

$$\begin{aligned} (IREFL) &:= \forall x \neg \dot{E}(x, x) \\ (SIM) &:= \forall x \forall y (\dot{E}(x, y) \rightarrow \dot{E}(y, x)). \end{aligned}$$

Definition

Graph theory is $T := Th((IREFL), (SIM))$.

- T is finitely axiomatizable.
- models of T are the graphs.
- T axiomatizes the class of graphs.

Example - The theory of partial order

- $\mathcal{L}_{\dot{\leq}} = (\dot{\leq}, \emptyset, \emptyset) = (\dot{\leq})$
- $\mathcal{L}_{\dot{\leq}}$-structures are $\mathcal{A} = (A, \leq)$, where $\leq$ is a binary relation.

Consider the following sentences:

$$\begin{aligned} (REFL) &:= \forall x (x \dot{\leq} x) \\ (ANTISIM) &:= \forall x \forall y (x \dot{\leq} y \wedge y \dot{\leq} x \to x = y) \\ (TRANZ) &:= \forall x \forall y \forall z (x \dot{\leq} y \wedge y \dot{\leq} z \to x \dot{\leq} z) \end{aligned}$$

Definition

The theory of partial order is

$$T := Th((REFL), (ANTISIM), (TRANZ)).$$

- T is finitely axiomatizable.
- models of T are partially ordered sets.
- T axiomatizes the class of partial order relations.

Example - The theory of strict order

- $\mathcal{L}_{\dot{<}} = (\dot{<}, \emptyset, \emptyset) = (\dot{<})$
- $\mathcal{L}_{\dot{<}}$-structures are $\mathcal{A} = (A, <)$, where $<$ is a binary relation.

Consider the following sentences:

$$\begin{aligned} (IREFL) &:= \forall x \neg (x \dot{<} x) \\ (TRANZ) &:= \forall x \forall y \forall z (x \dot{<} y \wedge y \dot{<} z \rightarrow x \dot{<} z) \end{aligned}$$

Definition

The theory of strict order is

$$T := Th((IREFL), (TRANZ)).$$

- T is finitely axiomatizable.
- models of T are the strictly ordered sets.
- T axiomatizes the class of strict order relations.

Example - The theory of total order

Consider the following sentence:

$$(TOTAL) \quad := \quad \forall x \forall y (x = y \vee x \dot{<} y \vee y \dot{<} x)$$

Definition

The theory of total order is

$$T := Th((IREFL), (TRANZ), (TOTAL)).$$

- T is finitely axiomatizable.
- models of T are totally (linear) ordered sets.
- T axiomatizes the class of total order relations.

Example - The theory of dense order

Consider the following sentence:

$$(DENS) \quad := \quad \forall x \forall y \big(x \dot{<} y \rightarrow \exists z (x \dot{<} z \wedge z \dot{<} y)\big).$$

Definition

The theory of dense order is

$$T := Th((IREFL), (TRANZ), (TOTAL), (DENS)).$$

- T is finitely axiomatizable.
- models of T are the densely ordered sets.
- T axiomatizes the class of dense order relations.

Example - Theory of equality

For all $n \geq 2$, we denote by $\exists^{\geq n}$ the following sentence:

$$\exists x_1 \ldots \exists x_n \big(\neg(x_1 = x_2) \wedge \neg(x_1 = x_3) \wedge \ldots \wedge \neg(x_{n-1} = x_n) \big),$$

written in a more compact way as follows:

$$\exists^{\geq n} = \exists x_1 \ldots \exists x_n \left(\bigwedge_{1 \leq i < j \leq n} \neg(x_i = x_j) \right).$$

Proposition 1.61

For any $\mathcal{L}$-structure $\mathcal{A}$ and any $n \geq 2$,

$$\mathcal{A} \models \exists^{\geq n} \iff A \textit{ has at least } n \textit{ elements.}$$

Proof: Easy exercise.

Example - Theory of equality

Notations

- For uniformity, let $\exists^{\geq 1} := \exists x(x = x)$.
- Denote $\exists^{\leq n} := \neg\exists^{\geq n+1}$ and $\exists^{=n} := \exists^{\leq n} \wedge \exists^{\geq n}$

Proposition 1.62

For any $\mathcal{L}$-structure $\mathcal{A}$ and any $n \geq 1$,

$$\begin{aligned} \mathcal{A} \models \exists^{\leq n} &\iff A \text{ has at most } n \text{ elements} \\ \mathcal{A} \models \exists^{=n} &\iff A \text{ has exactly } n \text{ elements.} \end{aligned}$$

Proof: Easy exercise.

Proposition 1.63

Let $T := Th(\{\exists^{\geq n} \mid n \geq 1\})$. Then for any $\mathcal{L}$-structure $\mathcal{A}$,

$$\mathcal{A} \models T \iff A \text{ is an infinite set.}$$

Proof: Easy exercise.

Complete theories

Let $\mathcal{L}$ be a first-order language.

Definition 1.64

A set of sentences Γ is said to be complete if for any sentence φ,

$$\Gamma \models \varphi \text{ or } \Gamma \models \neg\varphi.$$

Remark 1.65

A theory T is complete iff for any sentence φ, we have that $\varphi \in T$ or $\neg\varphi \in T$.

Definition 1.66

For any $\mathcal{L}$-structure $\mathcal{A}$, the theory of $\mathcal{A}$ is defined as:

$$Th(\mathcal{A}) := \{\varphi \mid \varphi \text{ is a sentence and } \mathcal{A} \models \varphi\}.$$

Proposition 1.67

For any $\mathcal{L}$-structure $\mathcal{A}$, $Th(\mathcal{A})$ is a complete theory and $\mathcal{A}$ is a model of $Th(\mathcal{A})$.

Proof: Easy exercise.

Definition 1.68

Two $\mathcal{L}$-structures $\mathcal{A}$ and $\mathcal{B}$ are called elementarily equivalent if for any sentence φ,

$$\mathcal{A} \models \varphi \quad \Longleftrightarrow \quad \mathcal{B} \models \varphi.$$

Notation: $\mathcal{A} \equiv \mathcal{B}$

Proposition 1.69

For any $\mathcal{L}$-structures $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \equiv \mathcal{B} \iff Th(\mathcal{A}) = Th(\mathcal{B})$.

Proof: Easy exercise.

Theorem 1.70 (Compactness Theorem)

A set Γ of sentences is satisfiable iff every finite subset of Γ is satisfiable.

- one of the central results in first-order logic

Let $\mathcal{L}$ be a first-order language.

Proposition 1.71

The class of finite $\mathcal{L}$-structures is not axiomatizable, that is there exists no set of sentences Γ such that

$$(*) \quad \text{for any } \mathcal{L}\text{-structure } \mathcal{A}, \quad \mathcal{A} \vDash \Gamma \iff \mathcal{A} \text{ is finite.}$$

Proof: Suppose, for the sake of contradiction, that there exists $\Gamma \subseteq Sen_{\mathcal{L}}$ such that (*) holds. Let

$$\Delta := \Gamma \cup \{\exists^{\geq n} \mid n \geq 1\}.$$

We prove that Δ is satisfiable with the help of the Compactness Theorem. Let Δ_0 be a finite subset of Δ. Then

$$\Delta_0 \subseteq \Gamma \cup \{\exists^{\geq n_1}, \ldots, \exists^{\geq n_k}\} \quad \text{for some } k \in \mathbb{N}.$$

Let $\mathcal{A}$ be a finite $\mathcal{L}$-structure such that $|A| \geq \max\{n_1, \ldots, n_k\}$. Then $\mathcal{A} \vDash \exists^{\geq n_i}$ for all $i = 1, \ldots, k$ and $\mathcal{A} \vDash \Gamma$, since $\mathcal{A}$ is finite.

We get that $\mathcal{A} \models \Gamma \cup \{\exists^{\geq n_1}, \ldots, \exists^{\geq n_k}\}$, so $\mathcal{A} \models \Delta_0$. Thus, Δ_0 is satisfiable.

Applying the Compactness Theorem, it follows that

$$\Delta = \Gamma \cup \{\exists^{\geq n} \mid n \geq 1\}$$

has a model $\mathcal{B}$.
Since $\mathcal{B} \models \Gamma$, we have that $\mathcal{B}$ is finite.
Since $\mathcal{B} \models \{\exists^{\geq n} \mid n \geq 1\}$, we have that $\mathcal{B}$ is infinite.
We have obtained a contradiction. $\square$

Corollary 1.72

The class of finite nonempty sets is not axiomatizable in $\mathcal{L}_=$.

Proof: Exercise.

Proposition 1.73

The class of infinite $\mathcal{L}$-structures is axiomatizable, but it is not finitely axiomatizable.

Proof: Denote by $\mathcal{K}_{Inf}$ the class of infinite $\mathcal{L}$-structures.
By Proposition 1.63, for any $\mathcal{L}$-structure $\mathcal{A}$,

$$\mathcal{A} \in \mathcal{K}_{Inf} \iff A \text{ is infinite } \iff \mathcal{A} \vDash \{\exists^{\geq n} \mid n \geq 1\}.$$

Hence,

$$\mathcal{K}_{Inf} = Mod(\{\exists^{\geq n} \mid n \geq 1\}),$$

so it is axiomatizable.

Suppose that $\mathcal{K}_{Inf}$ is finitely axiomatizable, hence there exists

$$\Gamma := \{\varphi_1, \ldots, \varphi_n\} \subseteq Sen_{\mathcal{L}} \text{ such that } \mathcal{K}_{Inf} = Mod(\Gamma).$$

Let $\varphi := \varphi_1 \wedge \ldots \wedge \varphi_n$. Then $\mathcal{K}_{Inf} = Mod(\varphi)$.
It follows that for any $\mathcal{L}$-structure $\mathcal{A}$,

$$\mathcal{A} \text{ is finite } \iff \mathcal{A} \notin \mathcal{K}_{Inf} \iff \mathcal{A} \not\models \varphi \iff \mathcal{A} \models \neg\varphi.$$

Thus, the class of finite $\mathcal{L}$-structures is axiomatizable, which is a contradiction to Proposition 1.71. $\square$.

Corollary 1.74

The class of infinite sets is axiomatizable in $\mathcal{L}_=$, but not finitely axiomatizable in $\mathcal{L}_=$.

Proof: Exercise.

Proposition 1.75

Let Γ be a set of sentences of $\mathcal{L}$ satisfying

(*)   for all $m \in \mathbb{N}$, Γ has a finite model of cardinality $\geq m$.

Then Γ has an infinite model.

Proof: Let

$$\Delta := \Gamma \cup \{\exists^{\geq n} \mid n \geq 1\}.$$

We prove that Δ is satisfiable with the help of the Compactness Theorem. Let Δ_0 be a finite subset of Δ. Then

$$\Delta_0 \subseteq \Gamma \cup \{\exists^{\geq n_1}, \ldots, \exists^{\geq n_k}\} \text{ for some } k \in \mathbb{N}.$$

Let $m := \max\{n_1, \ldots, n_k\}$. By (*), Γ has a finite model $\mathcal{A}$ such that $|A| \geq m$. Then $\mathcal{A} \vDash \exists^{\geq n_i}$ for all $i = 1, \ldots, k$, so $\mathcal{A} \vDash \Delta_0$. Applying the Compactness Theorem, it follows that Δ has a model $\mathcal{B}$. Hence, $\mathcal{B}$ is an infinite model of Γ. $\square$

Proposition 1.76

Assume that a sentence φ is true in all infinite $\mathcal{L}$-structures. Then there exists $m \in \mathbb{N}$ with the property that

$$\varphi \text{ is true in any finite } \mathcal{L}\text{-structure of cardinality } \geq m.$$

Proof: Suppose that the conclusion is not true. Let $\Gamma := \{\neg\varphi\}$. Then for all $m \in \mathbb{N}$, Γ has a finite model of cardinality $\geq m$. Applying Proposition 1.75, we get that Γ has an infinite model $\mathcal{A}$. Hence, $\mathcal{A} \not\models \varphi$, which contradicts the hypothesis. $\square$

Proposition 1.77

Let Γ be a set of sentences satisfying

(*) *for all $m \in \mathbb{N}$, Γ has a finite model of cardinality $\geq m$.*

Then

- (i) *Γ has an infinite model.*
- (ii) *The class of finite models of Γ is not axiomatizable.*
- (iii) *The class of infinite models of Γ is axiomatizable, but it is not finitely axiomatizable.*

Proof: Exercise.

Consider the language $\mathcal{L} = (\dot{+}, \dot{\times}, \dot{S}, \dot{0})$, where $\dot{+}, \dot{\times}$ are binary function symbols, $\dot{S}$ is a unary function symbol and $\dot{0}$ is a constant symbol.

For all $n \in \mathbb{N}$, define by induction the term $\Delta(n)$ of $\mathcal{L}$ as follows:

$$\Delta(0) = \dot{0}, \quad \Delta(n+1) = \dot{S}\Delta(n).$$

Let us consider the $\mathcal{L}$-structure $\mathcal{N} = (\mathbb{N}, +, \cdot, S, 0)$. Then $\Delta(n)^{\mathcal{N}} = n$ for all $n \in \mathbb{N}$. Hence, $\mathbb{N} = \{\Delta(n)^{\mathcal{N}} \mid n \in \mathbb{N}\}$.

Definition 1.78

An $\mathcal{L}$-structure $\mathcal{A}$ is called non-standard if there exists $a \in A$ such that $a \neq \Delta(n)^{\mathcal{A}}$ for any $n \in \mathbb{N}$. Such an element a is called non-standard.

Theorem 1.79

There exists a non-standard model of the theory $Th(\mathcal{N})$.

Proof: Let c be a new constant symbol, $\mathcal{L}^+ = \mathcal{L} \cup \{c\}$ and

$$\Gamma = Th(\mathcal{N}) \cup \{\neg(\Delta(n) = c) \mid n \in \mathbb{N}\}.$$

We prove that Γ is satisfiable by using the Compactness Theorem.
Let Γ_0 be a finite subset of Γ,

$$\Gamma_0 \subseteq Th(\mathcal{N}) \cup \{\neg(\Delta(n_1) = c), \ldots, \neg(\Delta(n_k) = c)\}.$$

Let $n_0 > \max\{n_1, \ldots, n_k\}$. Consider the extension $\mathcal{N}^+$ of $\mathcal{N}$ to $\mathcal{L}^+$ defined by taking $c^{\mathcal{N}^+} := n_0$. Then $\mathcal{N}^+ \models \Gamma_0$.
Applying the Compactness Theorem, we get that Γ has a model

$$\mathcal{A} = (A, +^{\mathcal{A}}, \cdot^{\mathcal{A}}, S^{\mathcal{A}}, 0^{\mathcal{A}}, c^{\mathcal{A}}).$$

It follows that $a := c^{\mathcal{A}}$ is a non-standard element of $\mathcal{A}$. $\square$

SYNTAX

Definition 1.80

The set $LogAx_{\mathcal{L}} \subseteq Form_{\mathcal{L}}$ of logical axioms of $\mathcal{L}$ consists of:

(i) *all tautologies.*

(ii) *formulas of the form*

$$t = t, \quad s = t \rightarrow t = s, \quad s = t \wedge t = u \rightarrow s = u,$$

for any terms s, t, u.

(iii) *formulas of the form*

$$t_1 = u_1 \wedge \ldots \wedge t_m = u_m \rightarrow ft_1 \ldots t_m = fu_1 \ldots u_m,$$

$$t_1 = u_1 \wedge \ldots \wedge t_m = u_m \rightarrow (Rt_1 \ldots t_m \leftrightarrow Ru_1 \ldots u_m),$$

for any $m \geq 1$, $f \in \mathcal{F}_m$, $R \in \mathcal{R}_m$ and any terms s_i, t_i $(i = 1, \ldots, m)$.

(iv) *formulas of the form*

$$\varphi_x(t) \rightarrow \exists x \varphi,$$

where $\varphi_x(t)$ is a free substitution ($\exists$-axioms).

Definition 1.81

The deduction rules (or inference rules) are the following: for any formulas φ, ψ,

(i) *from φ and $\varphi \to \psi$ infer ψ (modus ponens or (MP)):*

$$\frac{\varphi, \ \varphi \to \psi}{\psi}$$

(ii) *if $x \notin FV(\psi)$, then from $\varphi \to \psi$ infer $\exists x\varphi \to \psi$ ($\exists$-introduction):*

$$\frac{\varphi \to \psi}{\exists x\varphi \to \psi} \quad \text{if } x \notin FV(\psi).$$

Syntax

Let Γ be a set of formulas of $\mathcal{L}$.

Definition 1.82

The *Γ-theorems* *of $\mathcal{L}$ are inductively defined as follows:*

(Γ0) *Every logical axiom is a Γ-theorem.*

(Γ1) *Every formula of Γ is a Γ-theorem.*

(Γ2) *Γ is closed under modus ponens: if φ and $\varphi \to \psi$ are Γ-theorems, then ψ is a Γ-theorem.*

(Γ3) *Γ is closed under $\exists$-introduction: if $\varphi \to \psi$ is a Γ-theorem and $x \notin FV(\psi)$, then $\exists x\varphi \to \psi$ is a Γ-theorem.*

(Γ4) *Only the formulas obtained by applying rules (Γ0), (Γ1), (Γ2) and (Γ3) are Γ-theorems.*

If φ is a Γ-theorem, then we also say that φ is deduced from the hypotheses Γ.

Notations: The set of Γ-theorems of $\mathcal{L}$ is denoted by $Thm_{\mathcal{L}}(\Gamma)$.

Notations

$$\begin{array}{lcl} \mathit{Thm}_{\mathcal{L}} & := & \mathit{Thm}_{\mathcal{L}}(\emptyset) \\ \Gamma \vdash_{\mathcal{L}} \varphi & := & \varphi \in \mathit{Thm}_{\mathcal{L}}(\Gamma) \\ \vdash_{\mathcal{L}} \varphi & := & \varphi \in \mathit{Thm}_{\mathcal{L}} \\ \Gamma \vdash_{\mathcal{L}} \Delta & := & \Gamma \vdash_{\mathcal{L}} \varphi \text{ for any } \varphi \in \Delta. \end{array}$$

Definition 1.83

A formula φ is called a (logical) theorem of $\mathcal{L}$ if $\vdash_{\mathcal{L}} \varphi$.

Convention

When $\mathcal{L}$ is clear from the context, we write *LogAx*, *Thm*, *Thm*(Γ), $\Gamma \vdash \varphi$, $\vdash \varphi$, etc..

Lemma 1.84

Let Γ and Δ be sets of formulas. The following hold:

(i) *If $\Gamma \subseteq \Delta$, then $Thm(\Gamma) \subseteq Thm(\Delta)$, that is, for any formula φ,*

$$\Gamma \vdash \varphi \text{ implies } \Delta \vdash \varphi.$$

(ii) *$Thm \subseteq Thm(\Gamma)$, that is, for any formula φ,*

$$\vdash \varphi \text{ implies } \Gamma \vdash \varphi.$$

(iii) *$Thm(Thm(\Gamma)) = Thm(\Gamma)$, that is, for any formula φ,*

$$Thm(\Gamma) \vdash \varphi \text{ iff } \Gamma \vdash \varphi.$$

(iv) *If $\Gamma \vdash \Delta$, then $Thm(\Delta) \subseteq Thm(\Gamma)$, that is, for any formula φ,*

$$\Delta \vdash \varphi \text{ implies } \Gamma \vdash \varphi.$$

Definition 1.85

A Γ-proof (or proof from the hypotheses Γ) of $\mathcal{L}$ is a sequence of formulas $\theta_1, \ldots, \theta_n$ such that for all $i \in \{1, \ldots, n\}$, one of the following holds:

(i) *θ_i is an axiom;*

(ii) *$\theta_i \in \Gamma$;*

(iii) *there exist $k, j < i$ such that*

$$\theta_k = \theta_j \to \theta_i;$$

(iv) *there exists $j < i$ such that*

$$\theta_j = \varphi \to \psi \text{ and } \theta_i = \exists x\varphi \to \psi,$$

where φ, ψ are formulas and $x \notin FV(\psi)$.

A $\emptyset$-proof is called simply a proof.

Notations: The set of Γ-proofs of $\mathcal{L}$ is denoted by $Proof_{\mathcal{L}}(\Gamma)$ and the set of proofs of $\mathcal{L}$ is denoted by $Proof_{\mathcal{L}}$.

Definition 1.86

Let φ be a formula. A Γ-proof of φ or a proof of φ from the hypotheses Γ is a Γ-proof $\theta_1, \ldots, \theta_n$ such that $\theta_n = \varphi$.
If this is the case, n is called the length of the Γ-proof.

Proposition 1.87

Let Γ be a set of formulas. For any formula φ,

$$\Gamma \vdash \varphi \quad \text{iff} \quad \text{there exists a } \Gamma\text{-proof of } \varphi.$$

Proposition 1.88

For any set of formulas Γ and any formula φ,

$$\Gamma \vdash \varphi \textit{ iff there exists a finite subset } \Sigma \textit{ of } \Gamma \textit{ such that } \Sigma \vdash \varphi.$$

Proof: "$\Leftarrow$" Let $\Sigma \subseteq \Gamma$, Σ finite be such that $\Sigma \vdash \varphi$. Since $\Sigma \subseteq \Gamma$, it follows that $\Gamma \vdash \varphi$.
"$\Rightarrow$" Suppose that $\Gamma \vdash \varphi$. By Proposition 1.87, φ has a Γ-proof $\theta_1, \ldots, \theta_n = \varphi$. Let

$$\Sigma := \Gamma \cap \{\theta_1, \ldots, \theta_n\}.$$

Then Σ is finite, $\Sigma \subseteq \Gamma$ and $\theta_1, \ldots, \theta_n = \varphi$ is a Σ-proof of φ, hence $\Sigma \vdash \varphi$. $\square$

Tautology Theorem

Definition 1.89

We say that the formulas φ and ψ are tautologically equivalent if $F(\varphi) = F(\psi)$ for any $\mathcal{L}$-truth assignment F.

Example 1.90

$\varphi_1 \to (\varphi_2 \to \ldots \to (\varphi_n \to \psi)\ldots)$ and $(\varphi_1 \wedge \ldots \wedge \varphi_n) \to \psi$ are tautologically equivalent.

Proposition 1.91

Let $n \geq 1$ and $\varphi_1, \ldots, \varphi_n, \psi$ be formulas. The following are equivalent:

(i) ψ *is a tautological consequence of* $\{\varphi_1, \ldots, \varphi_n\}$.

(ii) $\varphi_1 \to (\varphi_2 \to \ldots \to (\varphi_n \to \psi)\ldots)$ *is a tautology.*

(iii) $(\varphi_1 \wedge \ldots \wedge \varphi_n) \to \psi$ *is a tautology.*

Theorem 1.92 (Tautology Theorem (Post))

If ψ is a tautological consequence of $\{\varphi_1, \ldots, \varphi_n\}$ and $\Gamma \vdash \varphi_1, \ldots, \Gamma \vdash \varphi_n$, then $\Gamma \vdash \psi$.

Proof: By Proposition 1.91, we have that

$$\chi := \varphi_1 \to (\varphi_2 \to \ldots \to (\varphi_n \to \psi) \ldots)$$

is a tautology. As tautologies are axioms of $\mathcal{L}$, it follows that $\Gamma \vdash \chi$. Since, by hypothesis, $\Gamma \vdash \varphi_1$, we can apply (MP) to get that

$$\Gamma \vdash \varphi_2 \to (\varphi_3 \to \ldots \to (\varphi_n \to \psi) \ldots).$$

We continue to apply (MP) until we get that $\Gamma \vdash \psi$. $\square$

Theorem 1.93 (Deduction Theorem)

Let $\Gamma \cup \{\psi\}$ be a set of formulas and φ be a sentence. Then

$$\Gamma \cup \{\varphi\} \vdash \psi \quad \textit{iff} \quad \Gamma \vdash \varphi \to \psi.$$

Proof: "$\Leftarrow$"

(1)	$\Gamma \vdash \varphi \to \psi$	by hypothesis
(2)	$\Gamma \cup \{\varphi\} \vdash \varphi \to \psi$	by Lema 1.84.(i)
(3)	$\Gamma \cup \{\varphi\} \vdash \varphi$	by the definition
(4)	$\Gamma \cup \{\varphi\} \vdash \psi$	(MP): (2), (3).

"$\Rightarrow$" Supplementary exercise. $\square$

Universal closure

Definition 1.94

Let φ be a formula with $FV(\varphi) = \{x_1, \ldots, x_n\}$. The universal closure of φ is the sentence

$$\overline{\forall\varphi} := \forall x_1 \ldots \forall x_n \varphi.$$

Notation If Γ is a set of formulas, $\overline{\forall\Gamma} := \{\overline{\forall\varphi} \mid \varphi \in \Gamma\}$.

Remark 1.95

φ sentence $\implies \overline{\forall\varphi} = \varphi$; Γ set of sentences $\implies \overline{\forall\Gamma} = \Gamma$.

Proposition 1.96

If Γ is a set of sentences, then for any φ,

$$\Gamma \vDash \varphi \quad \iff \quad \Gamma \vDash \overline{\forall\varphi}.$$

Proof: Exercise.

Theorem 1.97 (Soundness Theorem)

For any set of formulas Γ *and any formula* φ,

$$\Gamma \vdash \varphi \quad \textit{implies} \quad \overline{\forall\Gamma} \models \varphi.$$

Corollary 1.98

For any set of sentences Γ *and any formula* φ,

$$\Gamma \vdash \varphi \quad \textit{implies} \quad \Gamma \models \varphi.$$

Consistent sets

Let Γ be a set of formulas of $\mathcal{L}$.

Definition 1.99

Γ is called consistent if there exists a formula φ such that $\Gamma \not\vdash \varphi$.
Γ is said to be inconsistent if it is not consistent, that is $\Gamma \vdash \varphi$ for any formula φ.

Proposition 1.100

The following are equivalent:

(i) *Γ is inconsistent.*

(ii) *For any formula ψ, $\Gamma \vdash \psi$ and $\Gamma \vdash \neg\psi$.*

(iii) *There exists a formula ψ such that $\Gamma \vdash \psi$ and $\Gamma \vdash \neg\psi$.*

Proposition 1.101

Γ is inconsistent iff Γ has a finite inconsistent subset.

Proof: "$\Leftarrow$" Exercise.
"$\Rightarrow$" Suppose that Γ is inconsistent. By Proposition 1.100, there exists ψ such that $\Gamma \vdash \psi$ and $\Gamma \vdash \neg\psi$. Applying Proposition 1.88, we obtain finite subsets Σ_1, Σ_2 of Γ such that $\Sigma_1 \vdash \psi$ and $\Sigma_2 \vdash \neg\psi$. Let $\Sigma := \Sigma_1 \cup \Sigma_2$. Then Σ is a finite subset of Γ and $\Sigma \vdash \psi$ and $\Sigma \vdash \neg\psi$. Applying again Proposition 1.100, it follows that is Σ is inconsistent. □

An equivalent result is the following:

Proposition 1.102

Γ is consistent iff any finite subset of Γ is consistent.

Proposition 1.103

Let Γ be a set of formulas and φ be a sentence.

(i) $\Gamma \vdash \varphi$ *iff* $\Gamma \cup \{\neg\varphi\}$ *is inconsistent.*

(ii) $\Gamma \vdash \neg\varphi$ *iff* $\Gamma \cup \{\varphi\}$ *is inconsistent.*

Proof: (i) "$\Rightarrow$" Assume that $\Gamma \vdash \varphi$. Then $\Gamma \cup \{\neg\varphi\} \vdash \varphi$ and $\Gamma \cup \{\neg\varphi\} \vdash \neg\varphi$. Hence, $\Gamma \cup \{\neg\varphi\}$ is inconsistent.

"$\Leftarrow$"

(1)	$\Gamma \cup \{\neg\varphi\} \vdash \varphi$	$\Gamma \cup \{\neg\varphi\}$ is inconsistent
(2)	$\Gamma \vdash \neg\varphi \to \varphi$	Deduction Theorem
(3)	$\Gamma \vdash (\neg\varphi \to \varphi) \to \varphi$	tautology
(4)	$\Gamma \vdash \varphi$	(MP): (2),(3)

(ii) Exercise. $\square$

Theorem 1.104 (Completeness Theorem (version 1))

Every consistent set of sentences Γ is satisfiable.

Theorem 1.105 (Completeness Theorem (version 2))

For any set of sentences Γ and any sentence φ,

$$\Gamma \vdash \varphi \quad \Longleftrightarrow \quad \Gamma \models \varphi.$$

- The Completeness Theorem was proved by Gödel in 1929 in hid PhD thesis.
- Henkin gave in 1949 a simplified proof.

Proposition 1.106

Completeness Theorem (version 1) implies Completeness Theorem (version 2).

Proof: "$\Rightarrow$" Apply Soundness Theorem 1.97.
"$\Leftarrow$" Assume that $\Gamma \not\vdash \varphi$. Then, by Proposition 1.103.(i), $\Gamma \cup \{\neg\varphi\}$ is consistent. Apply Completeness Theorem (version 1) to get that $\Gamma \cup \{\neg\varphi\}$ has a model $\mathcal{A}$. Since $\mathcal{A} \vDash \Gamma$ and $\Gamma \vDash \varphi$, we obtain that $\mathcal{A}$ is a model of $\Gamma \cup \{\varphi\}$. In particular, $\mathcal{A} \vDash \varphi$ and $\mathcal{A} \vDash \neg\varphi$, so we have got a contradiction. $\square$

One can also prove that Completeness Theorem (version 2) implies Completeness Theorem (version 1). Hence, the two versions are equivalent.

Skolem normal form

Skolemization is a procedure used to eliminate the existential quantifiers from first-order sentences in prenex normal form by introducing new function/constant symbols, called Skolem function/constant symbols.

Let $\mathcal{L}$ be a first-order language and φ a sentence of $\mathcal{L}$ that is in prenex normal form:

$$\varphi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \, \theta,$$

where $n \in \mathbb{N}$, $Q_1, \dots, Q_n \in \{\forall, \exists\}$, $x_1, \dots, x_n$ are pairwise distinct variables and θ is a quantifier-free formula.

Skolem normal form

We associate with φ a quantifier-free or universal sentence φ^{Sk} in an extended language $\mathcal{L}^{Sk}(\varphi)$ as follows:

If φ is quantifier-free or universal, then $\varphi^{Sk} = \varphi$ and $\mathcal{L}^{Sk}(\varphi) = \mathcal{L}$.

Otherwise, φ has one of the forms:

- $\varphi = \exists x\, \psi$. We introduce a new constant symbol c and consider $\varphi^1 = \psi_x(c)$ and $\mathcal{L}^1 = \mathcal{L} \cup \{c\}$.
- $\varphi = \forall x_1 \ldots \forall x_k \exists x\, \psi$ $(k \geq 1)$. We introduce a new function symbol f of arity k and consider $\varphi^1 = \forall x_1 \ldots \forall x_k\, \psi_x(fx_1 \ldots x_k)$ and $\mathcal{L}^1 = \mathcal{L} \cup \{f\}$.

In both cases, φ^1 has one quantifier less than φ.

If φ^1 is quantifier-free or universal, then $\varphi^{Sk} = \varphi^1$. Otherwise, we form $\varphi^2, \varphi^3, \ldots$, until we reach a quantifier-free or universal sentence and this is φ^{Sk}.

φ^{Sk} is a Skolem normal form of φ.

Skolem normal form

Examples

- Let θ be a quantifier-free formula such that $FV(\theta) = \{x\}$ and $\varphi = \exists x\,\theta$. Then $\varphi^1 = \theta_x(c)$, where c is a new constant symbol. Since φ^1 is a quantifier-free sentence, it follows that $\varphi^{Sk} = \varphi^1 = \theta_x(c)$.
- Let R be a relation symbol of arity 3 and $\varphi = \exists x \forall y \forall z\, R(x, y, z)$. Then
$$\varphi^1 = \forall y \forall z\,(R(x, y, z))_x(c) = \forall y \forall z\, R(c, y, z),$$
where c is a new constant symbol. Since φ^1 is a universal sentence, it follows that $\varphi^{Sk} = \varphi^1 = \forall y \forall z\, R(c, y, z)$.
- Let P be a binary relation symbol and $\varphi = \forall y \exists z\, P(y, z)$. Then $\varphi^1 = \forall y\,(P(y, z))_z(f(y)) = \forall y\, P(y, f(y))$, where f is a new unary function symbol. Since φ^1 is a universal sentence, it follows that $\varphi^{Sk} = \varphi^1 = \forall y\, P(y, f(y))$.

Example

Let $\mathcal{L}$ be a first-order language containing a binary relation symbol R and a unary function symbol f. Let

$$\varphi := \forall y \exists z \forall u \exists v \, (R(y,z) \wedge f(u) = v).$$

$$\begin{aligned}
\varphi^1 &= \forall y \forall u \exists v \, (R(y,z) \wedge f(u) = v)_z(g(y)) \\
&= \forall y \forall u \exists v \, (R(y,g(y)) \wedge f(u) = v), \\
&\quad \text{where } g \text{ is a new unary function symbol} \\
\varphi^2 &= \forall y \forall u \, (R(y,g(y)) \wedge f(u) = v)_v(h(y,u)) \\
&= \forall y \forall u \, (R(y,g(y)) \wedge f(u) = h(y,u)), \\
&\quad \text{where } h \text{ is a new binary function symbol.}
\end{aligned}$$

Since φ^2 is a universal sentence, it follows that

$$\varphi^{Sk} = \varphi^2 = \forall y \forall u \, (R(y,g(y)) \wedge f(u) = h(y,u)).$$

Theorem 1.107 (Skolem normal form theorem)

Let φ be a sentence in prenex normal form.

(i) $\models \varphi^{Sk} \to \varphi$, *hence* $\varphi^{Sk} \models \varphi$ *in* $\mathcal{L}^{Sk}(\varphi)$.

(ii) φ *is satisfiable iff* φ^{Sk} *is satisfiable.*

Proof:

(i) We apply the fact that $\models \varphi_x(t) \to \exists x\varphi$, $\models \varphi$ implies $\models \forall x\varphi$ and $\models \forall x(\varphi \to \psi) \to (\forall x\varphi \to \forall x\psi)$ to conclude that $\models \varphi^1 \to \varphi$, $\models \varphi^2 \to \varphi^1$, etc..

(ii) "$\Leftarrow$" Apply (i). "$\Rightarrow$" Supplementary exercise. $\square$

Remark

Generally, φ and φ^{sk} are not logically equivalent as sentences in $\mathcal{L}^{Sk}(\varphi)$.

Proof: Let $\mathcal{L} = (R)$, where R is a binary relation symbol and $\varphi = \forall v_1 \exists v_2 R(v_1, v_2)$. Then $\varphi^{Sk} = \forall v_1 R(v_1, f(v_1))$ (where f is a new unary function symbol) and $\mathcal{L}^{Sk}(\varphi) = (f, R)$.
Let $\mathcal{L}^{Sk}(\varphi)$-structure

$$\mathcal{A} = (\mathbb{Z}, <, f^{\mathcal{A}}), \text{ where } f^{\mathcal{A}}(n) = n - 1 \text{ for all } n \in \mathbb{Z}.$$

Then $\mathcal{A} \models \varphi$, since for any integer $m \in \mathbb{Z}$ there exists an integer $n \in \mathbb{Z}$ such that $m < n$. On the other hand, $\mathcal{A} \not\models \varphi^{Sk}$, since for any $n \in \mathbb{Z}$, we have that $n \geq f^{\mathcal{A}}(n) = n - 1$. □

MODAL LOGICS

Textbook:

P. Blackburn, M. de Rijke, Y. Venema, Modal logic, Cambridge Tracts in Theoretical Computer Science 53, Cambridge University Press, 2001

Definition 2.1

A relational structure is a tuple $\mathcal{F}$ consisting of:

- *a nonempty set W, called the universe (or domain) of $\mathcal{F}$, and*
- *a set of relations on W.*

We assume that every relational structure contains at least one relation. The elements of W are called points, nodes, states, worlds, times, instances or situations.

Example 2.2

A partially ordered set $\mathcal{F} = (W, R)$, where R is a partial order relation on W.

Labeled Transition Systems (LTSs), or more simply, transition systems, are very simple relational structures widely used in computer science.

Definition 2.3

An LTS is a pair $(W, \{R_a \mid a \in A\})$*, where* W *is a nonempty set of states,* A *is a nonempty set of labels and, for every* $a \in A$*,*

$$R_a \subseteq W \times W$$

is a binary relation on W*.*

LTSs can be viewed as an abstract model of computation: the states are the possible states of a computer, the labels stand for programs, and $(u, v) \in R_a$ means that there is an execution of the program a starting in state u and terminating in state v.

Relational structures

Let W be a nonempty set and $R \subseteq W \times W$ be a binary relation.

We write usually Rwv instead of $(w, v) \in R$. If Rwv, then we say that v is R-accessible from w.

The inverse of R, denoted by R^{-1}, is defined as follows:

$$R^{-1}vw \quad \text{iff} \quad Rwv.$$

We define $R^n (n \geq 0)$ inductively:

$$R^0 = \{(w, w) \mid w \in R\}, \quad R^1 = R, \quad R^{n+1} = R \circ R^n.$$

Thus, for any $n \geq 2$, we have that $R^n wv$ iff there exists $u_1, \ldots, u_{n-1}$ such that $Rwu_1, Ru_1 u_2, \ldots, Ru_{n-1} v$.

BASIC MODAL LOGIC

Definition 2.4

The basic modal language ML_0 consists of:

- *a set $PROP$ of atomic propositions (denoted $p, q, r, v, \ldots$);*
- *the propositional connectives: $\neg$, $\rightarrow$;*
- *the propositional constant $\perp$ (false);*
- *parantheses: (,);*
- *the modal operator $\Diamond$ (diamond).*

The set $Sym(ML_0)$ of symbols of ML_0 is

$$Sym(ML_0) := PROP \cup \{\neg, \rightarrow, \perp, (,), \Diamond\}.$$

The expressions of ML_0 are the finite sequences of symbols of ML_0.

Basic modal language

Definition 2.5

The formulas of the basic modal language ML_0 are the expressions inductively defined as follows:

(F0) *Every atomic proposition is a formula.*

(F1) $\perp$ *is a formula.*

(F2) *If φ is a formula, then $(\neg\varphi)$ is a formula.*

(F3) *If φ and ψ are formulas, then $(\varphi \to \psi)$ is a formula.*

(F4) *If φ is a formula, then $(\Diamond\varphi)$ is a formula.*

(F5) *Only the expressions obtained by applying rules (F0), (F1), (F2), (F3), (F4) are formulas.*

Notation: The set of formulas is denoted by $Form(ML_0)$.

Remark

Formulas of ML_0 are defined, using the Backus-Naur notation, as follows:

$$\varphi ::= p \mid \bot \mid (\neg\varphi) \mid (\varphi \to \psi) \mid (\Diamond\varphi), \quad \text{where } p \in PROP.$$

Unique readability

If φ is a formula, then exactly one of the following holds:

- $\varphi = p$, where p is an atomic proposition;
- $\varphi = \bot$;
- $\varphi = (\neg\psi)$, where ψ is a formula;
- $\varphi = (\psi \to \chi)$, where ψ, χ are formulas;
- $\varphi = (\Diamond\psi)$, where ψ is a formula.

Furthermore, φ can be written in a unique way in one of these forms.

Derived connectives

Connectives $\vee$, $\wedge$, $\leftrightarrow$ and the constant $\top$ (true) are introduced as in classical propositional logic:

$$\begin{array}{ll} \varphi \vee \psi := ((\neg\varphi) \to \psi) & \varphi \wedge \psi := \neg(\varphi \to (\neg\psi)) \\ \varphi \leftrightarrow \psi := ((\varphi \to \psi) \wedge (\psi \to \varphi)) & \top := \neg\bot. \end{array}$$

Dual modal operator

The dual of $\Diamond$ is denoted by $\Box$ (box) and is defined as:

$$\Box\varphi := \neg\Diamond\neg\varphi$$

for every formula φ.

Basic modal language

Usually the external parantheses are omitted, we write them only when necessary. We write $\neg\varphi, \varphi \to \psi$, $\Diamond\varphi$.

To reduce the use of parentheses, we assume that

- modal operators $\Diamond$ and $\Box$ have higher precedence than the other connectives.
- $\neg$ has higher precedence than $\to, \wedge, \vee, \leftrightarrow$;
- $\wedge, \vee$ have higher precedence than $\to, \leftrightarrow$.

Basic modal language

Three readings of the modal operators $\Diamond$ and $\Box$ have been extremely influential.

Classical modal logic

In classical modal logic, $\Diamond\varphi$ is read as it is possible the case that φ. Then $\Box\varphi$ means it is not possible that not φ, that is necessarily φ.

Examples of formulas we would probably regard as correct principles

- $\Box\varphi \to \Diamond\varphi$ (whatever is necessary is possible)
- $\varphi \to \Diamond\varphi$ (whatever is, is possible).

The status of other formulas is harder to decide. What can we say about $\varphi \to \Box\Diamond\varphi$ (whatever is, is necessarily possible) or $\Diamond\varphi \to \Box\Diamond\varphi$ (whatever is possible, is necessarily possible)? Can we consider them as general truths? In order to give an answer to such questions, one has to define a semantics for the classical modal logic

Epistemic logic

In epistemic logic, the basic modal language is used to reason about knowledge. In this logic,

$\Box\varphi$ is read as the agent knows that φ.

We write $K\varphi$ instead of $\Box\varphi$.

As we are talking about knowledge, it is natural to consider to be true the formula

$K\varphi \to \varphi$ (if the agent knows that φ, then φ must hold)

If we assume that the agent is not omniscient, then the formula $\varphi \to K\varphi$ must be false.

Provability logic

In this logic,

$\Box\varphi$ is read as it is provable (in some arithmetical theory) that φ.

A central theme in provability logic is the search for a complete axiomatization of the provability principles that are valid for various arithmetical theories (such as Peano Arithmetic).

An important formula in this context is the Löb formula:

$$\Box(\Box p \to p) \to \Box p$$

Definition 2.6

A substitution *is a mapping* $\sigma : PROP \to Form(ML_0)$.

Such a substitution σ induces a mapping

$$(\cdot)^\sigma : Form(ML_0) \to Form(ML_0)$$

which we can recursively define as follows:

$$\begin{aligned}
p^\sigma &= \sigma(p) \\
\perp^\sigma &= \perp \\
(\neg\varphi)^\sigma &= \neg\varphi^\sigma \\
(\psi \to \varphi)^\sigma &= \psi^\sigma \to \varphi^\sigma \\
(\Diamond\varphi)^\sigma &= \Diamond\varphi^\sigma.
\end{aligned}$$

This definition formalizes what is meant by carrying out uniform substitution.

Substitution

One gets immediately that

- $\top^\sigma = \top$, $(\psi \wedge \varphi)^\sigma = \psi^\sigma \wedge \varphi^\sigma$, $(\psi \vee \varphi)^\sigma = \psi^\sigma \vee \varphi^\sigma$ and $(\psi \leftrightarrow \varphi)^\sigma = \psi^\sigma \leftrightarrow \varphi^\sigma$
- $(\Box\varphi)^\sigma = \Box\varphi^\sigma$.

Definition 2.7

We say that ψ is a substitution instance of φ if there is some substitution σ such that $\varphi^\sigma = \psi$.

Example 2.8

Consider the substitution σ defined as follows:

$$\sigma(p) = p \wedge \Box q,\ \sigma(q) = \Diamond\Diamond q \vee r,\ \sigma(v) = v \text{ if } v \in PROP \setminus \{p, q\}.$$

Then

$$(p \wedge q \wedge r)^\sigma = p \wedge \Box q \wedge (\Diamond\Diamond q \vee r) \wedge r.$$

SEMANTICS

Frames and models

In the sequel we give the semantics of modal languages by interpreting them in relational structures.
We will do this in two distinct ways:

- at the level of models, where the fundamental notion of satisfaction (or truth) is defined.
- at the level of frames, where the key notion of validity is defined.

Definition 2.9

A frame for ML_0 is a pair $\mathcal{F} = (W, R)$ such that

- *W is a nonempty set;*
- *R is a binary relation on W.*

That is, a frame for the basic modal language is simply a relational structure with a single binary relation.

Frames and models

Definition 2.10

A model *for ML_0 is a pair $\mathcal{M} = (\mathcal{F}, V)$, where*

- *$\mathcal{F} = (W, R)$ is a frame for ML_0;*
- *$V : PROP \to 2^W$ is a function called* valuation.

Thus, V assigns to each atomic proposition $p \in PROP$ a subset $V(p)$ of W. Informally, we think of $V(p)$ as the set of points in the model $\mathcal{M}$ where p is true.
Note that models for ML_0 can also be viewed as relational structures in a natural way:

$$\mathcal{M} = (W, R, \{V(p) \mid p \in PROP\}).$$

Thus, a model is a relational structure consisting of a domain, a single binary relation R and the unary relations $V(p), p \in PROP$.
A frame $\mathcal{F}$ and a model $\mathcal{M}$ are two relational structures based on the same universe. However, as we shall see, frames and models are used very differently.

Frames and models

Let $\mathcal{F} = (W, R)$ be a frame and $\mathcal{M} = (\mathcal{F}, V)$ be a model. We also write $\mathcal{M} = (W, R, V)$.

We say that the model $\mathcal{M} = (\mathcal{F}, V)$ is based on the frame $\mathcal{F} = (W, R)$ or that $\mathcal{F}$ is the frame underlying $\mathcal{M}$. Elements of W are called states in $\mathcal{F}$ or in $\mathcal{M}$. We often write $w \in \mathcal{F}$ or $w \in \mathcal{M}$.

Remark

Elements of W are also called worlds or possible worlds, having as inspiration Leibniz's philosophy and the reading of basic modal language in which

$$\Diamond\varphi \text{ means possibly } \varphi \text{ and } \Box\varphi \text{ means necessarily } \varphi.$$

In Leibniz's view, necessity means truth in all possible worlds and possibility means truth in some possible world.

We define now the notion of satisfaction.

Definition 2.11

Let $\mathcal{M} = (W, R, V)$ be a model and w a state in $\mathcal{M}$. We define inductively the notion

formula φ is satisfied (or true) in $\mathcal{M}$ at state w,
Notation $\mathcal{M}, w \Vdash \varphi$

$$
\begin{array}{rcl}
\mathcal{M}, w \Vdash p & \text{iff} & w \in V(p), \quad \text{where } p \in PROP \\
\mathcal{M}, w \Vdash \bot & & \text{never} \\
\mathcal{M}, w \Vdash \neg\varphi & \text{iff} & \text{it is not true that } \mathcal{M}, w \Vdash \varphi \\
\mathcal{M}, w \Vdash \varphi \to \psi & \text{iff} & \mathcal{M}, w \Vdash \varphi \text{ implies } \mathcal{M}, w \Vdash \psi \\
\mathcal{M}, w \Vdash \Diamond\varphi & \text{iff} & \text{there exists } v \in W \text{ such that} \\
 & & Rwv \text{ and } \mathcal{M}, v \Vdash \varphi.
\end{array}
$$

Let $\mathcal{M} = (W, R, V)$ be a model.

Notation

If $\mathcal{M}$ does not satisfy φ at w, we write $\mathcal{M}, w \nVdash \varphi$ and we say that φ is false in $\mathcal{M}$ at state w.

It follows from Definition 2.11 that for every state $w \in W$,

- $\mathcal{M}, w \nVdash \bot$
- $\mathcal{M}, w \Vdash \neg\varphi$ iff $\mathcal{M}, w \nVdash \varphi$.

Notation

We can extend the valuation V from atomic propositions to arbitrary formulas φ so that $V(\varphi)$ is the set of all states in $\mathcal{M}$ at which φ is true:

$$V(\varphi) = \{w \mid \mathcal{M}, w \Vdash \varphi\}.$$

Let $\mathcal{M} = (W, R, V)$ be a model and w a state in $\mathcal{M}$.

Proposition 2.12

For every formulas φ, ψ,

$$\begin{array}{lll} \mathcal{M}, w \Vdash \varphi \vee \psi & \text{iff} & \mathcal{M}, w \Vdash \varphi \text{ or } \mathcal{M}, w \Vdash \psi \\ \mathcal{M}, w \Vdash \varphi \wedge \psi & \text{iff} & \mathcal{M}, w \Vdash \varphi \text{ and } \mathcal{M}, w \Vdash \psi \end{array}$$

Proof: Exercise.

Proposition 2.13

For every formula φ,

$$\mathcal{M}, w \Vdash \Box\varphi \quad \text{iff} \quad \text{for every } v \in W, Rwv \text{ implies } \mathcal{M}, v \Vdash \varphi.$$

Proof: Exercise.

Frames and models

Let $\mathcal{M} = (W, R, V)$ be a model and w a state in $\mathcal{M}$.

Proposition 2.14

For every $n \geq 1$ and every formula φ, define

$$\Diamond^n \varphi := \underbrace{\Diamond\Diamond \dots \Diamond}_{n \text{ times}} \varphi, \qquad \Box^n \varphi := \underbrace{\Box\Box \dots \Box}_{n \text{ times}} \varphi.$$

Then

$$\begin{array}{lll} \mathcal{M}, w \Vdash \Diamond^n \varphi & \text{iff} & \text{there exists } v \in V \text{ s.t. } R^n wv \text{ and } \mathcal{M}, v \Vdash \varphi \\ \mathcal{M}, w \Vdash \Box^n \varphi & \text{iff} & \text{for every } v \in V, R^n wv \text{ implies } \mathcal{M}, v \Vdash \varphi. \end{array}$$

Proof: Exercise.

Frames and models

Let $\mathcal{M} = (W, R, V)$ be a model.

Definition 2.15

- *A formula φ is globally true or simply true in $\mathcal{M}$ if $\mathcal{M}, w \Vdash \varphi$ for every $w \in W$. Notation: $\mathcal{M} \Vdash \varphi$*
- *A formula φ is satisfiable in $\mathcal{M}$ if there exists a state $w \in W$ such that $\mathcal{M}, w \Vdash \varphi$.*

Definition 2.16

Let Σ be a set of formulas.

- *Σ is true at state w in $\mathcal{M}$ if $\mathcal{M}, w \Vdash \varphi$ for every $\varphi \in \Sigma$. Notation: $\mathcal{M}, w \Vdash \Sigma$*
- *Σ is globally true or simply true in $\mathcal{M}$ if $\mathcal{M}, w \Vdash \Sigma$ for every state w in $\mathcal{M}$. Notation: $\mathcal{M} \Vdash \Sigma$*
- *Σ is satisfiable in $\mathcal{M}$ if there exists a state $w \in W$ such that $\mathcal{M}, w \Vdash \Sigma$.*

Frames and models

Example 2.17

Consider the frame $\mathcal{F} = (W = \{w_1, w_2, w_3, w_4, w_5\}, R)$, where Rw_iw_j iff $j = i + 1$:

$$\begin{tikzcd} w_1 \arrow[r] & w_2 \arrow[r] & w_3 \arrow[r] & w_4 \arrow[r] & w_5 \end{tikzcd}$$

Let us choose a valuation V such that $V(p) = \{w_2, w_3\}$, $V(q) = \{w_1, w_2, w_3, w_4, w_5\}$ and $V(r) = \emptyset$.
Consider the model $\mathcal{M} = (\mathcal{F}, V)$. Then

(i) $\mathcal{M}, w_1 \Vdash \Diamond\Box p$

(ii) $\mathcal{M}, w_1 \nVdash \Diamond\Box p \to p$

(iii) $\mathcal{M}, w_2 \Vdash \Diamond(p \wedge \neg r)$

(iv) $\mathcal{M}, w_1 \Vdash q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond q)))$

(v) $\mathcal{M} \Vdash \Box q$.

Proof: Exercise.

Frames and models

The notion of satisfaction is internal and local. We evaluate formulas inside models, at some particular state w (the current state). Modal operators $\Diamond, \Box$ work locally: we verify the truth of φ only in the states that are R-accesibile from the current one.

At first sight this may seem a weakness of the satisfaction definition. In fact, it is its greatest source of strength, as it gives us great flexibility.
For example, if we take $R = W \times W$, then all states are accessible from the current state; this corresponds to the Leibnizian idea in its purest form.
Going to the other extreme, if we take $R = \{(v, v) \mid v \in W\}$, then no state has access to any other.
Between these extremes there is a wide range of options to explore.

Frames and models

We can ask ourselves the following natural questions:

- What happens if we impose some conditions on R (for example, reflexivity, symmetry, transitivity, etc.)?
- What is the impact of these conditions on the notions of necessity and possibility?
- What principles or rules are justified by these conditions?

Frames and validity

Validity in a frame is one of the key concepts in modal logic.

Definition 2.18

Let $\mathcal{F}$ be a frame and φ be a formula.

- *φ is valid at a state w in $\mathcal{F}$ if φ is true at w in every model $\mathcal{M} = (\mathcal{F}, V)$ based on $\mathcal{F}$.*
- *φ is valid in $\mathcal{F}$ if it is valid at every state w in $\mathcal{F}$.*
  Notation: $\mathcal{F} \Vdash \varphi$

Hence, a formula is valid in a frame if it is true at every state in every model based on the frame.

Frames and validity

Validity in a frame differs in an essential way from the truth in a model. Let us give a simple example.

Example 2.19

If $\varphi \vee \psi$ is true in a model $\mathcal{M}$ at w, then φ is true in $\mathcal{M}$ at w or ψ is true in $\mathcal{M}$ at w (by Proposition 2.117).

On the other hand, if $\varphi \vee \psi$ is valid in a frame $\mathcal{F}$ at w, it does not follow that φ is valid in $\mathcal{F}$ at w or ψ is valid in $\mathcal{F}$ at w ($p \vee \neg p$ is a counterexample).

Frames and validity

Definition 2.20

Let $\boldsymbol{M}$ be a class of models, $\boldsymbol{F}$ be a class of frames and φ be a formula. We say that

- *φ is true in $\boldsymbol{M}$ if it is true in every model in $\boldsymbol{M}$.*
  Notation: $\boldsymbol{M} \Vdash \varphi$
- *φ is valid in $\boldsymbol{F}$ if it is valid in every frame in $\boldsymbol{F}$.*
  Notation: $\boldsymbol{F} \Vdash \varphi$

Definition 2.21

The set of all formulas of ML_0 that are valid in a class of frames $\boldsymbol{F}$ is called the logic of $\boldsymbol{F}$ and is denoted by $\Lambda_{\boldsymbol{F}}$.

Example 2.22

Formulas $\Diamond(p \vee q) \to (\Diamond p \vee \Diamond q)$ and $\Box(p \to q) \to (\Box p \to \Box q)$ are valid in the class of all frames.

Proof: Let $\mathcal{F} = (W, R)$ be an arbitrary frame, w a state in $\mathcal{F}$ and $\mathcal{M} = (\mathcal{F}, V)$ be a model based on $\mathcal{F}$. We have to show that

$$\mathcal{M}, w \Vdash \Diamond(p \vee q) \to (\Diamond p \vee \Diamond q).$$

Suppose that $\mathcal{M}, w \Vdash \Diamond(p \vee q)$. Then there exists $v \in W$ such that Rwv and $\mathcal{M}, v \Vdash p \vee q$. We have two cases:

- $\mathcal{M}, v \Vdash p$. Then $\mathcal{M}, w \Vdash \Diamond p$, so $\mathcal{M}, w \Vdash \Diamond p \vee \Diamond q$.
- $\mathcal{M}, v \Vdash q$. Then $\mathcal{M}, w \Vdash \Diamond q$, so $\mathcal{M}, w \Vdash \Diamond p \vee \Diamond q$.

We let as an exercise to prove that $\Box(p \to q) \to (\Box p \to \Box q)$ is valid in the class of all frames. $\Box$

Example 2.23

Formula $\Diamond\Diamond p \to \Diamond p$ is not valid in the class of all frames.

Proof: We have to find a frame $\mathcal{F} = (W, R)$, a state w in $\mathcal{F}$ and a model $\mathcal{M} = (\mathcal{F}, V)$ such that

$$\mathcal{M}, w \nVdash \Diamond\Diamond p \to \Diamond p.$$

Consider the following frame: $\mathcal{F} = (W, R)$, where

$$W = \{0, 1, 2\}, \quad R = \{(0, 1), (1, 2)\}$$

and take a valuation V such that $V(p) = \{2\}$. Then $\mathcal{M}, 0 \Vdash \Diamond\Diamond p$, since $R^2 02$ and $\mathcal{M}, 2 \Vdash p$.
On the other hand, $\mathcal{M}, 0 \nVdash \Diamond p$, as 1 is the only state R-accesible from 0 and $\mathcal{M}, 1 \nVdash p$. $\square$

Definition 2.24

We say that a frame $\mathcal{F} = (W, R)$ is transitive if R is transitive.

Example 2.25

Formula $\Diamond\Diamond p \to \Diamond p$ is valid in the class of all transitive frames.

Proof: Let $\mathcal{F} = (W, R)$ be a transitive frame, w a state in $\mathcal{F}$ and $\mathcal{M} = (\mathcal{F}, V)$ be a model based on $\mathcal{F}$. Assume that $\mathcal{M}, w \Vdash \Diamond\Diamond p$. Then there exist $u, v \in W$ such that Rwu, Ruv and $\mathcal{M}, v \Vdash p$. Since R is transitive, it follows that Rwv and $\mathcal{M}, v \Vdash p$. Thus, $\mathcal{M}, w \Vdash \Diamond p$. $\square$

Modal consequences

We introduce two families of consequence relations: a local one and a global one. Both families are defined semantically; that is, in terms of classes of structures.
The basic ideas are the following;

- A relation of semantic consequence holds when the truth of the premises guarantees the truth of the conclusion.
- The inferences depend on the class of structures we are working with. (For example, inferences for transitive frames must be different than the ones for intransitive frames.)

Thus, the definition of the consequence relation must make reference to a class of structures $\boldsymbol{S}$.

Modal consequences

Let $\boldsymbol{S}$ be a class of structures (frames or models) for ML_0.
If $\boldsymbol{S}$ is a class of models, then a model from $\boldsymbol{S}$ is simply an element $\mathcal{M}$ of $\boldsymbol{S}$. If $\boldsymbol{S}$ is a class of frames, then a model from $\boldsymbol{S}$ is a model based on a frame in $\boldsymbol{S}$.

Definition 2.26 (Local semantic consequence)

Let Σ be a set of formulas and φ be a formula. We say that φ is a local semantic consequence of Σ over $\boldsymbol{S}$ if for all models $\mathcal{M}$ from $\boldsymbol{S}$ and all states w in $\mathcal{M}$,

$$\mathcal{M}, w \Vdash \Sigma \quad \text{implies} \quad \mathcal{M}, w \Vdash \varphi.$$

Notation: $\Sigma \Vdash_{\boldsymbol{S}} \varphi$

Thus, if Σ is true at a state of the model, then φ must be true at the same state.

Remark 2.27

$$\{\psi\} \Vdash_{\mathbf{S}} \varphi \text{ iff } \mathbf{S} \Vdash \psi \to \varphi.$$

Example 2.28

Let *Tran* be the class of transitive frames. Then

$$\{\Diamond\Diamond p\} \Vdash_{Tran} \Diamond p.$$

But $\Diamond p$ is NOT a local semantic consequence of $\Diamond\Diamond p$ over the class of all frames.

Modal consequences

We can define another notion of semantic consequence.

Definition 2.29 (Global semantic consequence)

Let Σ be a set of formulas and φ be a formula. We say that φ is a global semantic consequence of Σ over $\boldsymbol{S}$ *if for all structures $\mathcal{S}$ from* $\boldsymbol{S}$,

$$\mathcal{S} \Vdash \Sigma \quad \textit{implies} \quad \mathcal{S} \Vdash \varphi.$$

Notation: $\Sigma \Vdash^{g}_{\boldsymbol{S}} \varphi$

Here, depending on $\boldsymbol{S}$, $\Vdash$ means validity in a frame or truth in a model.

Modal consequences

The local and global consequence relations are different.

Example 2.30

Let $\boldsymbol{F}$ be the class of all frames. Then

- $\Box p$ is not a local semantic consequence of p over $\boldsymbol{F}$.
- $\Box p$ is a global semantic consequence of p over $\boldsymbol{F}$.

Proof:

- Let $\mathcal{M} = (W, R, V)$, where $W = \{w_1, w_2\}, R = W \times W$, $V(p) = \{w_1\}, V(q)$ arbitrary for $q \neq p$. Then $\mathcal{M}, w_1 \Vdash p$, but $\mathcal{M}, w_1 \nVdash \Box p$, since Rw_1w_2 and $\mathcal{M}, w_2 \nVdash p$.
- Let $\mathcal{F} = (W, R)$ be a frame such that $\mathcal{F} \Vdash p$. We must show that $\mathcal{F} \Vdash \Box p$, that is: for any model $\mathcal{M}$ based on $\mathcal{F}$ and for any state w in $\mathcal{M}$,

  $$\text{for every } v \in W, Rwv \text{ implies } \mathcal{M}, v \Vdash p.$$

  Let $v \in W$ be such that Rwv. Since $\mathcal{F} \Vdash p$, we have that $\mathcal{M} \Vdash p$, so $\mathcal{M}, v \Vdash p$. $\Box$

SYNTAX

Let ML_0 be the basic modal language.

Definition 2.31

A normal modal logic *is a set Λ of formulas of ML_0 satisfying the following properties:*

- *Λ contains the following* axioms*:*

$$
\begin{aligned}
(Taut) &\quad \text{all propositional tautologies}, \\
(K) &\quad \Box(p \to q) \to (\Box p \to \Box q), \\
(Dual) &\quad \Diamond p \leftrightarrow \neg\Box\neg p,
\end{aligned}
$$

where p, q are atomic propositions of ML_0.

Definition 2.31 (continued)

- Λ is closed under the following deduction rules:
  - modus ponens (MP):
    $$\frac{\varphi,\ \varphi \to \psi}{\psi}$$
    Hence, if $\varphi \in \Lambda$ and $\varphi \to \psi \in \Lambda$, then $\psi \in \Lambda$.
  - uniform substitution:
    $$\frac{\varphi}{\theta} \quad \text{where } \theta \text{ is a substitution instance of } \varphi$$
    Hence, if $\varphi \in \Lambda$, then $\theta \in \Lambda$.
  - generalization or necessitation:
    $$\frac{\varphi}{\Box\varphi}$$
    Hence, if $\varphi \in \Lambda$, then $\Box\varphi \in \Lambda$.

Lemma 2.32

Any normal modal logic Λ contains, for any formulas φ, ψ,

(K') $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$,

(Dual') $\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$.

Proof: Let p, q be atomic propositions and let σ be the substitution defined by

$$\sigma(p) = \varphi, \quad \sigma(q) = \psi, \quad \sigma(v) = v \text{ if } v \in PROP \setminus \{p, q\}.$$

Then $(K') = (K)^\sigma$ and $(Dual') = (Dual)^\sigma$. Since $(K), (Dual) \in \Lambda$ and Λ is closed under uniform substitution, it follows that $(K') \in \Lambda$ and $(Dual') \in \Lambda$. $\square$

We write (K) instead of (K') and $(Dual)$ instead of $(Dual')$.

Normal modal logics - tautologies

We add all propositional tautologies as axioms for simplicity, it is not necessary. We could add a small number of tautologies, which generates all of them. For example,

$$\begin{array}{ll} (A1) & \varphi \to (\psi \to \varphi) \\ (A2) & (\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)) \\ (A3) & (\neg\psi \to \neg\varphi) \to (\varphi \to \psi). \end{array}$$

Proposition 2.33

Any propositional tautology is valid in the class of all frames for ML_0.

Remark 2.34

Tautologies may contain modalities, too. For example, $\Diamond\psi \vee \neg\Diamond\psi$ is a tautology, since it has the same form as $\varphi \vee \neg\varphi$.

Normal modal logics - axiom (K)

Axiom (K) is sometimes called the distribution axiom and it is important because it allows us to transform $\Box(\varphi \rightarrow \psi)$ (a boxed formula) in an implication $\Box\varphi \rightarrow \Box\psi$, enabling further pure propositional reasoning to take place.
For example, assume that we want to prove $\Box\psi$ and we already have a proof that contains both $\Box(\varphi \rightarrow \psi)$ and $\Box\varphi$. Applying (K) and modus ponens, we get $\Box\varphi \rightarrow \Box\psi$. Applying again modus ponens, we obtain $\Box\psi$.

By Example 2.22,

Proposition 2.35

(K) *is valid in the class of all frames for* ML_0.

Normal modal logics - axiom (Dual)

Axiom (*Dual*) reflects the duality between $\Diamond$ and $\Box$. It is necessary because in ML_0 the primitive modal operator is $\Diamond$ and $\Box$ is a derived one. Hence, axiom (K) is an abbreviation for

$$\neg\Diamond\neg(\varphi \to \psi) \to (\neg\Diamond\neg\varphi \to \neg\Diamond\neg\psi).$$

If we had considered $\Box$ as our primitive modal operator, then (*Dual*) would not have been required.

Proposition 2.36

(*Dual*) *is valid in the class of all frames for* ML_0.

Proof: Exercise.

Normal modal logics - modus ponens

Proposition 2.37

- *modus ponens preserves satisfiability: for any model $\mathcal{M}$ and any state $w \in \mathcal{M}$,*

$$\text{if } \mathcal{M}, w \Vdash \varphi \to \psi \text{ and } \mathcal{M}, w \Vdash \varphi, \text{ then } \mathcal{M}, w \Vdash \psi.$$

- *modus ponens preserves truth: for any model $\mathcal{M}$,*

$$\text{if } \mathcal{M} \Vdash \varphi \to \psi \text{ and } \mathcal{M} \Vdash \varphi, \text{ then } \mathcal{M} \Vdash \psi.$$

- *modus ponens preserves validity: for any frame $\mathcal{F}$,*

$$\text{if } \mathcal{F} \Vdash \varphi \to \psi \text{ and } \mathcal{F} \Vdash \varphi, \text{ then } \mathcal{F} \Vdash \psi.$$

Proof: Easy exercise.

Proposition 2.38

Uniform substitution preserves validity: for any frame $\mathcal{F}$, if θ is a substitution instance of φ, then

$$\mathcal{F} \Vdash \varphi \quad \textit{implies} \quad \mathcal{F} \Vdash \theta.$$

Proof: Exercise.

Remark 2.39

Uniform substitution does NOT preserve satisfiability or truth.

Proof: Let p, q be distinct atomic propositions. Then q is a substitution instance of p, but from the fact that p is satisfiable/true in a model $\mathcal{M}$, we do not get that q is satisfiable/true in $\mathcal{M}$.

Normal modal logics - generalization

Generalization "modalizes" formulas by adding $\Box$ in front.

Proposition 2.40

- *Generalization preserves truth: for any model $\mathcal{M}$,*

$$\mathcal{M} \Vdash \varphi \quad \textit{implies} \quad \mathcal{M} \Vdash \Box\varphi.$$

- *Generalization preserves validity: for any frame $\mathcal{F}$,*

$$\mathcal{F} \Vdash \varphi \quad \textit{implies} \quad \mathcal{F} \Vdash \Box\varphi.$$

Proof: Exercise.

Remark 2.41

Generalization does NOT preserve satisfiability.

Normal modal logics

Theorem 2.42

For any class $\mathbf{F}$ of frames, $\Lambda_{\mathbf{F}}$, the logic of $\mathbf{F}$, is a normal modal logic.

Proof: It follows from the previous results.

Lemma 2.43

- *The collection of all formulas is a normal modal logic, called the inconsistent logic.*
- *If $\{\Lambda_i \mid i \in I\}$ is a collection of normal modal logics, then $\bigcap_{i \in I} \Lambda_i$ is a normal modal logic.*

Definition 2.44

$\mathbf{K}$ is the intersection of all normal modal logics.

Hence, $\mathbf{K}$ is the least normal modal logic.

Definition 2.45

A ***K**-proof is a sequence of formulas $\theta_1, \ldots, \theta_n$ such that for any $i \in \{1, \ldots, n\}$, one of the following conditions is satisfied:*

- *θ_i is an axiom (that is, a tautology, (K) or $(Dual)$);*
- *θ_i is obtained from previous formulas by applying one of the deductions rules modus ponens, uniform substitution or generalization.*

Definition 2.46

Let φ be a formula. A ***K**-proof of φ is a* ***K**-proof $\theta_1, \ldots, \theta_n$ such that $\theta_n = \varphi$.*
If φ has a ***K**-proof, we say that φ is* ***K**-provable.*
Notation: $\vdash_{\boldsymbol{K}} \varphi$.

Example 2.47

For any $p, q \in PROP$, $\vdash_{\boldsymbol{K}} \Box p \wedge \Box q \to \Box(p \wedge q)$.

Proof: We give the following $\boldsymbol{K}$-proof:

(1) $\vdash_{\boldsymbol{K}} p \to (q \to p \wedge q)$ tautology

(2) $\vdash_{\boldsymbol{K}} \Box(p \to (q \to p \wedge q))$ generalization: (1)

(3) $\vdash_{\boldsymbol{K}} \Box(p \to q) \to (\Box p \to \Box q)$ axiom (K)

(4) $\vdash_{\boldsymbol{K}} \Box(p \to (q \to p \wedge q)) \to (\Box p \to \Box(q \to p \wedge q))$
uniform substitution: (3), $q \mapsto (q \to p \wedge q)$

(5) $\vdash_{\boldsymbol{K}} \Box p \to \Box(q \to p \wedge q)$ (MP): (2), (4)

(6) $\vdash_{\boldsymbol{K}} \Box(q \to p \wedge q) \to (\Box q \to \Box(p \wedge q))$
uniform substitution: (3), $p \mapsto q, q \mapsto p \wedge q$

(7) $\vdash_{\boldsymbol{K}} \Box p \to (\Box q \to \Box(p \wedge q))$
propositional reasoning: (5),(6) and (MP),
$(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$ tautology

(8) $\vdash_{\boldsymbol{K}} \Box p \wedge \Box q \to \Box(p \wedge q)$
propositional reasoning:(7) and (MP),
$(\varphi \to (\psi \to \chi)) \leftrightarrow (\varphi \wedge \psi \to \chi)$ tautology. $\Box$

$\boldsymbol{K}$

Theorem 2.48

$$\boldsymbol{K} = \{\varphi \mid \vdash_{\boldsymbol{K}} \varphi\}.$$

The logic $\boldsymbol{K}$ is very weak. If we are interested in transitive frames, we would like a proof system which reflects this. For example, we know that $\Diamond\Diamond p \to \Diamond p$ is valid in the class of all transitive frames, so we would want a proof system that generates this formula. $\boldsymbol{K}$ does not do this, since $\Diamond\Diamond p \to \Diamond p$ is not valid in the class of all frames.

The idea is to extend $\boldsymbol{K}$ with additional axioms.

$\boldsymbol{K}\Gamma$

By Lemma 2.43, for any set Γ of formulas, there exists the least normal modal logic that contains Γ.

Definition 2.49

$\boldsymbol{K}\Gamma$ is the least normal modal logic that contains Γ. We say that $\boldsymbol{K}\Gamma$ is generated by Γ or axiomatized by Γ.

Definition 2.50

A $\boldsymbol{K}\Gamma$-proof is a sequence of formulas $\theta_1, \ldots, \theta_n$ such that for any $i \in \{1, \ldots, n\}$, one of the following conditions is satisfied:

- *θ_i is an axiom (that is, a tautology, (K) or $(Dual)$);*
- *$\theta_i \in \Gamma$;*
- *θ_i is obtained from previous formulas by applying one of the deduction rules modus ponens, uniform substitution or generalization.*

$\boldsymbol{K}\Gamma$

Definition 2.51

Let φ be a formula. A $\boldsymbol{K}\Gamma$-proof of φ is a $\boldsymbol{K}\Gamma$-proof $\theta_1, \ldots, \theta_n$ such that $\theta_n = \varphi$.
If φ has a $\boldsymbol{K}\Gamma$-proof, we say that φ is $\boldsymbol{K}\Gamma$-provable.
Notation: $\vdash_{\boldsymbol{K}\Gamma} \varphi$.

Theorem 2.52

$$\boldsymbol{K}\Gamma = \{\varphi \mid \vdash_{\boldsymbol{K}\Gamma} \varphi\}.$$

Example 2.53

If we extend $\boldsymbol{K}$ by adding $\Diamond\Diamond p \to \Diamond p$ as an axiom, we obtain the logic $\boldsymbol{K}4$.

Normal modal logics

In the following, the set $PROP$ of atomic propositions is countable.
Let Λ be a normal modal logic.

Definition 2.54

If $\varphi \in \Lambda$, we also say that φ is a Λ-theorem or a theorem of Λ and write $\vdash_\Lambda \varphi$. If $\varphi \notin \Lambda$, we write $\not\vdash_\Lambda \varphi$.

With these notations, the conditions from the definition of a normal modal logic are written as follows:

For any formulas φ, ψ, θ, the following hold:

(i) If φ is a tautology, then $\vdash_\Lambda \varphi$.

(ii) $\vdash_\Lambda (K)$ and $\vdash_\Lambda (Dual)$.

(iii) If $\vdash_\Lambda \varphi$ and $\vdash_\Lambda \varphi \to \psi$, then $\vdash_\Lambda \psi$.

(iv) If $\vdash_\Lambda \varphi$ and θ is a substitution instance of φ, then $\vdash_\Lambda \theta$.

(v) If $\vdash_\Lambda \varphi$, then $\vdash_\Lambda \Box\varphi$.

Definition 2.55

Let $\psi_1, \ldots, \psi_n, \varphi$ be formulas. We say that φ is deducible in propositional logic from assumptions $\psi_1, \ldots, \psi_n$ if

$$(\psi_1 \wedge \ldots \wedge \psi_n) \to \varphi \text{ is a tautology.}$$

Proposition 2.56

Λ is closed under propositional deduction: if φ is deducible in propositional logic from assumptions $\psi_1, \ldots, \psi_n$, then

$$\vdash_\Lambda \psi_1, \ldots, \vdash_\Lambda \psi_n \quad \text{implies} \quad \vdash_\Lambda \varphi.$$

Definition 2.57

Let $\Gamma \cup \{\varphi\}$ be a set of formulas. We say that φ is deducible in Λ from Γ or that φ is Λ-deducible from Γ if there exist formulas $\psi_1, \ldots, \psi_n \in \Gamma$ $(n \geq 0)$ such that

$$\vdash_\Lambda (\psi_1 \wedge \ldots \wedge \psi_n) \to \varphi.$$

(When $n = 0$, this means that $\vdash_\Lambda \varphi$).

Notation: $\Gamma \vdash_\Lambda \varphi$

We write $\Gamma \not\vdash_\Lambda \varphi$ if φ is not Λ-deducible from Γ.

Remark 2.58

The following are equivalent:

(i) $\Gamma \vdash_\Lambda \varphi$.

(ii) *There exist formulas* $\psi_1, \ldots, \psi_n \in \Gamma$ $(n \geq 0)$ *such that*

$$\vdash_\Lambda \psi_1 \to (\psi_2 \to \ldots \to (\psi_n \to \varphi).$$

Proof: $(i) \Rightarrow (ii)$ There exist formulas $\psi_1, \ldots, \psi_n \in \Gamma$ such that

(1) $\vdash_\Lambda (\psi_1 \wedge \ldots \wedge \psi_n) \to \varphi$ by (i)

(2) $\vdash_\Lambda \big((\psi_1 \wedge \ldots \wedge \psi_n) \to \varphi\big) \to \big(\psi_1 \to (\psi_2 \to \ldots \to (\psi_n \to \varphi))\big)$ tautology

(3) $\vdash_\Lambda \psi_1 \to (\psi_2 \to \ldots \to (\psi_n \to \varphi))$ (MP): (1),(2).

$(ii) \Rightarrow (i)$ There exist formulas $\psi_1, \ldots, \psi_n \in \Gamma$ such that

(1) $\vdash_\Lambda \psi_1 \to (\psi_2 \to \ldots \to (\psi_n \to \varphi))$ by (ii)

(2) $\vdash_\Lambda \big(\psi_1 \to (\psi_2 \to \ldots \to (\psi_n \to \varphi))\big) \to \big((\psi_1 \wedge \ldots \wedge \psi_n) \to \varphi\big)$ tautology

(3) $\vdash_\Lambda (\psi_1 \wedge \ldots \wedge \psi_n) \to \varphi$ (MP): (1),(2). $\square$

Proposition 2.59 (Basic properties)

Let φ be a formula and Γ, Δ be sets of formulas.

(i) $\emptyset \vdash_\Lambda \varphi$ *iff* $\vdash_\Lambda \varphi$.

(ii) $\vdash_\Lambda \varphi$ *implies* $\Gamma \vdash_\Lambda \varphi$.

(iii) $\varphi \in \Gamma$ *implies* $\Gamma \vdash_\Lambda \varphi$.

(iv) *If* $\Gamma \vdash_\Lambda \varphi$ *and* $\Gamma \subseteq \Delta$, *then* $\Delta \vdash_\Lambda \varphi$.

(v) $\Gamma \vdash_\Lambda \varphi$ *iff there exists a finite subset* Σ *of* Γ *such that* $\Sigma \vdash_\Lambda \varphi$.

Proof: Easy exercise.

Proposition 2.60

(i) *If* $\Gamma \vdash_\Lambda \varphi$ *and* ψ *is deducible in propositional logic from* φ*, then* $\Gamma \vdash_\Lambda \psi$*.*

(ii) *If* $\Gamma \vdash_\Lambda \varphi$ *and* $\Gamma \vdash_\Lambda \varphi \to \psi$*, then* $\Gamma \vdash_\Lambda \psi$*.*

(iii) *If* $\Gamma \vdash_\Lambda \varphi$ *and* $\{\varphi\} \vdash_\Lambda \psi$*, then* $\Gamma \vdash_\Lambda \psi$*.*

Proof: Exercise.

Proposition 2.61 (Deduction Theorem)

For any set of formulas Γ *and any formulas* φ, ψ*,*

$$\Gamma \vdash_\Lambda \varphi \to \psi \quad \textit{iff} \quad \Gamma \cup \{\varphi\} \vdash_\Lambda \psi.$$

Proof: Exercise.

Definition 2.62

A set Γ of formulas is called Λ*-consistent if* $\Gamma \nvdash_\Lambda \bot$.
If Γ is not Λ-consistent, we say that Γ is Λ-inconsistent.
A formula φ is Λ-consistent if $\{\varphi\}$ is; otherwise, it is called Λ-inconsistent.

Remark 2.63

Let Γ, Δ be sets of formulas such that $\Gamma \subseteq \Delta$.

(i) *If Δ is Λ-consistent, then Γ is Λ-consistent.*

(ii) *If Γ is Λ-inconsistent, then Δ is Λ-inconsistent.*

Proposition 2.64

Let Γ be a set of formulas. The following are equivalent:

(i) *Γ is Λ-inconsistent.*

(ii) *There exists a formula ψ such that $\Gamma \vdash_\Lambda \psi$ and $\Gamma \vdash_\Lambda \neg\psi$.*

(iii) *$\Gamma \vdash_\Lambda \varphi$ for any formula φ.*

Proof: Exercise.

Proposition 2.65

(i) *$\Gamma \vdash_\Lambda \varphi \iff \Gamma \cup \{\neg\varphi\}$ is Λ-inconsistent.*

(ii) *$\Gamma \vdash_\Lambda \neg\varphi \iff \Gamma \cup \{\varphi\}$ is Λ-inconsistent.*

Proof: Exercise.

Proposition 2.66

Γ is Λ-consistent iff any finite subset of Γ is Λ-consistent.

Proof: Exercise.

Normal logics - soundness

In the following, we say "normal logic " instead of "normal modal logic".

Let $\boldsymbol{S}$ be a class of structures (frames or models) for ML_0.

Notation:

$$\Lambda_{\boldsymbol{S}} := \{\varphi \mid \mathcal{S} \Vdash \varphi \text{ for any structure } \mathcal{S} \text{ from } \boldsymbol{S}\}.$$

Definition 2.67

A normal logic Λ is sound with respect to $\boldsymbol{S}$ if $\Lambda \subseteq \Lambda_{\boldsymbol{S}}$.

Thus, Λ is sound with respect to $\boldsymbol{S}$ iff for any formula φ and for any structure $\mathcal{S}$ in $\boldsymbol{S}$,

$$\vdash_\Lambda \varphi \quad \text{implies} \quad \mathcal{S} \Vdash \varphi.$$

If Λ is sound with respect to $\boldsymbol{S}$, we say also that $\boldsymbol{S}$ is a class of frames (or models) for Λ.

Theorem 2.68 (Soundness theorem for $\boldsymbol{K}$*)*

$\boldsymbol{K}$ *is sound with respect to the class of all frames.*

Proof: We apply Theorem 2.42 and the fact that $\boldsymbol{K}$ is the least normal logic. $\square$.

Definition 2.69

A normal logic Λ is

(i) strongly complete with respect to $\boldsymbol{S}$ if for any set of formulas $\Gamma \cup \{\varphi\}$,

$$\Gamma \Vdash_{\boldsymbol{S}} \varphi \quad \textit{implies} \quad \Gamma \vdash_{\Lambda} \varphi.$$

(ii) weakly complete with respect to $\boldsymbol{S}$ if for any formula φ,

$$\boldsymbol{S} \Vdash \varphi \quad \textit{implies} \quad \vdash_{\Lambda} \varphi.$$

Λ is strongly (weakly) complete with respect to a single structure $\mathcal{S}$ if it is strongly (weakly) complete with respect to the class $\boldsymbol{S} := \{\mathcal{S}\}$.

Obviously, weak completeness is a particular case of strong completeness; just take $\Gamma = \emptyset$ in Definition 2.69.(i). Hence, strong completeness with respect to a class of frames implies weak completeness with respect to that class. The reciprocal is not true.

Remark 2.70

Λ is weakly complete with respect to $\boldsymbol{S}$ iff $\Lambda_{\boldsymbol{S}} \subseteq \Lambda$.

Thus, if if we prove that a normal logic Λ is both sound and weakly complete with respect to a class of structures $\boldsymbol{S}$, we obtain a perfect match between the syntactic and semantic perspectives: $\Lambda = \Lambda_{\boldsymbol{S}}$.

Given a semantically specified normal logic $\Lambda_{\boldsymbol{S}}$ (that is, the logic of some class of structures of interest), a very important problem is to find a simple set of formulas Γ such that $\Lambda_{\boldsymbol{S}}$ is the logic generated by Γ; we say that Γ axiomatizes $\boldsymbol{S}$.

Proposition 2.71

The following are equivalent:

- *(i)* *Λ is strongly complete with respect to $\boldsymbol{S}$.*
- *(ii)* *Any Λ-consistent set of formulas is satisfiable in a model $\mathcal{M}$ from $\boldsymbol{S}$.*

Proof: (i)$\Rightarrow$(ii) Let Γ be a Λ-consistent set. Suppose that Γ is not satisfiable in a model $\mathcal{M}$ from $\boldsymbol{S}$, hence there exists no model $\mathcal{M}$ from $\boldsymbol{S}$ and no state $w \in \mathcal{M}$, such that $\mathcal{M}, w \Vdash \Gamma$. Then $\Gamma \Vdash_{\boldsymbol{S}} \perp$. Since Λ is strongly complete with respect to $\boldsymbol{S}$, it follows that $\Gamma \vdash_{\Lambda} \perp$. We have obtained that Γ is Λ-inconsistent, a contradiction.

$(ii) \Rightarrow (i)$ Let $\Gamma \cup \{\varphi\}$ be a set of formulas such that $\Gamma \Vdash_{\boldsymbol{S}} \varphi$. We remark easily that $\Gamma \cup \{\neg\varphi\}$ is not satisfiable in any model from $\boldsymbol{S}$ (for any model $\mathcal{M}$ from $\boldsymbol{S}$ and any state w in $\mathcal{M}$, if $\mathcal{M}, w \Vdash \Gamma$, then $\mathcal{M}, w \Vdash \varphi$, so $\mathcal{M}, w \nVdash \neg\varphi$). It follows by (ii) that $\Gamma \cup \{\neg\varphi\}$ is Λ-inconsistent. Apply now Proposition 2.65 to conclude that $\Gamma \vdash_\Lambda \varphi$. □

Corollary 2.72

Λ is weakly complete with respect to $\boldsymbol{S}$ iff any Λ-consistent formula is satisfiable in a model $\mathcal{M}$ from $\boldsymbol{S}$.

Proof: Exercise.

Normal logics - completeness

The message of Proposition 2.71 is the following:

completeness theorems are essentially model existence theorems.

We prove the strong completeness of a normal logic Λ with respect to a class of structures by showing that every Λ-consistent set of formulas can be satisfied in some suitable model from that class.

Thus the fundamental question is:

how do we build (suitable) satisfying models?

In the following we give an answer to this question:

we build models using maximal consistent sets of formulas, more precisely canonical models.

Let Λ be a normal logic.

Definition 2.73

A set of formulas Γ is called maximal Λ-consistent if Γ is Λ-consistent and for any set of formulas Δ,

$$\text{if } \Gamma \subseteq \Delta \text{ and } \Delta \text{ is } \Lambda\text{-consistent, then } \Delta = \Gamma.$$

Notation:

We write Λ-MCS instead of "maximal Λ-consistent". When Λ is clear from the context, we write simply MCS.

Proposition 2.74

Let Γ be a Λ-consistent set. The following are equivalent:

(i) *Γ is a Λ-MCS.*

(ii) *For any formula φ, if $\Gamma \cup \{\varphi\}$ is Λ-consistent, then $\varphi \in \Gamma$.*

Proof: Exercise.

Canonical models

Proposition 2.75

The set $Form(ML_0)$ of formulas of ML_0 is countable.

Proposition 2.76 (Lindenbaum's Lemma)

If Γ is a Λ-consistent set of formulas, then there exists a Λ-MCS Γ^+ such that $\Gamma \subseteq \Gamma^+$.

Proof: Let $\varphi_0, \varphi_1, \ldots, \varphi_n, \ldots$ be an enumeration of the formulas of ML_0. We define inductively the following sequence of sets of formulas:

$$\begin{array}{rcl} \Gamma_0 & = & \Gamma, \\ \Gamma_{n+1} & = & \begin{cases} \Gamma_n \cup \{\varphi_n\} & \text{if } \Gamma_n \cup \{\varphi_n\} \text{ is } \Lambda\text{-consistent} \\ \Gamma_n & \text{otherwise} \end{cases} \end{array}$$

By construction, $\Gamma = \Gamma_0 \subseteq \Gamma_1 \subseteq \ldots \Gamma_n \subseteq \ldots$ and, for any $n \in \mathbb{N}$, Γ_n is Λ-consistent.

Let

$$\Gamma^+ = \bigcup_{n\geq 0} \Gamma_n.$$

Claim 1: Γ^+ is Λ-consistent.

Proof of claim: Suppose that Γ^+ is Λ-inconsistent. By Proposition 2.66, Γ^+ has a finite Λ-inconsistent subset $\Delta = \{\psi_1, \psi_2, \ldots, \psi_k\}$. For any $i = 1, \ldots, k$ there exists $N_i \in \mathbb{N}$ such that $\psi_i \in \Gamma_{N_i}$. Let $N := max\{N_1, N_2, \ldots, N_k\}$. Then $\Delta \subseteq \Gamma_N$, hence Γ_N is Λ-inconsistent. We have obtained a contradiction.

Claim 2: Γ^+ is a Λ-MCS.

Proof of claim: We apply Proposition 2.74. Let ψ be a formula such that $\Gamma^+ \cup \{\psi\}$ is Λ-consistent. Let r be such that $\varphi_r = \psi$. Then $\Gamma_r \cup \{\psi\}$ is Λ-consistent, since

$$\Gamma_r \cup \{\psi\} = \Gamma_r \cup \{\varphi_r\} \subseteq \Gamma^+ \cup \{\varphi_r\} = \Gamma^+ \cup \{\psi\}.$$

Hence, $\Gamma_{r+1} = \Gamma_r \cup \{\psi\}$. It follows that $\psi \in \Gamma_{r+1} \subseteq \Gamma^+$. □

Proposition 2.77

Let Γ be a Λ-MCS.

(i) *Γ is closed under modus ponens: if $\varphi \in \Gamma$ and $\varphi \to \psi \in \Gamma$, then $\psi \in \Gamma$.*

(ii) $\Lambda \subseteq \Gamma$.

(iii) *For any formula φ, exactly one of the following holds: $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$ (equivalently, $\varphi \in \Gamma$ iff $\neg\varphi \notin \Gamma$).*

(iv) *For any formula φ,*

$$\varphi \in \Gamma \text{ iff } \Gamma \vdash_\Lambda \varphi.$$

(v) *For any formulas φ, ψ,*

$$\varphi \to \psi \in \Gamma \text{ iff } (\varphi \in \Gamma \text{ implies } \psi \in \Gamma).$$

Proof: Exercise.

Canonical models

With the help of MCSs, we define the special model called canonical model.

Definition 2.78

The canonical model $\mathcal{M}^{\Lambda} = (W^{\Lambda}, R^{\Lambda}, V^{\Lambda})$ *of* Λ *is defined as follows:*

- W^{Λ} *is the set of all* Λ*-MCSs.*
- R^{Λ} *is defined by: for any* $w, v \in W^{\Lambda}$,

  $$R^{\Lambda}wv \text{ iff for any formula } \varphi, \varphi \in v \text{ implies } \Diamond\varphi \in w.$$

  R^{Λ} *is called the canonical relation.*
- V^{Λ} *is defined by: for any* $p \in PROP$,

  $$V^{\Lambda}(p) = \{w \in W^{\Lambda} \mid p \in w\}.$$

  V^{Λ} *is called the canonical valuation.*

$\mathcal{F}^{\Lambda} = (W^{\Lambda}, R^{\Lambda})$ is called the canonical frame for Λ.

Proposition 2.79

For any $w, v \in W^\Lambda$,

$$R^\Lambda wv \text{ iff for any formula } \varphi, \Box\varphi \in w \text{ implies } \varphi \in v.$$

Proof: Exercise.

Proposition 2.80 (Existence Lemma)

Let $w \in W^\Lambda$. *If* φ *is a formula with the property that* $\Diamond\varphi \in w$, *then there exists a state* $v \in W^\Lambda$ *such that* $R^\Lambda wv$ *and* $\varphi \in v$.

Canonical models - Truth Lemma

By the definition of a canonical model, for any atomic proposition p, we have that p is true at a state w in M^Λ iff $p \in w$.
The Truth Lemma extends this equation "truth=membership" to arbitrary formulas.

Proposition 2.81 (Truth Lemma)

Let $w \in W^\Lambda$. For any formula φ,

$$\mathcal{M}^\Lambda, w \Vdash \varphi \quad \textit{iff} \quad \varphi \in w.$$

Proof: By induction on φ.

- $\varphi = p \in PROP$. Then $\mathcal{M}^\Lambda, w \Vdash p$ iff $w \in V^\Lambda(p)$ iff $p \in w$.
- $\varphi = \bot$. Apply the fact that $\mathcal{M}^\Lambda, w \nVdash \bot$ and $\bot \notin w$.
- $\varphi = \neg\psi$. We obtain that $\mathcal{M}^\Lambda, w \Vdash \neg\psi$ iff $\mathcal{M}^\Lambda, w \nVdash \psi$ iff $\psi \notin w$ (by the induction hypothesis for ψ) iff $\neg\psi \in w$ (by Proposition 2.77.(iii)).

Canonical models - Truth Lemma

- $\varphi = \psi \to \chi$. We have that $\mathcal{M}^\Lambda, w \Vdash \psi \to \chi$ iff $\big(\mathcal{M}^\Lambda, w \Vdash \psi$ implies $\mathcal{M}^\Lambda, w \Vdash \chi\big)$ iff $\big(\psi \in w$ implies $\chi \in w\big)$ (by the induction hypotheses for ψ, χ) iff $\psi \to \chi \in w$ (by Proposition 2.77.(iv)).
- $\varphi = \Diamond\psi$.
  $\Rightarrow$ Suppose that $\mathcal{M}^\Lambda, w \Vdash \Diamond\psi$. Applying the induction hypothesis for ψ, it follows that
  $$\text{there exists } v \in W^\Lambda \text{ such that } R^\Lambda wv \text{ and } \psi \in v.$$
  By the definition of R^Λ, it follows that $\Diamond\psi \in w$.
  $\Leftarrow$ Suppose that $\Diamond\psi \in w$. Applying the Existence Lemma, it follows that there exists $v \in W^\Lambda$ such that $R^\Lambda wv$ and $\psi \in v$. By the induction hypothesis for ψ, we obtain that there exists $v \in W^\Lambda$ such that $R^\Lambda wv$ and $\mathcal{M}^\Lambda, v \Vdash \psi$. Thus, $\mathcal{M}^\Lambda, w \Vdash \Diamond\psi$. $\square$

Theorem 2.82 (Canonical model theorem - version 1)

Every Λ-consistent set is satisfiable in the canonical model $\mathcal{M}^\Lambda$.

Proof: Let Γ be a Λ-consistent set. By Lindenbaum's Lemma, there exists $w \in W^\Lambda$ such that $\Gamma \subseteq w$. By the Truth Lemma, it follows that $\mathcal{M}^\Lambda, w \Vdash \varphi$ for any $\varphi \in \Gamma$. Thus, $\mathcal{M}^\Lambda, w \Vdash \Gamma$. $\square$

Applying Proposition 2.71, we get

Theorem 2.83 (Canonical model theorem - version 2)

Λ is strongly complete with respect to the canonical model $\mathcal{M}^\Lambda$.

These results are the essential tools for obtaining completeness theorems for normal logics with respect to classes of frames.

Completeness theorem for $\boldsymbol{K}$

Theorem 2.84

$\boldsymbol{K}$ is strongly complete with respect to the class of all frames for ML_0.

Proof: We apply Proposition 2.71. Let Γ be a $\boldsymbol{K}$-consistent set of formulas. We have to find a model $\mathcal{M}$ in which Γ is satisfiable. By Theorem 2.82, we can take $\mathcal{M} := \mathcal{M}^{\boldsymbol{K}}$, the canonical model of $\boldsymbol{K}$. $\square$

Theorem 2.85

$\boldsymbol{K}$ is sound and weakly complete with respect to the class of all frames for ML_0.

Proof: We apply the previous result and Theorem 2.68. $\square$.

*Logic **K**4*

Let

$$(4) \quad \Diamond\Diamond p \rightarrow \Diamond p$$

We use the notation $\boldsymbol{K}4$ for the normal logic generated by (4).
Thus, $\boldsymbol{K}4$ is the least normal logic that contains (4).

The canonical model for $\boldsymbol{K}4$ is $\mathcal{M}^{\boldsymbol{K}4} = (W^{\boldsymbol{K}4}, R^{\boldsymbol{K}4}, V^{\boldsymbol{K}4})$ and the canonical frame for $\boldsymbol{K}4$ is $\mathcal{F}^{\boldsymbol{K}4} = (W^{\boldsymbol{K}4}, R^{\boldsymbol{K}4})$.

By Theorem 2.82, it follows that

Proposition 2.86

Every $\boldsymbol{K}4$-consistent set is satisfiable in the canonical model $\mathcal{M}^{\boldsymbol{K}4}$.

Proposition 2.87

The canonical frame $\mathcal{F}^{\boldsymbol{K}4} = (W^{\boldsymbol{K}4}, R^{\boldsymbol{K}4})$ is transitive.

Proof: Let $w, v, u \in W^{\boldsymbol{K}4}$ be such that $R^{\boldsymbol{K}4}wv$ and $R^{\boldsymbol{K}4}vu$. We have to show that $R^{\boldsymbol{K}4}wu$, that is

$$\text{for any formula } \varphi, \ \varphi \in u \text{ implies } \Diamond\varphi \in w.$$

Let $\varphi \in u$ be a formula. Since $R^{\boldsymbol{K}4}vu$, we have that $\Diamond\varphi \in v$. Since $R^{\boldsymbol{K}4}wv$, it follows that $\Diamond\Diamond\varphi \in w$. As w is a $\boldsymbol{K}4$-MCS, we can apply Proposition 2.77.(ii) to get that $\boldsymbol{K}4 \subseteq w$. In particular, $\Diamond\Diamond\varphi \to \Diamond\varphi \in w$. We apply now modus ponens (Proposition 2.77.(i)) to conclude that $\Diamond\varphi \in w$. $\square$.

Theorem 2.88

$\boldsymbol{K}4$ *is strongly complete with respect to Tran, the class of transitive frames.*

Proof: We apply Proposition 2.71. Let Γ be a $\boldsymbol{K}4$-consistent set. By Theorem 2.82, Γ is satisfiable in $\mathcal{M}^{\boldsymbol{K}4}$. Applying Proposition 2.87, we obtain that $\mathcal{F}^{\boldsymbol{K}4} \in \mathit{Tran}$. $\square$

Theorem 2.89

$\boldsymbol{K}4$ *is sound and weakly complete with respect to Tran; that is,* $\boldsymbol{K}4 = \Lambda_{\mathit{Tran}}$.

Proof: By Theorem 2.42 and Example 2.25 we obtain that Λ_{Tran} is a normal logic that contains (4). Hence, $\boldsymbol{K}4 \subseteq \Lambda_{\mathit{Tran}}$, that is $\boldsymbol{K}4$ is sound with respect to *Tran*.
It follows immediately from Theorem 2.88 that $\boldsymbol{K}4$ is weakly complete with respect to *Tran*, that is $\boldsymbol{K}4 \supseteq \Lambda_{\mathit{Tran}}$. $\square$

Logic $\boldsymbol{T}$

Let

$$(T) \quad p \to \Diamond p$$

We use the notation $\boldsymbol{T}$ for the normal logic generated by (T).

Definition 2.90

We say that a frame $\mathcal{F} = (W, R)$ is reflexive if R is reflexive.

Notation: Ref is the class of reflexive frames

Proposition 2.91

(T) is valid in Ref.

Proof: Let $\mathcal{F} = (W, R)$ be a reflexive frame, w a state in $\mathcal{F}$ and $\mathcal{M} = (\mathcal{F}, V)$ a model based on $\mathcal{F}$. Suppose that $\mathcal{M}, w \Vdash p$. Since R is reflexive, it follows that Rww and $\mathcal{M}, w \Vdash p$. Thus, $\mathcal{M}, w \Vdash \Diamond p$. $\square$

Logic $\boldsymbol{T}$

The canonical model for $\boldsymbol{T}$ is $\mathcal{M}^{\boldsymbol{T}} = (W^{\boldsymbol{T}}, R^{\boldsymbol{T}}, V^{\boldsymbol{T}})$ and the canonical frame is $\mathcal{F}^{\boldsymbol{T}} = (W^{\boldsymbol{T}}, R^{\boldsymbol{T}})$.
Applying Theorem 2.82, we get that

Proposition 2.92

Every $\boldsymbol{T}$-consistent set is satisfiable in the canonical model $\mathcal{M}^{\boldsymbol{T}}$.

Proposition 2.93

The canonical frame $\mathcal{F}^{\boldsymbol{T}} = (W^{\boldsymbol{T}}, R^{\boldsymbol{T}})$ is reflexive.

Proof: Let $w \in W^{\boldsymbol{T}}$. We have to show that $R^{\boldsymbol{T}}ww$, that is

$$\text{for any formula } \varphi,\ \varphi \in w \text{ implies } \Diamond\varphi \in w.$$

Let $\varphi \in w$ be a formula. Since w is a $\boldsymbol{T}$-MCS, we can apply Proposition 2.77.(ii) to get that $\boldsymbol{T} \subseteq w$. In particular, $\varphi \to \Diamond\varphi \in w$. We apply now modus ponens (Proposition 2.77.(i)) to conclude that $\Diamond\varphi \in w$. $\square$.

Theorem 2.94

$\boldsymbol{T}$ is strongly complete with respect to Ref.

Proof: We apply Proposition 2.71. Let Γ be a $\boldsymbol{T}$-consistent set. By Theorem 2.82, Γ is satisfiable in $\mathcal{M}^{\boldsymbol{T}}$. Applying Proposition 2.93, we obtain that $\mathcal{F}^{\boldsymbol{T}} \in Ref$. $\square$

Theorem 2.95

$\boldsymbol{T}$ is sound and weakly complete with respect to Ref; that is, $\boldsymbol{T} = \Lambda_{Ref}$.

Proof: By Theorem 2.42 and Proposition 2.91 we obtain that Λ_{Ref} is a normal logic that contains (T). Hence, $\boldsymbol{T} \subseteq \Lambda_{Ref}$, that is $\boldsymbol{T}$ is sound with respect to Ref.
It follows immediately from Theorem 2.94 that $\boldsymbol{T}$ is weakly complete with respect to Ref, that is $\boldsymbol{T} \supseteq \Lambda_{Ref}$. $\square$

Logic $\boldsymbol{B}$

Let

$$(B) \quad p \to \Box\Diamond p$$

We use the notation $\boldsymbol{B}$ for the normal logic $\boldsymbol{KB}$ generated by (B).

Definition 2.96

We say that a frame $\mathcal{F} = (W, R)$ is symmetric if R is symmetric.

Theorem 2.97

$\boldsymbol{B}$ *is strongly complete with respect to the class of symmetric frames.*

Theorem 2.98

$\boldsymbol{B}$ *is sound and weakly complete with respect to the class of symmetric frames.*

Logic $\boldsymbol{S}4$

We use the notation $\boldsymbol{S}4$ for the normal logic $\boldsymbol{KT}4$ generated by (T) and (4).

Theorem 2.99

$\boldsymbol{S}4$ *is strongly complete with respect to the class of reflexive and transitive frames.*

Theorem 2.100

$\boldsymbol{S}4$ *is sound and weakly complete with respect to the class of reflexive and transitive frames.*

*Logic **S**5*

We use the notation $\boldsymbol{S}5$ for the normal logic $\boldsymbol{KT}4\boldsymbol{B}$ generated by (T), (4) and (B).

Theorem 2.101

$\boldsymbol{S}5$ is strongly complete with respect to the class of frames whose relation is an equivalence relation.

Theorem 2.102

$\boldsymbol{S}5$ is sound and weakly complete with respect to the class of frames whose relation is an equivalence relation.

Logic ***KD***

Let

$$(D) \quad \Box p \to \Diamond p$$

Let ***KD*** be the normal logic generated by (D).

Definition 2.103

We say that a frame $\mathcal{F} = (W, R)$ is right-unbounded if for all $x \in W$ there exists $y \in W$ such that Rxy.

Theorem 2.104

KD *is strongly complete with respect to the class of right-unbounded frames.*

Theorem 2.105

KD *is sound and weakly complete with respect to the class of right-unbounded frames.*

Logic $\boldsymbol{K}4.3$

Let

$$(.3) \quad \Diamond p \wedge \Diamond q \to \Diamond(p \wedge \Diamond q) \vee \Diamond(p \wedge q) \vee \Diamond(q \wedge \Diamond p)$$

Let $\boldsymbol{K}4.3$ be the normal logic generated by (4) and (.3).

Definition 2.106

We say that a frame $\mathcal{F} = (W, R)$ has no branching to the right if for all $x, y, z \in W$,

$$Rxy \text{ and } Rxz \quad \text{implies} \quad Ryz \text{ or } y = z \text{ or } Rzy.$$

Theorem 2.107

$\boldsymbol{K}4.3$ *is strongly complete with respect to the class of transitive frames that have no branching to the right.*

Theorem 2.108

$\boldsymbol{K}4.3$ *is sound and weakly complete with respect to the class of transitive frames that have no branching to the right.*

Logic ***KL***

Let

$$(L) \quad \Box(\Box p \to p) \to \Box p$$

This axiom we call (L) (for Löb) is also known as G (for Gödel). Let ***KL*** be the normal logic generated by (L).

Theorem 2.109

KL *is not sound and strongly complete with respect to any class of frames.*

Theorem 2.110

KL *is weakly complete with respect to the class of frames whose relation is a finite strict order relation (that is, the class of finite irreflexive transitive frames).*

Definition 2.111

A modal language ML consists of:

- *a set PROP of atomic propositions;*
- *the propositional connectives:* $\neg$, $\rightarrow$;
- *the propositional constant* $\perp$ *(false);*
- *parantheses:* (,);
- *a set O of modal operators or modalities;*
- *an arity function* $\rho : O \rightarrow \mathbb{N}$.

ML is uniquely determined by $PROP$ and the pair $\tau := (O, \rho)$. We shall use the notation $ML := ML(PROP, \tau)$ to specify this fact. τ is called the similarity type of ML.

Basic modal language

Let τ_0 be the similarity type of the basic modal language ML_0; that is, $ML_0 = ML(PROP, \tau_0)$. Then $\tau_0 = (\{\Diamond\}, \rho)$ with $\rho(\Diamond) = 1$.

Modal languages

Let $ML := ML(PROP, \tau)$ be a modal language, where $\tau = (O, \rho)$.

- Atomic propositions are denoted by $p, q, r, v, \ldots$.
- Elements of O are denoted by Δ, Δ_0, $\Delta_1, \ldots$ and are called modal operators.
- For every $m \in \mathbb{N}$, let $O_m := \{\Delta \in O \mid \rho(\Delta) = m\}$ be the set of modal operators with arity m.
- Unary modal operators are those with arity 1. We refer to them as diamonds and we denote them by $\Diamond_a$ or $\langle a \rangle$, where a is an element from an index set.
- The definition allows modal operators with arity 0, which are also called modal constants.

The set $Sym(ML)$ of symbols of ML is

$$Sym(ML) := PROP \cup \{\neg, \rightarrow, \bot, (,)\} \cup O.$$

Modal languages

The set *Expr*(ML) of expressions of ML is the set of all finite sequences of symbols of ML.

Definition 2.112

The formulas of ML are the expressions inductively defined as follows:

- (F0) *Every atomic proposition is a formula.*
- (F1) $\perp$ *is a formula.*
- (F2) *If φ is a formula, then $(\neg\varphi)$ is a formula.*
- (F3) *If φ and ψ are formulas, then $(\varphi \to \psi)$ is a formula.*
- (F4) *If $m \in \mathbb{N}$, $\Delta \in O_m$ and $\varphi_1, \ldots, \varphi_m$ are formulas, then $(\Delta\varphi_1 \ldots \varphi_m)$ is a formula.*
- (F5) *Only the expressions obtained by applying rules (F0), (F1), (F2), (F3), (F4) are formulas.*

Modal languages

Notation: The set of formulas is denoted by *Form*(*ML*).

- Every formula is obtained by applying the rules (F0), (F1), (F2), (F3), (F4) a finite number of times.
- $Form(ML) \subseteq Expr(ML)$. Formulas are the "well-formed" expressions.

Remark 2.113

Formulas of ML are defined, using the Backus-Naur notation, as follows:

$$\varphi ::= p \mid \bot \mid (\neg\varphi) \mid (\varphi \to \psi) \mid (\Delta\varphi_1 \ldots \varphi_{\rho(\Delta)}),$$

where $p \in PROP$.

Proposition 2.114 (Unique readability)

If φ is a formula, then exactly one of the following holds:

- $\varphi = p$, *where* p *is an atomic proposition;*
- $\varphi = \bot$;
- $\varphi = (\neg\psi)$, *where* ψ *is a formula;*
- $\varphi = (\psi \to \chi)$, *where* ψ, χ *are formulas;*
- $\varphi = (\Delta\psi_1 \dots \psi_m)$, *where* $m \in \mathbb{N}$, $\Delta \in O_m$ *and* $\psi_1, \dots, \psi_m$ *are formulas.*

Furthermore, φ *can be written in a unique way in one of these forms.*

Modal languages

Derived connectives

Connectives $\vee$, $\wedge$, $\leftrightarrow$ and the constant $\top$ (true) are introduced as in classical propositional logic:

$$\varphi \vee \psi := ((\neg\varphi) \rightarrow \psi) \qquad \varphi \wedge \psi := \neg(\varphi \rightarrow (\neg\psi))$$
$$\varphi \leftrightarrow \psi := ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)) \qquad \top := \neg\bot.$$

- Usually the external parantheses are omitted, we write them only when necessary. We write $\neg\varphi, \varphi \rightarrow \psi$, $\Delta\varphi_1 \ldots \varphi_m$.
- To reduce the use of parentheses, we assume that
  - modal operators have higher precedence than the other connectives.
  - $\neg$ has higher precedence than $\rightarrow, \wedge, \vee, \leftrightarrow$;
  - $\wedge, \vee$ have higher precedence than $\rightarrow, \leftrightarrow$.

- We write sometimes $\Delta(\varphi_1, \ldots, \varphi_m)$ instead of $\Delta\varphi_1 \ldots \varphi_m$.
- Binary modal operators are those with arity 2. For them we use infix notation; that is, we write $\varphi\Delta\psi$ instead of $\Delta\varphi\psi$.

Dual modal operators

We define dual operators for the modalities of arity ≥ 1. Let $m \in \mathbb{N}, m \geq 1$ and $\Delta \in O_m$. The dual ∇ of Δ is defined as follows:

$$\nabla\varphi_1 \ldots \varphi_m := \neg\Delta(\neg\varphi_1) \ldots (\neg\varphi_m).$$

As in basic modal logic, the dual of a diamond is called a box. The dual of $\Diamond_a$ is denoted by $\Box_a$ and the dual of $\langle a \rangle$ is denoted by $[a]$. Thus,

$$\Box_a\varphi = \neg\Diamond_a\neg\varphi, \quad [a] = \neg\langle a \rangle\neg\varphi.$$

Definition 2.115

A frame for ML is a pair

$$\mathcal{F} = (W, \{R_\Delta \mid \Delta \in O\})$$

such that

- *W is a nonempty set;*
- *for every $\Delta \in O$, R_Δ is a relation on W with arity $\rho(\Delta) + 1$.*

Thus, frames are relational structures in this case, too.

Notations

- We write sometimes $\mathcal{F} = (W, R_\Delta)_{\Delta \in O}$.
- If O has a finite number of operators $\Delta_1, \ldots, \Delta_n$, we write

$$\mathcal{F} = (W, R_{\Delta_1}, R_{\Delta_2}, \ldots, R_{\Delta_n}).$$

Frames and models

The notion of model is defined exactly as for the basic modal language.

Definition 2.116

A model for ML is a pair $\mathcal{M} = (\mathcal{F}, V)$, where $\mathcal{F} = (W, \{R_\Delta \mid \Delta \in O\})$ is a frame for ML and $V : PROP \to 2^W$ is a valuation.

We say that the model $\mathcal{M} = (\mathcal{F}, V)$ is based on the frame $\mathcal{F}$ or that $\mathcal{F}$ is the frame underlying $\mathcal{M}$. Elements of W are called states in $\mathcal{F}$ or in $\mathcal{M}$. We sometimes write $w \in \mathcal{F}$ or $w \in \mathcal{M}$. We write also $\mathcal{M} = (W, \{R_\Delta \mid \Delta \in O\}, V)$.

Frames and models

Let $\mathcal{M} = (W, \{R_\Delta \mid \Delta \in O\}, V)$ be a model and w a state in $\mathcal{M}$. The notion

formula φ is satisfied (or true) in $\mathcal{M}$ at state w,

Notation $\mathcal{M}, w \Vdash \varphi$

is defined inductively. The clauses for atomic propositions, $\bot, \neg, \rightarrow$ are the same as for the basic modal language (see Definition 2.11)

For the modal operators, we have two cases:

- If $\Delta \in O_m$ with $m \geq 1$, then for any formulas $\varphi_1, \ldots, \varphi_m$,

$$\mathcal{M}, w \Vdash \Delta\varphi_1 \ldots \varphi_m \text{ iff there exist } v_1, \ldots, v_m \in W \text{ s.t. } R_\Delta w v_1 \ldots v_m \text{ and } \mathcal{M}, v_i \Vdash \varphi_i \text{ for every } i = 1, \ldots, m$$

- If $\rho(\Delta) = 0$, then

$$\mathcal{M}, w \Vdash \Delta \quad \text{iff} \quad w \in R_\Delta.$$

Thus, modal constants do not access other states. Their semantics is identical to that of the atomic propositions, only that the unary relations used to interpret them are not given by the valuation, they are part of the underlying frame.

If $\mathcal{M}$ does not satisfy φ at w, we write $\mathcal{M}, w \nVdash \varphi$ and we say that φ is false in $\mathcal{M}$ at state w.

Proposition 2.117

For every formulas φ, ψ,

$$\begin{aligned} \mathcal{M}, w \Vdash \varphi \vee \psi & \quad \textit{iff} \quad \mathcal{M}, w \Vdash \varphi \textit{ or } \mathcal{M}, w \Vdash \psi \\ \mathcal{M}, w \Vdash \varphi \wedge \psi & \quad \textit{iff} \quad \mathcal{M}, w \Vdash \varphi \textit{ and } \mathcal{M}, w \Vdash \psi \end{aligned}$$

Proposition 2.118

Suppose that $\Delta \in O_m, m \geq 1$ and that ∇ is its dual operator. Then for any formulas $\varphi_1, \ldots, \varphi_m$,

$$\begin{aligned} \mathcal{M}, w \Vdash \nabla\varphi_1 \ldots \varphi_m \quad & \textit{iff} \quad \textit{for any } v_1, \ldots, v_m \in W, R_\Delta w v_1 \ldots v_m \\ & \qquad \textit{implies } \mathcal{M}, v_i \Vdash \varphi_i \textit{ for some } i = 1, \ldots, m. \end{aligned}$$

Proof: Exercise.

Frames and models

We define as for the basic modal language the following notions:

- (globally) true or satisfiable formulas in a model (see Definition 2.15);
- formulas that are valid (at a state) in a frame (see Definition 2.18);
- sets of formulas that are true at a state, (globally) true or satisfiable in a model (see Definition 2.16).

As before, these notions are extended to classes of structures (frames or models) for *ML* and we define the local and global semantic consequence relations exactly as in Definitions 2.26 and 2.29.

Temporal logic

Let us consider the similarity type $\tau = (O, \rho)$, where

$$O = \{\langle F\rangle, \langle P\rangle\} \text{ and } \rho(\langle F\rangle) = \rho(\langle P\rangle) = 1.$$

The language determined by τ (and $PROP$) is called the basic temporal language and it is the core language underlying temporal logic, one of the most important modal logics, with numerous applications in computer science.

The intended interpretation for the modal operators $\langle F\rangle, \langle P\rangle$ is:

- $\langle F\rangle \varphi$ is read as φ will be true at some Future time. Hence, F comes from Future.
- $\langle P\rangle \varphi$ is read as φ was true at some Past time. Hence, P comes from Past.

Temporal logic

It is traditional to write $\langle F \rangle$ as F and $\langle P \rangle$ as P. The dual of F is denoted by G and the dual of P is denoted by H.
The interpretation for the operators G, H is:

- $G\varphi$ is read as it is always Going to be the case that φ.
- $H\varphi$ is read as it always Has been the case that φ.

The frames for this language have the following form:

$$\mathcal{F} = (T, R_F, R_P)$$

consisting of a nonempty set T (of time instances) and two binary relations on T: R_F (the into-the-future relation) and R_P (the into-the-past relation), used to interpret F and P respectively.

However, taking into account the intended reading of the operators F and P, most of these frames are inappropriate. It is clear that we would like to use frames in which R_P is the converse or R_F, that is

$$\text{for every } s, t \in T,\ R_F st \text{ iff } R_P ts.$$

Definition 2.119

A bidirectional frame is a frame $\mathcal{F} = (T, R, R^{-1})$, where R is a binary relation. A bidirectional model is a model based on a bidirectional frame.

We interpret the basic temporal language only in bidirectional models. Thus, if $\mathcal{M} = (T, R, R^{-1}, V)$ is a bidirectional model, then

$$\begin{array}{lll} \mathcal{M}, t \Vdash F\varphi & \text{iff} & \text{there exists } s \in T \text{ such that } Rts \text{ and } \mathcal{M}, s \Vdash \varphi \\ \mathcal{M}, t \Vdash P\varphi & \text{iff} & \text{there exists } s \in T \text{ such that } R^{-1}ts \text{ and } \mathcal{M}, s \Vdash \varphi. \end{array}$$

Temporal logic

Once we have imposed the above restriction, it is not necessary to mention R^{-1} explicitly, as it is determined by R. Hence, we can interpret the basic temporal language in models $\mathcal{M} = (T, R, V)$ based on frames $\mathcal{F} = (T, R)$ by using the clauses:

$$\begin{array}{lll} \mathcal{M}, t \Vdash F\varphi & \text{iff} & \text{there exists } s \in T \text{ such that } Rts \text{ and } \mathcal{M}, s \Vdash \varphi \\ \mathcal{M}, t \Vdash P\varphi & \text{iff} & \text{there exists } s \in T \text{ such that } Rst \text{ and } \mathcal{M}, s \Vdash \varphi. \end{array}$$

We have thus pointed out the fundamental interaction between F and P: F looks forward along R and P looks backwards along R. Obviously, for our frames to start looking genuinely temporal, the binary relation R must have some other properties (for example, transitivity, to capture the flow of time)

Dynamic logic

Another important modal logic is propositional dynamic logic (PDL).

The language of propositional dynamic logic has an infinite collection of diamonds (that is, unary modalities).
Each of these diamonds has the form $\langle \pi \rangle$, where π denotes a program.

- The intended interpretation of $\langle \pi \rangle \, \varphi$ is: some terminating execution of π from the present state leads to a state in which φ holds.
- The dual assertion $[\pi]\varphi$ states that every execution of π from the present state leads to a state in which φ holds.

Dynamic logic

PDL becomes highly expressive due to the following idea:

the inductive structure of the programs is made explicit in its syntax.

Complex programs are built out of basic programs using some program constructors.

There are different versions of PDL depending on the choice of these constructors. In the sequel we introduce the basic version called regular PDL.

Dynamic logic

Let Π_0 be a set of atomic programs, denoted $a, b, c, \ldots$.

Definition 2.120

The set Π of regular programs is defined inductively as follows:

- $\Pi_0 \subseteq \Pi$.
- *If $\pi_1, \pi_2 \in \Pi$, then $\pi_1 \cup \pi_2 \in \Pi$ and $\pi_1; \pi_2 \in \Pi$.*
- *If $\pi \in \Pi$, then $\pi^* \in \Pi$;*

We use the following terminology for the operators: $\cup$ is the choice, ; is the composition and $*$ is the iteration.

Compound programs have the following intutitive meaning:

$\pi_1 \cup \pi_2$	Execute either π_1 or π_2, the choice being nondeterministic.
$\pi_1; \pi_2$	First execute π_1, then execute π_2.
π^*	Execute π a finite (possibly zero) number of times.

Dynamic logic

Let us consider the similarity type $\tau = (O, \rho)$, where

$$O = \{<\pi> \mid \pi \in \Pi\} \text{ and } \rho(<\pi>) = 1 \text{ for every } \pi \in \Pi.$$

The modal language determined by τ (and *PROP*) is called the language of regular propositional dynamic logic (regular PDL).

Example

The formula

$$<\pi^*>\varphi \leftrightarrow \varphi \vee <\pi;\pi^*>\varphi$$

says that a state in which φ holds can be reached by executing π a finite number of times if and only if either φ holds in the current state, or we can execute π once and then find a state in which φ holds after finitely many iterations of π.

Dynamic logic

The frames for this language have the following form:

$$\mathcal{F} = (W, \{R_\pi \mid \pi \in \Pi\})$$

where W is a nonempty set of program states and, for every $\pi \in \Pi$, R_π is a binary relation on W.

For every $\pi \in \Pi$, $R_\pi wu$ means

there is an execution of π which begins in state w and ends in state u.

R_π is the set of input/output pairs of states of the program π.

As with the basic temporal language, most of these frames are inappropriate. Given our readings of $\cup$, ; and $*$ as choice, composition, and iteration, we are only interested in the so-called regular frames.

Definition 2.121

A regular frame is a frame

$$\mathcal{F} = (W, \{R_\pi \mid \pi \in \Pi\})$$

such that, for every $\pi \in \Pi$, R_π is defined by the following inductive clauses:

- *If $\pi \in \Pi_0$, then R_π is an arbitrary binary relation on W.*
- *If $\pi = \pi_1 \cup \pi_2$, then $R_\pi = R_{\pi_1} \cup R_{\pi_2}$.*
- *If $\pi = \pi_1 ; \pi_2$, then $R_\pi = R_{\pi_1} \circ R_{\pi_2}$.*
- *If $\pi = \pi_1^*$, then R_π is the reflexive transitive closure $(R_{\pi_1})^*$ of R_{π_1}, that is*

$$R_\pi = (R_{\pi_1})^* = \bigcup_{n \in \mathbb{N}} (R_{\pi_1})^n.$$

Definition 2.122

A regular model is a model based on a regular frame, that is, a regular frame together with a valuation.

These are the models that capture the intended interpretation for regular PDL.

Regular frames/models are also called standard frames/models.

A wide range of other program constructors have been studied; see, for example,

D. Harel, D. Kozen, J. Tiuryn, Dynamic Logic, MIT Press (2006)

for a comprehensive study of dynamic logic.

Dynamic logic

Add in Definition 2.120 of the set Π of programs the following inductive clause:

- If $\pi_1, \pi_2 \in \Pi$, then $\pi_1 \cap \pi_2 \in \Pi$.

The operator $\cap$ is called intersection and has the following meaning:

Execute both π_1 and π_2, in parallel.

The intended interpretation of $\langle \pi_1 \cap \pi_2 \rangle \varphi$ is:

If we execute both π_1 and π_2 in the current state, then there is at least one state reachable by both programs in which φ holds.

Dynamic logic

Add in Definition 2.120 of the set Π of programs the following inductive clause:

- If φ is a formula, then $\varphi? \in \Pi$.

The operator ? is called test and has the following meaning:

Test whether φ holds in the current state and if so, continue; if not, fail.

The test constructor has an unusual syntax: it allows us to make a modality from any formula. This is the rich test version of PDL.

$$\begin{array}{lcl} \textbf{if } \varphi \textbf{ then } \pi_1 \textbf{ else } \pi_2 & := & \varphi?;\pi_1 \cup \neg\varphi?;\pi_2 \\ \textbf{while } \varphi \textbf{ do } \pi & := & (\varphi?;\pi)^*;\neg\varphi? \end{array}$$