My lecture notes of Logic 1
(course of Prof. Tudor Jebelean)

Martin Köhler

WS 2005

Contents

List of Figures

List of Tables

Chapter 1

First Lecture on 13.10.2005

1.1 What is logic?

Note: Logic is “Reasoning about reasoning”

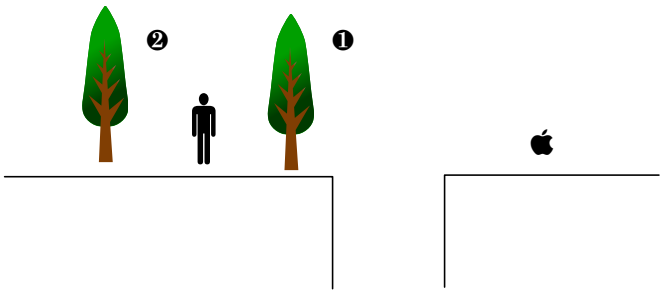


Figure 1.1: Problem: Person wants to get the apple

Problem:

- Person wants to cross the gap to an apple
- He could try to jump, but would fall down
- He could try it with the tree, but it would break
- So he tries to make experiments, just to explore/study the problem in his head, f.ex. with the second tree

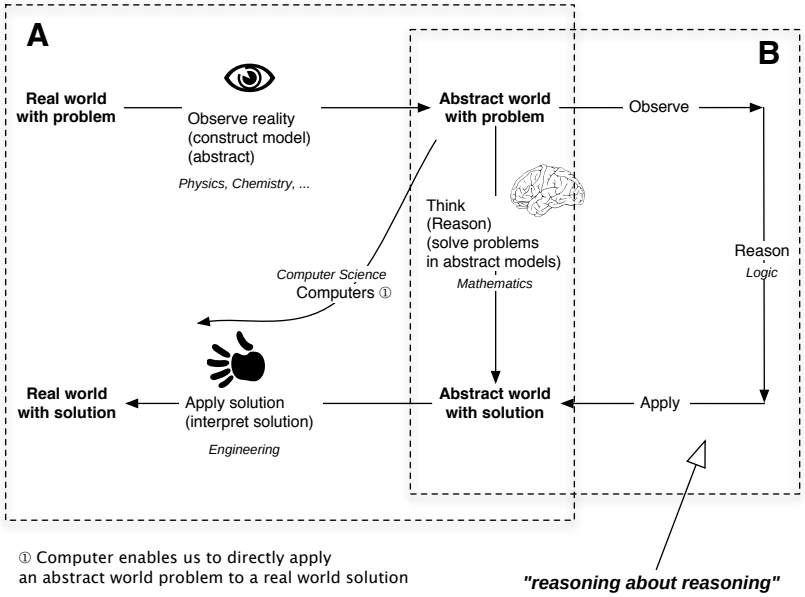


Figure 1.2: Essential steps in human problem solving (A)

Note: "programs are essentially logical formulae"

1.2 Why Logic ?

1.2.1 Russel Paradoxon

Sets: $a \in A$

Consider things like

- set of all students in this room
- set of all even numbers
- set of all numbers
- set of all sets

Problem:

$$\text{Obtain a contradiction:} \quad \begin{array}{c} A \notin A \quad \text{f.ex. } \{1\} \notin \{1\} \\ X = \{A \,|A \notin A\} \\ \forall_B B \in X \Leftrightarrow B \notin B \\ X \in X \Leftrightarrow X \notin X \\ X \in X \Rightarrow X \notin X \quad \lightning \\ X \notin X \Rightarrow X \in X \quad \lightning \end{array}$$

$$\text{Only the conjunction } \neg A \wedge A \text{ is wrong} \quad \begin{cases} \text{not a contradiction} & A \Rightarrow \neg A \equiv \neg A \vee \neg A \equiv \neg A \\ \text{not a contradiction} & \neg A \Rightarrow A \equiv \neg(\neg A) \vee A \equiv A \end{cases}$$

Via the rule $P \Rightarrow Q \equiv (\neg P) \vee Q$. Corresponding truth table is:

$\mathscr{B}_{\Rightarrow}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$

So $X = \{A \,|A \notin A\}$ has the implicit axiom $\exists_M \forall_B B \in M \Leftrightarrow B \notin B$. This is in this case wrong, we do simply not make such assumptions.

$$X = \left\{ A \,\middle|_{A \in M} \quad \underbrace{\mathscr{P}[A]}_{\text{"property"}} \right\}$$

Note: "there is no truth in logic"

Note: "there is no sets of all sets"

1.2.2 Termination Problem

$$\underbrace{p \downarrow x}$$

Does the program p terminate with input x ?

One can think of a(n encoded) program as a long number, and of an input as also a long number.

$$\text{(Program } \spadesuit) \quad T[p] = \begin{cases} \mathbb{T} & \text{if } p \downarrow p \\ \mathbb{F} & \text{if } p \not\downarrow p \end{cases}$$

$$X[p, x] = \begin{cases} \mathbb{T} & \text{if } p \downarrow x \\ \mathbb{F} & \text{if } p \not\downarrow x \end{cases}$$

$$\text{(Program)} \quad S[p] = \begin{cases} \mathbb{T} & \text{if } \neg T[p] \\ S[p] & \text{if } T[p] \end{cases}$$

read like: calls $T[p]$,

- if it returns false, then returns true
- if it returns true, then it returns $S[p]$

$$\forall_p S \downarrow p \quad \Leftrightarrow \quad p \not\downarrow p$$
$$\text{so this would mean that: } S \downarrow S \quad \Leftrightarrow \quad S \not\downarrow S$$

So this shows that such a program like ♠ - which decides if another program terminates - cannot exist.

The termination problem is *undecidable*.

But this doesn't mean that a special program can't be verified, but only means that a most general verification is not possible.

1.2.3 Sorting

$$\{4,3,1,2\} \rightarrow \{1,2,3,4\}$$

Intuitive Def.: $\ldots, a, b, \ldots \quad a \leq b$

Logical Def.:

$$\forall_s \Big(\; isSorted[S] \;\; \Leftrightarrow \;\; \Big(\forall_{X,Y} \forall_{a,b} \Big(\big(\; S \;\; = \;\; X \smile \langle a,b\rangle \smile Y\Big) \Rightarrow (a \leq b)\Big)\Big) \; \Big)$$

Note: The symbol $\smile$ is used here to express concatenation.

$$\begin{array}{rcl} Sort[S] & & \\ isSorted[S] & \Rightarrow & Sort[S] = S \\ \neg isSorted[S] & \Rightarrow & \exists_{X,Y} \exists_{a,b} \left((S = X \smile \langle a,b\rangle \smile Y) \wedge (a > b)\right) \end{array}$$

$$(a > b) \quad \Rightarrow \quad (Sort\,[X \smile \langle a,b\rangle \smile Y] = Sort\,[X \smile \langle b,a\rangle \smile Y])$$

Implementation in Mathematica:

```
In: S[{X___,a_,b_,Y___}]/; (a>b):=S[{X,b,a,Y}];
In: S[L_]:=L;
In: S[{3,4,2,1}]
Out: {1,2,3,4}
```

1) Logic Syntax (formulae)

$$\begin{array}{c} \forall_s \Big(\; isSorted[S] \;\; \Rightarrow \;\; (Sort[S] = S) \; \Big) \wedge \\ \forall_s \Big(\neg \; isSorted[S] \;\; \Rightarrow \;\; (Sort[S] = Sort[Modified[S]]) \; \Big) \end{array}$$

2) Different Syntax: Functional programming style

$$Sort[S] \;\; = \;\; \left\{ \begin{array}{cc} S & \text{if } isSorted[S] \\ Sort[Modified[S]] & \text{otherwise} \end{array} \right.$$

3) Different Syntax: Imperative programming style

```
while (¬isSorted[S]) {
    S←Modified[S]
}
```

If the program terminates, then S is sorted.

- The non-termination of 3) corresponds to the non-existance of the function 2).
- So one needs to prove that it terminates.
- Check partial correctness (if certain problem properties are not given).

Motivation: Bugs in programs and their consequences, for example:

- Div-Error in Intel processors (Programming errors)
- AT&T breakdown
- rocket got lost
- ...

Idea: Solve this problems by analysing them mathematically

- all our models are approximative in the real world
- but in the world of the program, everything is concrete, but nevertheless so much is buggy (paradoxon)

1.3 How ?

$$\text{"}\underbrace{\text{it rains}}_{A}\quad\text{and}\quad\underbrace{\text{it is sunny}}_{B}\text{"}$$

or f.ex.

$$\text{"}\underbrace{\text{it rains}}_{A}\quad\text{and}\quad\underbrace{\text{it is not sunny}}_{\neg B}\text{"}$$

1.3.1 Propositional Logic

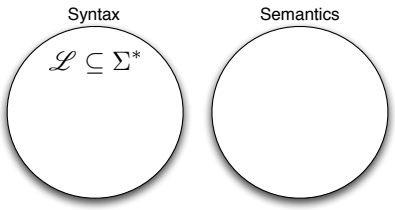


Figure 1.3: Syntax and Semantics - Now Syntax

1.3.1.1 Syntax

Let's formalize the language of propositional logic expressions:

$$\mathscr{L}\ldots\text{Language with "words" like} \begin{cases} A \wedge B \\ A \wedge \neg B \\ (\neg A \wedge B) \Leftrightarrow (A \Rightarrow B) \\ A \wedge \neg A \end{cases}$$

$$\text{Set of "symbols" "alphabet"} \quad \Sigma = \{(,)\} \cup \{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow\} \cup \{\mathbb{T}, \mathbb{F}\} \cup \Theta$$

Note: Θ is the set of propositional variables. F.ex. this could be $\{A, B, C, P, Q, \ldots, A_1, A_2, \ldots\}$. This set Θ should be at least $\infty_{\text{"enumerable"}}$.

All our formulae are words over this alphabet Σ.

For example, the word $A \wedge B$ has three symbols. The sequence is important, so

$$\begin{array}{ccccccc} \text{"}A & \wedge & B\text{"} & \neq & \text{"}\wedge & A & B\text{"} \\ \uparrow & \uparrow & \uparrow & & & & \\ 0 & 1 & 2 & & & & \end{array}$$

$$\begin{aligned} \text{"}A \wedge B\text{"} : \{0,1,2\} &\to \Sigma \\ \text{"}A \wedge B\text{"}[0] &= A \\ \text{in general, } (\omega\,[\in \Sigma^n]) : \{0,\ldots,n\} &\to \Sigma \end{aligned}$$

$$\begin{aligned} \Sigma^* &= \bigcup_{n\geq 0} \Sigma^n \quad \text{(Set of all words [over the alphabet } \Sigma\text{])} \\ \Sigma^0 &= \underset{\substack{\uparrow \\ \text{"empty word"}}}{\{\lambda\}} \end{aligned}$$

A word $(\neg A \vee B)$ can be coded as $\langle (, \neg, A, \vee, B, \ldots \rangle$. This notation is inconvient, but to be mathematically precise, one would have to write it this way.

Generalized Inductive Definitions

- *(0)* $\mathbb{T}, \mathbb{F} \in \mathscr{L}$ or one can also write $\{\mathbb{T}, \mathbb{F}\} \subset \mathscr{L}$
- *(1)* if $\underbrace{\vartheta \in \Theta}_{\text{"variable"}}$, then $\underbrace{\vartheta \in \mathscr{L}}_{\text{"word", logical formula}}$ (a propositional variable ϑ "is" also a logical formula)
- *(2)* if $\varphi, \psi \in \mathscr{L}$, then $\underbrace{\underbrace{\neg\varphi}, \underbrace{(\varphi \wedge \psi)}, \underbrace{(\varphi \vee \psi)}, \underbrace{(\varphi \Rightarrow \psi)}, \underbrace{(\varphi \Leftrightarrow \psi)} \in \mathscr{L}}_{\text{"are also words in the language"}}$
  - Note: The binary operators can also be formulated this way: $(\varphi \alpha \psi) \in \mathscr{L}$, for $\alpha \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$
- *(3)* These are all the formulae

Σ^* obeys / has also these properties *(0), (1)* and *(2)*

Way to reduce it to the little as possible set

$$\text{English} \quad \left\{ \begin{array}{c} \text{if a subset } \mathscr{L}' \text{ of } \Sigma^* \text{ satifies (0),(1) and (2),} \\ \text{then } \mathscr{L} \text{ is a subset of } \mathscr{L}'. \end{array} \right.$$

Formula $\diamond$:

$$\begin{array}{c} \forall_{\mathscr{L}' \subseteq \Sigma^*} \text{if } (\{\mathbb{T}, \mathbb{F}\} \subseteq \mathscr{L}' \text{ and } \vartheta \subseteq \mathscr{L}' \text{ and } \neg\mathscr{L}' \subseteq \mathscr{L}' \text{ and} \\ \text{and } \forall_{\alpha \in \{\wedge,\vee,\Rightarrow,\Leftrightarrow\}} \text{"("}\mathscr{L}'\alpha\mathscr{L}'\text{")"} \subseteq \mathscr{L}') \text{ then } \mathscr{L} \subseteq \mathscr{L}' \end{array}$$

Note:

$$\begin{array}{ccc} \text{(we speak in)} & \text{meta-language} & \text{(f.ex. symbol } \vartheta) \\ & \downarrow & \\ \text{(about)} & \text{object-language} & \text{(f.ex. symbol "}A \wedge B\text{")} \end{array}$$

Note: Preferred notations of Prof. Jebelean:

$$1)\quad \begin{cases} f(x) & \text{is ambiguous:} & \begin{cases} a*(b+c) & \text{(arith. term)} \\ f(b+c) & \text{Function-call} \end{cases} \\ & \downarrow & \\ f[x] & \text{is in this context unambiguous:} & \begin{cases} a*(b+c) & \text{(arith. term)} \\ f[b+c] & \text{Function-call} \end{cases} \end{cases}$$

$$2)\quad \begin{cases} (\forall x)(x \geq 1 \Rightarrow x > 0) \\ (\forall x \in \mathbb{R})(x \geq 1 \Rightarrow x > 0) \end{cases} \rightarrow_{\text{better syntax}} \forall_{\substack{x,y \\ x,y \in \mathbb{R}}} \ldots$$

$L \ldots$ "Length"

- *(0)* $L[\mathbb{T}] = L[\mathbb{F}] = 1$
- *(1)* $L[\vartheta] = 1,$ if $\vartheta \in \Theta$
- *(2)*
  - $L[\neg\varphi] = 1 + L[\varphi]$
  - $L[(\varphi\alpha\psi)] = L[\varphi] + 3 + L[\psi]$, if $\alpha \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$ and $\varphi, \psi \in \mathscr{L}$

Reformulation of formula $\diamond$ with property $\mathscr{P}$:

$$\forall_{\mathscr{P}} \quad \text{if} \quad \left(\begin{array}{c} \forall_{v \in \Theta \cup \{\mathbb{T},\mathbb{F}\}} \mathscr{P}[v] \text{ and} \\ \forall_{\varphi,\psi \in \Sigma^*} \text{if } (\mathscr{P}[\varphi] \text{ and } \mathscr{P}[\psi]) \text{ then } \forall_{\alpha \in \{\wedge,\vee,\Rightarrow,\Leftrightarrow\}} \mathscr{P}[\varphi\alpha\psi] \end{array} \right)$$
$$\text{then} \quad \left(\forall_{\varphi \in \Sigma^*} \text{if } \varphi \in \mathscr{L} \text{ then } \mathscr{P}[\varphi]\right)$$

$$\omega \in \mathscr{L}' \quad \rightsquigarrow \quad \mathscr{P}[\omega]$$

Another way

$$\mathscr{L}_0 = \{\mathbb{T}, \mathbb{F}\} \cup \Theta \qquad \mathscr{S} \rightarrow \neg\mathscr{S} \cup \bigcup_{\alpha \in \{\wedge,\vee,\Rightarrow,\Leftrightarrow\}} (\mathscr{S}\alpha\mathscr{S})$$

$$\forall_{n \geq 0} \mathscr{L}_{n+1} = \mathscr{L}_n \cup \neg\mathscr{L}_n \cup \bigcup_{\alpha \in \{\wedge,\vee,\Rightarrow,\Leftrightarrow\}} (\mathscr{L}_n\alpha\mathscr{L}_n)$$

$$\mathscr{L} = \bigcup_{n \geq 0} \mathscr{L}_n$$

If I know, that a set of strings is increasing, then

$$\mathscr{L}_0 \subseteq \mathscr{L}_1 \subseteq \cdots \subseteq \mathscr{L}_n \subseteq \ldots$$

Or one can simply write $\forall_{n\geq 0}\mathscr{L}_n \subseteq \mathscr{L}_{n+1}$

$$\mathscr{L} = \bigcup_{n\geq 0} \mathscr{L}_n \quad = \quad \left\{ \varphi|_{\varphi\in\Sigma^*} \exists_{n\geq 0}\varphi \in \mathscr{L}_n \right\}$$

1.3.1.2 Semantics

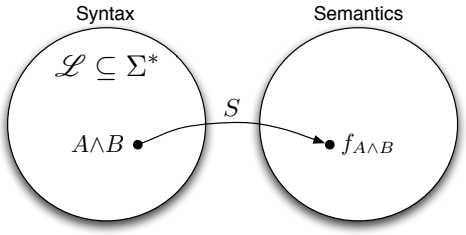


Figure 1.4: Syntax and Semantics - Now Semantics

$A \wedge B$ says us / means "this is only true if both are true"
to describe this mathematically, f.ex. use truth tables :

$\mathscr{B}_\wedge$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

Table 1.1: Truth table for conjunction connective

Can also be described with $f_{A\wedge B} : \mathscr{I} \to \{\mathbb{T}, \mathbb{F}\}$, with $\mathscr{I} : \{\mathcal{I} : \{A, B\} \to \{\mathbb{T}, \mathbb{F}\}\}$

Note: $\mathcal{I}$ is called an *"interpretation"*, whereas $\mathscr{I}$ is the *"set of interpretations"* for all propositional variables, in this case this is only $\{A, B\}$.

So to become more general, one can replace $\{A, B\}$ with the whole set of all propositional variables Θ, resulting in

$$\mathscr{I} : \{\mathcal{I} : \Theta \to \{\mathbb{T}, \mathbb{F}\}\}$$

$\langle\varphi\rangle_{\mathcal{I}}$ ... "Truth evaluation of a formula φ under the interpretation $\mathscr{I}$"

$$\begin{aligned}
\langle\mathbb{F}\rangle_{\mathcal{I}} &= \mathbb{F} \\
\langle\mathbb{T}\rangle_{\mathcal{I}} &= \mathbb{T} \\
\\
\langle\vartheta\rangle_{\mathcal{I}} &= \mathcal{I}[\vartheta] \\
\langle\neg\varphi\rangle_{\mathcal{I}} &= \underbrace{OppositeOf\left[\langle\varphi\rangle_{\mathcal{I}}\right]}_{\text{man schreibt } \mathscr{B}_{\neg}\left[\langle\varphi\rangle_{\mathcal{I}}\right]} \\
\\
\langle\varphi \vee \psi\rangle_{\mathcal{I}} &= \mathscr{B}_{\vee}\left[\langle\varphi\rangle_{\mathcal{I}}, \langle\psi\rangle_{\mathcal{I}}\right]
\end{aligned}$$

$\mathscr{B}_{\neg}$ is given by the truth table for the $\neg$ Operator.

f.ex. Application: Theorem Proving For example $A \wedge \neg A$ is "~~always false~~": The correct term for this is *"unsatisfiable"*: $\forall_{\mathcal{I}} f_{A\wedge\neg A}[\mathcal{I}] = \mathbb{F}$.

When $\langle\varphi\rangle_{\mathcal{I}} = \mathbb{T}$ for a special $\mathcal{I}$, this is called *"$\mathcal{I}$ satisfies φ"* or *"$\mathcal{I}$ is a model for φ"*.

$(A \Rightarrow (A \Rightarrow B)) \Rightarrow B$ is "~~always true~~": The correct term for this is *"valid"*: $\forall_{\mathcal{I}} \langle\ldots\rangle_{\mathcal{I}} = \mathbb{T}$.

A example where a formula is *"invalid", but "satisfiable"* is: $A \wedge B$ (depends on $\mathcal{I}$).

Chapter 2

Second Lecture on 20.10.2005

Propositional Logic

- Syntax, Semantics
- Simplification
- Proof Systems

2.1 Syntax

"these are all the formulae" $\begin{cases} \mathbb{T}, \mathbb{F}, \ \vartheta : A, B, \ldots, P, Q \\ \neg\varphi \\ (\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \Rightarrow \psi), (\varphi \Leftrightarrow \psi) \end{cases}$

Note: from time to time, we might omit the parentheses

2.2 Semantics

2.2.1 Semantics of logical connectives

f.ex. the semantics for $A \wedge B$ is given by $f_{A \wedge B} : \mathscr{I} \to \{\mathbb{T}, \mathbb{F}\}$ with $\mathscr{I} = \{\mathcal{I} : \{A, B\} \to \{\mathbb{T}, \mathbb{F}\}\}$.

Note: ($\mathcal{I}$ is an *interpretation* (and $\mathscr{I}$ is the *set of all interpretations*)).

We introduce the notation $\langle A \wedge B \rangle_{\mathcal{I}} = \mathscr{B}_\wedge \left(\langle A \rangle_{\mathcal{I}}, \langle B \rangle_{\mathcal{I}} \right) = \mathscr{B}_\wedge (\mathbb{T}, \mathbb{F}) = \mathbb{F}$ (for an interpretation $\mathcal{I} = \{A \to \mathbb{T}, B \to \mathbb{F}\}$)

Note: $\mathscr{B}_\wedge$ is a function, given by the truth table of the logical AND-operator.

We transform the problem of evaluating "$A \wedge B$" to the AND of it's components.

	$\mathscr{B}_\neg$
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$

$\mathscr{B}_\wedge$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

$\mathscr{B}_\vee$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$

$\mathscr{B}_\Rightarrow$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$

$\mathscr{B}_\Leftrightarrow$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$

Table 2.1: The semantics of logical connectives

$$\langle \neg\varphi \rangle_I = \mathbb{T} \quad \text{iff } \langle \varphi \rangle_I \neq \mathbb{T}$$
$$\text{i.e.: } not\,(\langle \varphi \rangle_I = \mathbb{T})$$
$$\langle \varphi \wedge \psi \rangle_I = \mathbb{T} \quad \text{iff } \langle \varphi \rangle_I = \mathbb{T} \text{ and } \langle \psi \rangle_I = \mathbb{T}$$

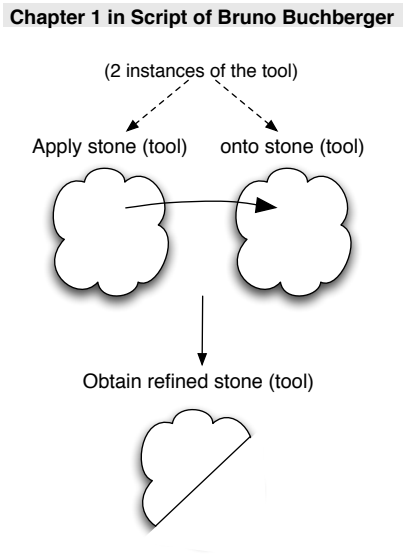


Figure 2.1: "Little Movie" story

2.2.2 Definition of $\langle \Box \rangle_I$

$\langle \mathbb{T} \rangle_I = \mathbb{T}, \langle \mathbb{F} \rangle_I = \mathbb{F}$

Note: in the case of $\langle \mathbb{T} \rangle_I$, $\mathbb{T}$ is a symbol of the *object level*, whereas in the case of $= \mathbb{T}$, $\mathbb{T}$ is a symbol of the *meta level*.

$\langle \vartheta \rangle_I = \mathcal{I}(\vartheta)$

$\langle \neg\varphi \rangle_I = \mathcal{B}_\neg(\langle \varphi \rangle_I)$

$\langle \varphi \wedge \psi \rangle_I = \mathcal{B}_\wedge(\langle \varphi \rangle_I, \langle \psi \rangle_I)$

The same thing can be done analogically to define $\langle \varphi \vee \psi \rangle_I, \langle \varphi \Rightarrow \psi \rangle_I, \langle \varphi \Leftrightarrow \psi \rangle_I$

$$\varphi \rightarrow \begin{cases} f_\varphi : \mathcal{I} \rightarrow \{\mathbb{T}, \mathbb{F}\} \text{ mit } \mathcal{I} = \{\mathcal{I} : \Theta \rightarrow \{\mathbb{T}, \mathbb{F}\}\} \\ f_\varphi(\mathcal{I}) = \langle \varphi \rangle_I \end{cases}$$

Note: Θ denotes the set of all *propositional variables*, whereas ϑ denotes a single *variable* within this set.

$F : \mathcal{L} \rightarrow \mathcal{S}$
$F(\varphi) = f_\varphi$

Based on these semantics we can see the possible transformations.

2.2.3 Example for Simplification

Example ★: For an interpretation $\mathcal{I} = \{A \to \mathbb{T}, B \to \mathbb{F}\}$ one can simplify the object level expression "$(A \wedge (A \Rightarrow B)) \Rightarrow B$" with transformations on the meta level:

$$
\begin{aligned}
\langle (A \wedge (A \Rightarrow B)) \Rightarrow B \rangle_{\mathcal{I}} &= \mathscr{B}_{\Rightarrow}\left(\langle (A \wedge (A \Rightarrow B)) \rangle_{\mathcal{I}}, \langle B \rangle_{\mathcal{I}}\right) \\
&= \mathscr{B}_{\Rightarrow}\left(\mathscr{B}_{\wedge}\left(\langle A \rangle_{\mathcal{I}}, \langle (A \Rightarrow B) \rangle_{\mathcal{I}}\right), \langle B \rangle_{\mathcal{I}}\right) \\
&= \mathscr{B}_{\Rightarrow}\left(\mathscr{B}_{\wedge}\left(\langle A \rangle_{\mathcal{I}}, \mathscr{B}_{\Rightarrow}\left(\langle A \rangle_{\mathcal{I}}, \langle B \rangle_{\mathcal{I}}\right)\right), \langle B \rangle_{\mathcal{I}}\right) \\
&= \mathscr{B}_{\Rightarrow}\left(\underbrace{\mathscr{B}_{\wedge}\left(\mathbb{T}, \underbrace{\mathscr{B}_{\Rightarrow}(\mathbb{T}, \mathbb{F})}_{\mathbb{F}}\right)}_{\mathbb{F}}, \mathbb{F}\right) \\
&= \mathbb{T}
\end{aligned}
$$

Here a schema is given which makes it shorter to write, but in order to be mathematically precise, one would have to use the above version:

$$
\underbrace{\underbrace{\underbrace{\begin{array}{cccccccc} (& A & \wedge (& A & \Rightarrow & B &)) & \Rightarrow \quad B \\ & \mathbb{T} & & \mathbb{T} & & \mathbb{F} & & \mathbb{F} \end{array}}_{\mathbb{F}}}_{\mathbb{F}}}_{\mathbb{T}}
$$

2.2.4 Properties "valid" and "satisfiable"

$\left(\text{For any } \mathcal{I} \in \mathscr{I} : f_{\varphi}(\mathcal{I}) = \mathbb{T}\right)$ is also called *"valid"* (the opposite is *"invalid"*)

$\left(\text{For any } \mathcal{I} \in \mathscr{I} : f_{\varphi}(\mathcal{I}) = \mathbb{F}\right)$ is also called *"unsatisfiable"* (the opposite is *"satisfiable"*)

$\langle \varphi \rangle_{\mathcal{I}} = \mathbb{T}$ is also called "$\mathcal{I}$ satisfies φ" or also "$\mathcal{I}$ is a model of φ"

2.2.5 Definition of $\varphi_1, \ldots, \varphi_n \models \psi$

$$
\varphi_1, \ldots, \varphi_n \models \psi \text{ iff for any } \mathcal{I} \left[\in \underbrace{\mathscr{I}_{Var(\psi) \cup \bigcup_{i=1}^{n} Var(\varphi_i)}}_{\substack{\text{The index: "Union of all variables} \\ \text{occurring in } \varphi_1, \ldots, \varphi_n \text{ and } \psi\text{"}}} \right]
$$

(If $\langle \varphi_1 \rangle_{\mathcal{I}} = \ldots = \langle \varphi_n \rangle_{\mathcal{I}} = \mathbb{T}$, then $\langle \psi \rangle_{\mathcal{I}} = \mathbb{T}$)

Note: $\models$ is equivalent to implication, helps us to make a distinction between object level and meta level

Note: We will not explicitly write the expression in brackets in the future, because it's too complex.

2.2.6 Definition of $\varphi \equiv \psi$

We want to define $\varphi \equiv \psi$.

Naive try:

$$\varphi \equiv \psi \text{ iff for any } \mathcal{I} \in \mathscr{I} : f_\varphi(\mathcal{I}) = f_\psi(\mathcal{I})$$

Problem: For the $\mathcal{I}$ it is not clear what is meant. $\mathcal{I}_\varphi$? $\mathcal{I}_\psi$?

For example $A \equiv B$:

$$\begin{aligned} \mathscr{I}_A &= \{\mathcal{I} : \{A\} \to \{\mathbb{T}, \mathbb{F}\}\} \\ \mathscr{I}_B &= \{\mathcal{I} : \{B\} \to \{\mathbb{T}, \mathbb{F}\}\} \\ \text{in general:} \quad ! &\quad \mathcal{I} : \Theta \to \{\mathbb{T}, \mathbb{F}\} \end{aligned}$$

This general set $\mathscr{I}$ of all the interpretations ($\mathcal{I} : \Theta \to \{\mathbb{T}, \mathbb{F}\}$) can become a very large set, so we only take the subset of relevant interpretations, therefore we use: $\mathscr{I}_{\{A,B\}} : \{\mathcal{I} : \{A, B\} \to \{\mathbb{T}, \mathbb{F}\}\}$

Ergo: Solution is then

$$\varphi \equiv \psi \text{ iff for any } \mathcal{I} \in \mathscr{I}_{Var(\varphi) \cup Var(\psi)} : f_\varphi(\mathcal{I}) = f_\psi(\mathcal{I})$$

But as mentioned above, this is inconvenient to write, so this is often omitted.

2.2.7 Simplification rules

- Commutativity: $\begin{cases} \varphi \wedge \psi \equiv \psi \wedge \varphi \\ \text{(same with } \vee\text{)} \end{cases}$
- Associativity: $\begin{cases} \varphi_1 \wedge (\varphi_2 \wedge \varphi_3) \equiv (\varphi_1 \wedge \varphi_2) \wedge \varphi_3 \\ \text{(same with } \vee\text{)} \end{cases}$
- Idempotence: $\begin{cases} \varphi \wedge \varphi \equiv \varphi \\ \text{(same with } \vee\text{)} \end{cases}$

This can be extended for more complex expressions, for example $(A \vee ((A \vee B) \vee (C \vee B)))\vee C \equiv (A \vee B \vee C)$:

$$\begin{aligned} (((\vartheta_1 \vee \vartheta_2) \vee \vartheta_3) \vee \vartheta_4) \vee \vartheta_5 &\equiv \vartheta_1 \vee \vartheta_2 \vee \vartheta_3 \vee \vartheta_4 \vee \vartheta_5 \\ \text{this can also be notated} &: \bigvee \{\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5\} \end{aligned}$$

- Properties of negation

$$\begin{aligned} \neg\neg\varphi &\equiv \varphi \\ \neg\varphi \vee \varphi &\equiv \mathbb{T} \\ \neg\varphi \wedge \varphi &\equiv \mathbb{F} \\ \neg(\varphi \vee \psi) &\equiv (\neg\varphi) \wedge (\neg\psi) \\ \neg(\varphi \wedge \psi) &\equiv (\neg\varphi) \vee (\neg\psi) \end{aligned}$$

- Distributivity

$$\begin{aligned} \varphi \wedge (\psi_1 \vee \psi_2) &\equiv (\varphi \wedge \psi_1) \vee (\varphi \wedge \psi_2) \\ \varphi \vee (\psi_1 \wedge \psi_2) &\equiv (\varphi \vee \psi_1) \wedge (\varphi \vee \psi_2) \end{aligned}$$

- Elimination of $\Rightarrow$ and $\Leftrightarrow$

$$\begin{aligned} \varphi \Rightarrow \psi &\equiv (\neg\varphi) \vee \psi \\ \varphi \Leftrightarrow \psi &\equiv (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi) \\ &\equiv (\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi) \\ &\equiv (\varphi \wedge \psi) \vee (\neg\varphi \wedge \neg\psi) \end{aligned}$$

Example usage of rules:

$$\begin{aligned} A \vee B \vee \neg A &\equiv \\ (A \vee \neg A) \vee B &\equiv \\ \mathbb{T} \vee B &\equiv \mathbb{T} \end{aligned}$$

- Properties of truth constants

$$\begin{aligned} \mathbb{T} \vee \varphi &\equiv \mathbb{T} & \mathbb{F} \vee \varphi &\equiv \varphi \\ \mathbb{T} \wedge \varphi &\equiv \varphi & \mathbb{F} \wedge \varphi &\equiv \mathbb{F} \\ \neg\mathbb{T} &\equiv \mathbb{F} & \neg\mathbb{F} &\equiv \mathbb{T} \end{aligned}$$

$$\begin{aligned} \mathbb{T} \Rightarrow \varphi &\equiv \varphi & \mathbb{F} \Rightarrow \varphi &\equiv \mathbb{T} \text{ (false implies anything)} \\ \text{(because } \mathbb{T} \Rightarrow \varphi &\equiv (\neg\mathbb{T}) \vee \varphi \equiv \mathbb{F} \vee \varphi \equiv \varphi) & \text{(because } \mathbb{F} \Rightarrow \varphi &\equiv (\neg\mathbb{F}) \vee \varphi \equiv \mathbb{T} \vee \varphi \equiv \mathbb{T}) \end{aligned}$$

$$\begin{aligned} \varphi \Rightarrow \mathbb{T} &\equiv \mathbb{T} & \varphi \Rightarrow \mathbb{F} &\equiv \neg\varphi \end{aligned}$$

Note: Homework 2.1 was to write the rules for the elimination of the truth constants for $\Leftrightarrow$ (in analogy to $\varphi \Rightarrow \mathbb{T} \equiv \mathbb{T}$ and $\varphi \Rightarrow \mathbb{F} \equiv \neg\varphi$)

2.3 Proof

For example, prove $\mathbb{T} \vee \varphi \equiv \mathbb{T}$

By definition:

$$\begin{aligned} \text{For any } \mathcal{I}: & \quad & & \text{(note that again, this only refers to} \\ & & & \text{the relevant interpretations as explained above)} \\ f_{\mathbb{T}\vee\varphi}(\mathcal{I}) &=^{?} & f_{\mathbb{T}}(\mathcal{I}) & \\ \langle \mathbb{T} \vee \varphi \rangle_{\mathcal{I}} &=^{?} & \langle \mathbb{T} \rangle_{\mathcal{I}} & \\ \mathscr{B}_{\vee}(\langle \mathbb{T} \rangle_{\mathcal{I}}, \langle \varphi \rangle_{\mathcal{I}}) &=^{?} & \langle \mathbb{T} \rangle_{\mathcal{I}} & \\ &=_{\text{Yeah!}} & \mathbb{T} & \end{aligned}$$

$\mathscr{B}_\vee$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	Either this case:$\mathbb{T}$	or this case:$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$

Proof for $\neg\neg\varphi \equiv \varphi$

$$\begin{aligned} \langle\neg\neg\varphi\rangle_\mathcal{I} &=^? \langle\varphi\rangle_\mathcal{I} \\ \langle\neg\neg\varphi\rangle_\mathcal{I} &\equiv \mathscr{B}_\neg(\langle\neg\varphi\rangle_\mathcal{I}) \\ &\equiv \mathscr{B}_\neg(\mathscr{B}_\neg(\langle\varphi\rangle_\mathcal{I})) \equiv^! \langle\varphi\rangle_\mathcal{I} \end{aligned}$$

$$\mathscr{B}_\neg(\mathscr{B}_\neg(c)) \equiv^! c \quad \text{case} \quad \begin{cases} c = \mathbb{T} & \mathscr{B}_\neg(\mathscr{B}_\neg(\mathbb{T})) \equiv_{\text{Yeah!}} \mathbb{T} \\ c = \mathbb{F} & \mathscr{B}_\neg(\mathscr{B}_\neg(\mathbb{F})) \equiv_{\text{Yeah!}} \mathbb{F} \end{cases}$$

Another way to prove, is shown here with the example of proving $\varphi \wedge (\psi_1 \vee \psi_2) \equiv (\varphi \wedge \psi_1) \vee (\varphi \wedge \psi_2)$

$\langle\varphi\rangle_\mathcal{I}$	$\langle\psi_1\rangle_\mathcal{I}$	$\langle\psi_2\rangle_\mathcal{I}$	$\psi_1 \vee \psi_2$	LHS	$\varphi \wedge \psi_1$	$\varphi \wedge \psi_2$	RHS
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	...	...	...	...	...
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$	...	...	...	...	...
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	...	...	...	...	...
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	...	...	...	...	...
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$	...	...	...	...	...
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	...	...	...	...	...

Note: Completing this table was the second part of Homework 2

Note: LHS means Left Hand Side, RHS means Right Hand Side

2.3.1 Strategies

$$\text{Eliminate } \neg \begin{cases} \neg\neg\varphi \equiv \varphi \\ \neg\varphi \vee \varphi \equiv \mathbb{T} \\ \neg\varphi \wedge \varphi \equiv \mathbb{F} \end{cases}$$

$$\begin{array}{c}\text{Push negation:} \\ \text{(Negation only directly before a literal)}\end{array} \quad \begin{cases} \neg(\varphi \vee \psi) & \equiv & (\neg\varphi) \wedge (\neg\psi) \\ \neg(\varphi \wedge \psi) & \equiv & (\neg\varphi) \vee (\neg\psi) \end{cases}$$

$$\begin{array}{c}\text{Distributivity} \\ \text{("ausmultiplizieren")}\end{array} \quad \left\{ \varphi \vee (\psi_1 \wedge \psi_2) \underset{\rightarrow}{\equiv} (\varphi \vee \psi_1) \wedge (\varphi \vee \psi_2) \right.$$

Note: Every formula can be transformed to a *conjunction of disjunctions of literals* **!**

Conjunctive Normal Form (CNF): $\underline{(\ldots \vee \ldots \vee \ldots)} \wedge \ldots \wedge \underline{(\ldots \vee \ldots \vee \ldots)}$

Note: Each of these underlined *"disjunction of literals"* is called *"clause"*, so a formula in CNF is a *conjunction of clauses*.

$$
\begin{aligned}
(A \wedge (A \Rightarrow B)) \Rightarrow B &\equiv \text{(Replace implications...)} \\
\neg(A \wedge (\neg A \vee B)) \vee B &\equiv \text{(Push Negation/"De Morgan"...)} \\
(\neg A \vee \neg(\neg A \vee B)) \vee B &\equiv \text{(Push Negation ...)} \\
(\neg A \vee (\neg\neg A \wedge \neg B)) \vee B &\equiv \text{(Distributivity} \rightarrow \text{Second solution below)} \\
(\underbrace{(\neg A \vee A)}_{\mathbb{T}} \wedge (\neg A \vee \neg B)) \vee B &\equiv \\
(\neg A \vee \underbrace{(\neg B \vee B)}_{\mathbb{T}}) &\equiv \mathbb{T}
\end{aligned}
$$

Second solution:

$$
\begin{aligned}
(\neg A \vee (\neg\neg A \wedge \neg B)) \vee B &\equiv \text{(Distributivity, other possiblity)} \\
(\neg A \vee ((A \vee B) \wedge \underbrace{(\neg B \vee B)}_{\mathbb{T}})) &\equiv \\
\underbrace{\underbrace{(\neg A \vee A)}_{\mathbb{T}} \vee B}_{\mathbb{T}} &\equiv \mathbb{T}
\end{aligned}
$$

2.3.2 What is equivalence ?

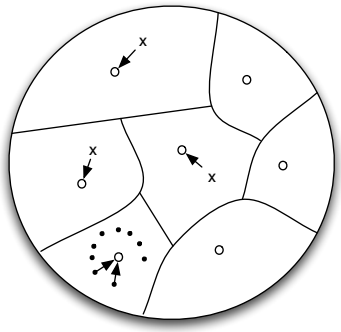


Figure 2.2: Equivalence classes

"From every element in an equivalence class we obtain the same formula if we apply our transformations over and over again" $\simeq$ "Normal form"

Assume that $\varphi = \underline{(A)} \wedge \underline{(\neg A \vee B)} \wedge \underline{(\neg B \vee C)} \wedge \underline{(\neg C)} \equiv \mathbb{F}$ holds in some $\mathcal{I} = \{A \rightarrow \mathbb{T}, B \rightarrow \mathbb{T}, C \rightarrow \mathbb{T}, D \rightarrow \mathbb{T}\}$

Note: This formula is in CNF and has 2 clauses with 2 literals and 2 clauses with 1 literal.

But: Even if this formula is in normal form, there is still an equivalent and more minimal expression which is also in normal form !!!

2.3.3 Proof Systems

Steps for example proof:

- Prove: $(A \wedge (A \Rightarrow B)) \Rightarrow B$
- "deduction rule"
  - Assume: $A \wedge (A \Rightarrow B)$
    - this is decomposed into parts ...
      - Assume: A
      - Assume: $A \Rightarrow B$
    - ... by "modus ponens":
      - Know: B
  - Prove: B

the *formal steps for this proof*:

$$
\begin{array}{rl}
\{\} \vdash (A \wedge (A \Rightarrow B) \Rightarrow B & \text{("Proof situation")} \\
\downarrow_1 & \\
\{A \wedge (A \Rightarrow B)\} \vdash B & \\
\downarrow_2 & \\
\{A, A \Rightarrow B\} \vdash B & \\
\downarrow_3 & \\
\{A, A \Rightarrow B, \underline{B}\} \vdash \underline{B} & \\
\downarrow_4 & \\
\mathbb{T} &
\end{array}
$$

General: *"rewriting rules"*

Note: $\Phi \vdash \Psi$ is called a *"sequent"*

$$
\text{R1 (rule 1)} \quad = \quad \left\{ \begin{array}{ccc} \Phi & \vdash & \psi_1 \Rightarrow \psi_2 \\ & \downarrow & \\ \Phi \cup \{\psi_1\} & \vdash & \psi_2 \end{array} \right.
$$

$$
\text{R2 (rule 2)} \quad = \quad \left\{ \begin{array}{ccc} \Phi \cup \{\varphi_1 \wedge \varphi_2\} & \vdash & \Psi \\ & \downarrow & \\ \Phi \cup \{\varphi_1, \varphi_2\} & \vdash & \Psi \end{array} \right.
$$

$$
\begin{array}{c} \text{R3 (rule 3)} \\ \text{modus ponens} \end{array} \quad = \quad \left\{ \begin{array}{ccc} \Phi \cup \{\varphi_1, \varphi_1 \Rightarrow \varphi_2\} & \vdash & \Psi \\ & \downarrow & \\ \Phi \cup \{\varphi_1, \varphi_1 \Rightarrow \varphi_2, \varphi_2\} & \vdash & \Psi \end{array} \right.
$$

$$\text{R4 (rule 4)} \quad = \quad \begin{cases} \Phi \vdash \psi \\ \quad \downarrow \quad \text{if } \psi \in \Phi \\ \quad \mathbb{T} \end{cases}$$

Proofs are written like this (from below to above, but without the arrow):

$$\uparrow \qquad \frac{\dfrac{\dfrac{\dfrac{\mathbb{T}}{A, A \Rightarrow B, B \vdash B}}{A, A \Rightarrow B \vdash B}}{A \wedge (A \Rightarrow B) \vdash B}}{\vdash (A \wedge (A \Rightarrow B)) \Rightarrow B}$$

("premises")

Inference rules are written like that:

$$\frac{\Phi, \psi_1 \vdash \psi_2}{\Phi \vdash \psi_1 \Rightarrow \psi_2} \qquad \text{(R1)}$$

$$\frac{\Phi, \varphi_1, \varphi_2 \vdash \Psi}{\Phi, \varphi_1 \wedge \varphi_2 \vdash \Psi} \qquad \text{(R2)}$$

$$\frac{\Phi, \varphi_1, \varphi_1 \Rightarrow \varphi_2, \varphi_2 \vdash \Psi}{\Phi, \varphi_1, \varphi_1 \Rightarrow \varphi_2 \vdash \Psi} \qquad \text{(R3)}$$

... and read like that:

$$\frac{\text{If we are able to prove this}}{\text{then we can be sure that this holds}}$$

Keep in mind:

- We want to develop rules that help us to construct proofs
- To actually construct a proof one must combine these rules
- In this LVA, we don't try to automate this (but there is research on that topic)

$$\Phi, \varphi \vdash \varphi \text{ "is axiom"}$$

2.3.4 Informal Proof Tree

Prove: $\underbrace{B \vee C, B \Rightarrow A, C \Rightarrow D}_{\text{Assume}} \vdash \underbrace{A \vee D}_{\text{Prove}}$

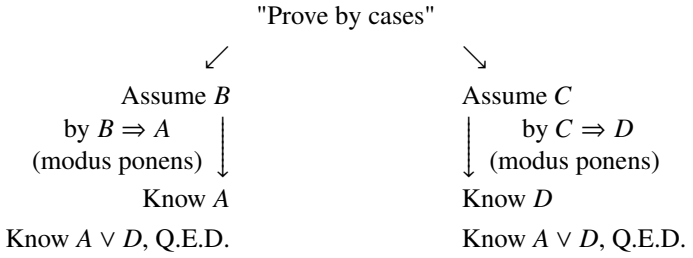


Figure 2.3: Informal Proof Tree

2.3.5 Sequent

$$\Phi, \varphi \quad \vdash \quad \varphi \vee \psi$$

considered as conjunction		considered as disjunction
Φ, φ	$\vdash$	φ, ψ "Axiom"
Φ, φ	$\vdash$	Ψ, φ "Axiom"

So we consider working with two sets:

"Sequent":	Φ	$\vdash$	Ψ
	conjunction		disjunction

Definition: A *"Sequent"* is a pair of two formulae.
The set on the LHS is a conjunction
The set on the RHS is a disjunction

The sequent $\Phi \vdash \Psi$ holds iff $\bigwedge \Phi \Rightarrow \bigvee \Psi$

Note: $\bigwedge$ (analogically $\bigvee$) means the AND- (analogically OR-)Operation over all set elements

$$\varphi_1, \ldots, \varphi_n \vdash \psi_1, \ldots, \psi_m \text{ holds}$$
$$\text{iff } (\varphi_1 \wedge \cdots \wedge \varphi_n) \Rightarrow (\psi_1 \vee \cdots \vee \psi_m) \text{ is valid}$$

2.3.6 Proof Tree Notation

Now we reformulate the proof tree (given in 2.3) with a new notation:

$$\dfrac{\dfrac{\underline{B, A, B \Rightarrow A, C \Rightarrow D \vdash A, D}\ (\text{"axiom"})}{B, A, B \Rightarrow A, C \Rightarrow D \vdash A \vee D} \qquad \dfrac{\underline{C, D, B \Rightarrow A, C \Rightarrow D \vdash A, D}\ (\text{"axiom"})}{C, D, B \Rightarrow A, C \Rightarrow D \vdash A \vee D}}{B, B \Rightarrow A, C \Rightarrow D \vdash A \vee D} \qquad \dfrac{}{C, B \Rightarrow A, C \Rightarrow D \vdash A \vee D}$$
$$\dfrac{}{B \vee C, B \Rightarrow A, C \Rightarrow D \vdash A \vee D}$$

$$\frac{\Phi, \varphi_1 \vdash \Psi \qquad \Phi, \varphi_2 \vdash \Psi}{\Phi, \varphi_1 \vee \varphi_2 \vdash \Psi}$$

Figure 2.4: Rule for cases ("Fallunterscheidung") $(\vee \vdash)$

$$\frac{\Phi \vdash \Psi, \psi_1, \psi_2}{\Phi \vdash \Psi, \psi_1 \vee \psi_2}$$

Figure 2.5: R2 $(\vdash \vee)$

As already mentioned above (see 2.3.5 on the previous page), for a sequent $\Phi \vdash \Psi$:

$$\begin{gathered}\varphi_1, \ldots, \varphi_n \vdash \psi_1, \ldots, \psi_m \text{ holds} \\ \text{iff } (\varphi_1 \wedge \cdots \wedge \varphi_n) \Rightarrow (\psi_1 \vee \cdots \vee \psi_m) \text{ is valid}\end{gathered}$$

so we can say by applying rule $(A \vdash B) \equiv (\neg A \vee B)$

$$\begin{gathered}\varphi_1, \ldots, \varphi_n \vdash \psi_1, \ldots, \psi_m \text{ holds} \\ \text{iff } (\neg\varphi_1 \vee \cdots \vee \neg\varphi_n) \vee (\psi_1 \vee \cdots \vee \psi_m) \text{ is valid}\end{gathered}$$

So we see that

$$\begin{array}{ccc}\bigwedge \Phi & \Rightarrow & \bigvee \Psi \\ & \equiv & \\ \neg\Phi & \vee & \Psi\end{array}$$

This also enables us to move (negated) clauses from Φ to Ψ (and vice versa) because:

$$\begin{array}{rcl}(\neg\varphi_1 \vee \cdots \vee \neg\varphi_n \vee \neg\psi_1) & \vee & (\psi_2) \\ & \downarrow & \text{(moving } \neg\psi_1\text{)} \\ (\neg\varphi_1 \vee \cdots \vee \neg\varphi_n) & \vee & (\neg\psi_1 \vee \psi_2)\end{array}$$

2.3.7 Example proof

Prove $((A \vee B) \Rightarrow C) \Rightarrow ((A \Rightarrow C) \wedge (B \Rightarrow C))$
Assume $(A \vee B) \Rightarrow C$
Prove $(A \Rightarrow C) \wedge (B \Rightarrow C)$

$\swarrow$ $\qquad\qquad$ $\searrow$

Prove $A \Rightarrow C$
Assume A, Know: $A \vee B$ $\xrightarrow{\text{by MP}}$ C
Prove C

Prove $B \Rightarrow C$
Assume B, Know: $A \vee B$ $\xrightarrow{\text{by MP}}$ C
Prove C

Note: MP means "modus ponens" here

$$\frac{\Phi \vdash \Psi, \psi_1 \qquad \Phi \vdash \Psi, \psi_2}{\Phi \vdash \Psi, \psi_1 \wedge \psi_2}$$

Figure 2.6: Rule $(\vdash \wedge)$

Proof tree with the new notation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\neg B, A \vdash A, C \text{ (“axiom”)}}{\neg A, \neg B, A \vdash C}}{\neg A \wedge \neg B, A \vdash C} \qquad C, A \vdash C \text{ (“axiom”)}}{\neg(A \vee B) \vee C, A \vdash C}}{(A \vee B) \Rightarrow C, A \vdash C}}{(A \vee B) \Rightarrow C \vdash A \Rightarrow C} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\neg A, B \vdash B, C \text{ (“axiom”)}}{\neg A, \neg B, B \vdash C}}{\neg A \wedge \neg B, B \vdash C} \qquad C, B \vdash C \text{ (“axiom”)}}{\neg(A \vee B) \vee C, B \vdash C}}{(A \vee B) \Rightarrow C, B \vdash C}}{(A \vee B) \Rightarrow C \vdash B \Rightarrow C}$$

$$((A \vee B) \Rightarrow C) \vdash ((A \Rightarrow C) \wedge (B \Rightarrow C))$$

Note: Homework 2.3 will be to complete the tree on the right side

2.3.8 Summary

- *”sequent”:* $\Phi \vdash \Psi$
- *“sequent holds”:* $\bigwedge \Phi \Rightarrow \bigvee \Psi$
- *“inference rule”:* $\frac{S_1 \ldots S_n}{S}$ iff (if $S_1, \ldots, S_n$ hold, then S holds)
  - Set of inference rules: *“calculus”*
- *proof (tree):* tree
  - a *tree* is a graph with properties
    - root (is ancestor of all nodes)
    - leaves (nodes which have no successors)
    - some nodes have successors
  - a tree *“is a proof of S”* iff
    1. S is the root of the tree
    2. Whenever $S_1, \ldots, S_n$ are successors of a node S they are an instance of an inference rule.
    3. The leaves are the axioms

2.3.9 Calculus: “The small calculus”

2.3.9.1 Calculus definition

$\neg, \wedge, \vee$

	premises		conclusions	
$\neg$	$\dfrac{\Phi \vdash \Psi, \varphi}{\Phi, \neg\varphi \vdash \Psi}$	$(\neg \vdash)$	$\dfrac{\Phi, \psi \vdash \Psi}{\Phi \vdash \Psi, \neg\psi}$	$(\vdash \neg)$
$\wedge$	$\dfrac{\Phi, \varphi_1, \varphi_2 \vdash \Psi}{\Phi, \varphi_1 \wedge \varphi_2 \vdash \Psi}$	$(\wedge \vdash)$	$\dfrac{\Phi \vdash \Psi, \psi_1 \qquad \Phi \vdash \Psi, \psi_2}{\Phi \vdash \Psi, \psi_1 \wedge \psi_2}$	$(\vdash \wedge)$
$\vee$	$\dfrac{\Phi, \varphi_1 \vdash \Psi \qquad \Phi, \varphi_2 \vdash \Psi}{\Phi, \varphi_1 \vee \varphi_2 \vdash \Psi}$	$(\vee \vdash)$	$\dfrac{\Phi \vdash \Psi, \psi_1, \psi_2}{\Phi \vdash \Psi, \psi_1 \vee \psi_2}$	$(\vdash \vee)$

Axioms:

- $\Phi, \varphi \vdash \Psi, \varphi$
- $\Phi \vdash \Psi$ is axiom iff $\Phi \cap \Psi \neq \emptyset$

2.3.9.2 Calculus properties

- *"correct"* iff (if a sequent has a proof, then the sequent holds)
- *"complete"* iff (if a sequent holds, then it has a proof)

This can also be expressed in traditional logic:

$$\Phi \vdash \Psi \quad \text{iff} \quad \Phi \models \Psi$$

Note: Homework 2.4 will be to prove informally: $\Phi \vdash \Psi$ holds iff $\Phi \models \Psi$

Note: $\Phi \models \Psi$ can be read as "is a semantical logical consequence"

Chapter 3

Third Lecture on 27.10.2005

3.1 Recapitualation of 1st Homework

3.1.1 Task 2 (Grammar for propositional Logic)

$$\mathscr{L} \quad = \quad \begin{cases} \mathbb{T}, \mathbb{F}, \vartheta \in \mathscr{L}, \vartheta \in \Theta \\ (\neg\varphi), (\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \Rightarrow \psi), (\varphi \Leftrightarrow \psi) \in \mathscr{L} \text{ , where } \varphi, \psi \in \mathscr{L} \end{cases}$$

"These are all"

Now let's define the grammar:

$$P \quad = \begin{cases} W \to \mathbb{T} \mid \mathbb{F} \mid A \mid B \mid C \quad \text{, or more general: } \Theta = \{A, B, C\} \quad W \to v, \text{ for each } v \in \{\mathbb{T}, \mathbb{F}\} \cup \Theta \\ W \to (\neg W) \mid (W \wedge W) \mid \ldots \end{cases}$$

$$\text{Grammar } G \quad = \quad (\underbrace{\Sigma}_{\text{"alphabet"}} , \underbrace{\Sigma_N}_{\substack{\text{"nonterminal} \\ \text{symbols"}}} , \underbrace{S}_{\substack{\text{"nonterminal} \\ \text{start symbol"}}} , \underbrace{P}_{\substack{\text{set of} \\ \text{productions}}})$$

$$\begin{aligned} \Sigma &= \{\mathbb{T}, \mathbb{F}\} \cup \Theta \cup \{(,), \neg, \vee, \wedge, \Rightarrow, \Leftrightarrow\} \\ \Sigma_N &= \{W\} \\ S &= \Sigma_N \end{aligned}$$

f.ex. $\alpha W \beta \Rightarrow \alpha(\neg W)\beta$ $\qquad \mathscr{L} = \{\alpha \mid W \Rightarrow^* \mathscr{L}\}$

(the language is the transitive closure ("transitive Hülle"))

3.1.2 Task 1 (Comments and Questions)

Some answers were provided to comments

TODO: ... insert all of these answers

One answer about the usage of some symbols in this course:

iff	...	if and only if (logical meta-level equivalence)
$\equiv$ / $\not\equiv$	...	semantical equivalence of formulae
$\Leftrightarrow$	...	object level equivalence
$\rightsquigarrow$	...	"is transformed into", also "from something we can obtain something else"

3.2 Propositional Logic - Recapitulation

- Syntax / Semantics (Notions of interpretation and truth evaluation)
- Valid / Unsatisfiable, $\models$ (semantical logicial consequence), $\equiv$ (semantical logical equivalence)
- Rewriting, f.ex. $(A \wedge B) \vee C \rightsquigarrow \underbrace{(A \vee C)}_{\text{clause}} \wedge \underbrace{(B \vee C)}_{\text{clause}}$
  - CNF: conjunction of disjunction of literals
- proof systems
  - examples
  - sequents, inference rules
  - proof trees
  - “calculus” (the “short”)

3.2.1 Sequents

“Prove a formula φ“: show that φ is valid

In practice, our formulae which we want to prove are of this type:

$$\varphi_1 \wedge \cdots \wedge \varphi_n \quad \Rightarrow \quad \psi \quad \text{"proof situation"}$$

Note: In $\varphi_1, \ldots, \varphi_n \vdash \psi$, the symbol $\vdash$ is only “notation” and “has no meaning”, because it has no definition. It is just a notation for $\langle \{\varphi_1, \ldots, \varphi_n\}, \psi \rangle$. Later we will overload this symbol.

But on the other hand, $\models$ has a definition:

$$\begin{array}{rc} \text{if} & A \wedge B \models B \\ \text{then :} & \text{For all } \mathcal{I}, \text{ if } \langle A \wedge B \rangle_{\mathcal{I}} = \mathbb{T} \\ & \text{then } \langle B \rangle_{\mathcal{I}} = \mathbb{T} \end{array}$$

The sequent $\Phi \vdash \Psi$ holds $\underbrace{\text{iff } \Phi \models \Psi}$

$$\text{iff } (\bigwedge \Phi \Rightarrow \bigvee \Psi) \text{ is valid}$$

$$(\{\varphi_1, \ldots, \varphi_n\} \vdash \{\psi_1, \ldots, \psi_m\}) \text{ holds}$$
$$\text{iff}$$
$$((\varphi_1 \wedge \cdots \wedge \varphi_n) \models (\psi_1 \vee \cdots \vee \psi_m)) \text{ is valid}$$

Problem: The sets of the sequent could also be empty sets, f.ex. $\{A, \neg A\} \vdash \{\}$, and we should also consider this case !

$\bigvee$	$\langle \bigvee \{\psi_1, \psi_2, \psi_3\} \rangle_{\mathcal{I}} = \mathbb{T}$ iff $\langle \psi_1 \rangle_{\mathcal{I}} = \mathbb{T}$ or $\langle \psi_2 \rangle_{\mathcal{I}} = \mathbb{T}$ or $\langle \psi_3 \rangle_{\mathcal{I}} = \mathbb{T}$	$\langle \bigvee \Psi \rangle_{\mathcal{I}} = \mathbb{T}$ iff for some $\psi \in \Psi, \langle \psi \rangle_{\mathcal{I}} = \mathbb{T}$
	So in case $\bigvee\{\}$ { there exists $\psi \in \Psi$ s.t. $\langle \psi \rangle_{\mathcal{I}} = \mathbb{T}$ ↓ false ! $\psi \in \emptyset$ s.t. … } ergo $\bigvee\{\} = \mathbb{F}$	
$\bigwedge$	$\langle \bigwedge \{\psi_1, \psi_2, \psi_3\} \rangle_{\mathcal{I}} = \mathbb{T}$ iff $\langle \psi_1 \rangle_{\mathcal{I}} = \mathbb{T}$ and $\langle \psi_2 \rangle_{\mathcal{I}} = \mathbb{T}$ and $\langle \psi_3 \rangle_{\mathcal{I}} = \mathbb{T}$	$\langle \bigwedge \Psi \rangle_{\mathcal{I}} = \mathbb{T}$ iff for all $\psi \in \Psi, \langle \psi \rangle_{\mathcal{I}} = \mathbb{T}$
	So in case $\bigwedge\{\}$ { for all $\psi \in \Psi$: $\langle \psi \rangle_{\mathcal{I}} = \mathbb{T}$ ↓ true ! this is always satisfied } ergo $\bigwedge\{\} = \mathbb{T}$	

$\bigvee \Psi$	$(\exists \psi : \psi \in \Psi) \langle \psi \rangle_{\mathcal{I}} = \mathbb{T}$	$(\exists \psi) : \psi \in \Psi \wedge \langle \psi \rangle_{\mathcal{I}} = \mathbb{T}$
$\bigwedge \Psi$	$(\forall \psi : \psi \in \Psi) \langle \psi \rangle_{\mathcal{I}} = \mathbb{T}$	$(\forall \psi) : \psi \in \Psi \Rightarrow \langle \psi \rangle_{\mathcal{I}} = \mathbb{T}$

Analogy:

$\exists x \in \mathbb{R} : x \in \mathbb{Q}$	$(\exists x) x \in \mathbb{R} \wedge x \in \mathbb{Q}$	incorrect: ~~$(\exists x) x \in \mathbb{R} \Rightarrow x \in \mathbb{Q}$~~ ~~$(\exists x) x \notin \mathbb{R} \vee x \in \mathbb{Q}$~~
$\forall x \in \mathbb{Q} : x \in \mathbb{R}$	$(\forall x) x \in \mathbb{Q} \Rightarrow x \in \mathbb{R}$	incorrect: ~~$(\forall x) x \in \mathbb{Q} \wedge x \in \mathbb{R}$~~

$\langle \bigvee \Phi \rangle_{\mathcal{I}} = \mathbb{T}$ iff $\exists \varphi \in \Phi : \langle \varphi \rangle_{\mathcal{I}} = \mathbb{T}$ $\quad (\exists \varphi) \varphi \in \Phi \wedge \langle \varphi \rangle_{\mathcal{I}} = \mathbb{T}$

$\langle \bigvee \{\} \rangle_{\mathcal{I}} = \mathbb{F}$ $\quad (\exists \varphi) \underbrace{\varphi \in \{\}}_{\mathbb{F}} \wedge \ldots \Big\} \mathbb{F}$

$\langle \bigwedge \Phi \rangle_{\mathcal{I}} = \mathbb{T}$ iff $\forall \varphi \in \Phi : \langle \varphi \rangle_{\mathcal{I}} = \mathbb{T}$ $\quad (\forall \varphi) \varphi \in \Phi \Rightarrow \langle \varphi \rangle_{\mathcal{I}} = \mathbb{T}$

$\langle \bigwedge \{\} \rangle_{\mathcal{I}} = \mathbb{T}$ $\quad (\forall \varphi) \underbrace{\varphi \in \{\}}_{\mathbb{F}} \Rightarrow \ldots \Big\} \mathbb{T}$

Sequent $\Phi \vdash \Psi$

↙ ↘

if left side is empty: $\{\} \vdash \Psi$ also written: $\vdash \Psi$ (this is like having no assumptions)

if right side is empty: $\Phi \vdash \{\}$ also written: $\Phi \vdash$

$$
\begin{array}{rcl}
(\Phi \vdash \Psi) \text{ holds} & \text{iff} & \left(\bigwedge \Phi \Rightarrow \bigvee \Psi\right) \text{ is valid} \\
(\{\varphi_1, \ldots, \varphi_n\} \vdash \{\psi_1, \ldots, \psi_m\} \text{ holds}) & \text{iff} & \underbrace{((\varphi_1 \wedge \cdots \wedge \varphi_n) \Rightarrow (\psi_1 \vee \cdots \wedge \psi_m))} \text{ is valid} \\
 & & \underbrace{(\neg\,(\varphi_1 \wedge \cdots \wedge \varphi_n) \vee (\psi_1 \vee \cdots \wedge \psi_m))} \\
 & & (\neg\varphi_1 \vee \cdots \vee \neg\varphi_n \vee \psi_1 \vee \cdots \wedge \psi_m)
\end{array}
$$

3.2.2 Inference rules

We have introduced the notation $\frac{S_1 \ldots S_n}{S}$ with the meaning $\frac{\text{Set of sequents}}{\text{Sequent}}$.

$$
\frac{S_1 \ldots S_n}{S} \qquad \begin{array}{c}\text{("is an inference rule")} \\ \downarrow \end{array} \qquad \text{iff} \quad (\text{if } S_1, \ldots, S_n \text{ hold, then } S \text{ holds})
$$

3.2.3 Proof tree

- The root is a sequent (“assumptions”)
- The edges of the tree are inference rules
- The leaves must be axioms

$$
\dfrac{\dfrac{\dfrac{A_1''}{S_1''}}{\dfrac{S_1'}{S_1}} \quad \dfrac{\dfrac{A_2''}{S_2''}}{} \qquad \dfrac{\dfrac{A''}{S_2'}}{S_2}}{S}
$$

3.2.3.1 Tree

A *tree* is is an *oriented graph* $\langle G, \rightarrow\rangle$ where $\rightarrow \subseteq G \times G$

$$
x \overset{\text{"arrow"}}{\rightarrow} y \quad \begin{cases} x \text{ "is predecessor of" } y \\ y \text{ "is successor of" } x \end{cases}
$$

$$
Succ[x] \quad = \quad \{y \in G \,|\, x \rightarrow^* y\}
$$

Note: To be precise, $Succ[x]$ has an index $\langle G, \rightarrow\rangle$, but this is usually not written (implicit)

Tree (restrictions to the graph)

- there exists a root which is unique $t \in G,\ Succ[t] = G \backslash \{t\},\ Pred[t] = \{\}$
- any $t' \neq t : \exists! s : s \rightarrow t'$
- *“leaf”*: $Succ[x] = \{\}$

3.2.3.2 Proof tree as a special tree

Proof tree (with respect to a *set of inference rules* $\mathscr{R}$) of a sequent S:

"having the property that"
$\downarrow$

- Is a tree $\langle G, \rightarrow \rangle$:
  - G is a set of sequents
  - $\rightarrow$ is induced by $\mathscr{R}$: $\left\{ \begin{array}{l} \text{Whenever } \frac{S'_1 \ldots S'_n}{S'} \in \mathscr{R}, \\ \text{then } [S'] \rightarrow [S'_1], \ldots, [S'_n] \\ \text{(Notation: } \underbrace{[s]}_{s \in \{S', S'_1, \ldots, S'_n\}} \text{ are instances of } s) \end{array} \right.$
  - the root is S (the sequent which we want to prove)
  - the leaves are axioms

Note: $\left[\frac{\varphi_1, \varphi_2 \vdash \psi}{\varphi_1 \wedge \varphi_2 \vdash \psi}\right]$: $\frac{A, (B \Rightarrow C) \vdash C}{A \wedge (B \Rightarrow C) \vdash C}$
"instance"

3.2.4 Calculus

A calculus $\mathscr{C} = \left\langle \underbrace{\mathscr{R}}_{\text{"inference rules"}}, \underbrace{\mathscr{A}}_{\text{"axioms"}} \right\rangle$

Theorem: If there is a proof tree for S, then S holds ! *("correctness of calculus")*

3.2.4.1 Short calculus

$\neg, \wedge, \vee$

$\mathscr{R}$ is given by the table:

	$\neg$	$\wedge$	$\vee$
assumptions	$\frac{\Phi \vdash \Psi, \varphi}{\Phi, \neg\varphi \vdash \Psi}\ (\neg \vdash)$	$\frac{\Phi, \varphi_1, \varphi_2 \vdash \Psi}{\Phi, \varphi_1 \wedge \varphi_2 \vdash \Psi}\ (\wedge \vdash)$	...
goals	$\frac{\Phi, \psi \vdash \Psi}{\Phi \vdash \Psi, \neg\psi}\ (\vdash \neg)$	$\frac{\Phi \vdash \Psi, \psi_1 \quad \Phi \vdash \Psi, \psi_2}{\Phi \vdash \Psi, \psi_1 \wedge \psi_2}\ (\vdash \wedge)$	...

$\mathscr{A} = \{\Phi \vdash \Psi \,|\, \Phi \cap \Psi \neq \{\}\}$

Note: Homework 3.1 was to formulate the rules for $\vee$

3.2.4.2 Correctness of an inference rule

Correctness of $(\neg \vdash)$

if $\underbrace{\Phi \vdash \Psi, \varphi}$ holds — then $\underbrace{\Phi, \neg\varphi \vdash \Psi}$ holds

$\neg\varphi_1 \vee \cdots \vee \neg\varphi_n \vee \varphi \vee \psi_1 \vee \cdots \vee \psi_m$ $\equiv$ $\neg\varphi_1 \vee \cdots \vee \neg\varphi_n \vee \underbrace{\neg\neg\varphi}_{\varphi} \vee \psi_1 \vee \cdots \vee \psi_m$

valid — $\uparrow$ works in other direction ! ("also reversible") — φ valid

Note: $\Phi = \{\varphi_1, \ldots, \varphi_n\}$ and $\Psi = \{\psi_1, \ldots, \psi_m\}$ could also be empty !

Note: Homework 3.2 was to prove the correctness and the reversibility of the rule $(\vdash \neg)$

Correctness of $(\wedge \vdash)$

$$\begin{gathered}(((\bigwedge \Phi) \wedge \varphi_1 \wedge \varphi_2) \Rightarrow \bigvee \Psi) \\ \equiv \\ (((\bigwedge \Phi) \wedge (\varphi_1 \wedge \varphi_2)) \Rightarrow \bigvee \Psi)\end{gathered}$$

Correctness of $(\vdash \wedge)$ This:

$$\left(\underbrace{\left(\bigwedge \Phi\right)}_{\alpha} \Rightarrow \underbrace{\left(\bigvee \Psi\right) \vee \psi_1}_{\beta_1} \right) \quad \begin{matrix}\wedge \\ \text{valid}\end{matrix} \quad \left(\underbrace{\left(\bigwedge \Phi\right)}_{\alpha} \Rightarrow \underbrace{\left(\bigvee \Psi\right) \vee \psi_2}_{\beta_2} \right)$$

$$\alpha \text{ valid and } \beta \text{ valid iff } (\alpha \wedge \beta) \text{ valid } \ldots \ \mathscr{B}_{\wedge}\left(\langle\alpha\rangle_I, \langle\beta\rangle_I\right)$$

can be transformed to:

$$\underbrace{\left(\bigwedge \Phi\right)}_{\alpha} \quad \Rightarrow \quad \underbrace{\left(\bigvee \Psi\right) \vee (\psi_1 \wedge \psi_2)}_{\beta_1 \wedge \beta_2}$$

because:

$$\begin{aligned}(\alpha \Rightarrow \beta_1) \wedge (\alpha \Rightarrow \beta_2) &\equiv \\ (\neg\alpha \vee \beta_1) \wedge (\neg\alpha \vee \beta_2) &\equiv \\ \neg\alpha \vee (\beta_1 \wedge \beta_2) &\equiv \quad \alpha \Rightarrow (\beta_1 \wedge \beta_2)\end{aligned}$$

Note: Homework 3.3 was to prove correctness for the rules of disjunction.

- Tip: use axioms and rules, f.ex:

$$\underbrace{\begin{gathered}\varphi \wedge (\bigwedge \Phi) \Rightarrow \varphi \vee (\bigvee \Psi) \\ \underline{\neg\varphi} \vee \neg\left(\bigwedge \Phi\right) \vee \underline{\varphi} \vee \left(\bigvee \Psi\right)\end{gathered}}_{\mathbb{T}}$$

Every $\underbrace{\text{"atom sequent"}}_{\underline{A}, B \vdash C, \underline{A}, \underline{B}}$ that holds, has this property: $\left\{ \Phi \vdash \Psi \;\middle|\; \underbrace{\Phi \cap \Psi \neq \emptyset}_{\begin{matrix}\text{f.ex. here} \\ \text{it is } \{A, B\}\end{matrix}} \right\}$

3.2.4.3 Completeness

$$\begin{array}{lll} \text{if } \Phi, \Psi \subseteq \Theta & \text{, then} & \\ \Phi \vdash \Psi \text{ holds} & \overset{\Leftarrow}{\text{iff}} & \Phi \cap \Psi \neq \emptyset \\ \Phi \vdash \Psi \text{ holds} & \overset{\Rightarrow}{\text{iff}} & (\Phi \vdash \Psi) \in \mathscr{A} \end{array}$$

$\Phi = \{A_1, \ldots, A_n\},\ \Psi = \{B_1, \ldots, B_m\}$

Assume $\Phi \vdash \Psi$ holds: $\neg A_1 \vee \cdots \vee \neg A_n \vee B_1 \vee \cdots \vee B_m$ is valid

By contradiction: Assume $\Phi \cap \Psi = \emptyset$

$$\mathcal{I} = \begin{cases} A_1 = \cdots = A_n = \mathbb{T} \\ B_1 = \cdots = B_m = \mathbb{F} \end{cases}$$

A sequent that holds, but is not an axiom: $\{A, \neg A\} \vdash \{\}$

$$\Phi \vdash \Psi$$

$$\begin{array}{lll} & & \varphi_1, \ldots, \varphi_m \vdash \psi_1, \ldots, \psi_m \\ & & \neg\varphi_1 \vee \cdots \vee \neg\varphi_m \vee \psi_1 \vee \cdots \vee \psi_m \\ \text{if } \Phi \text{ is the empty set,} & \text{then the } \varphi\text{'s are not present,} & \text{only } \psi_1 \vee \cdots \vee \psi_m \\ \text{if } \Psi \text{ is the empty set,} & \text{then the } \psi\text{'s are not present,} & \text{only } \neg\varphi_1 \vee \cdots \vee \neg\varphi_m \end{array}$$

3.2.4.4 Proof for correctness

$$\dfrac{\dfrac{\underline{\underline{A}} \vdash B, C, \underline{A}\ \text{(is axiom)}}{A, \underline{\neg} A \vdash B, C} \qquad A, \underline{B} \vdash \underline{B}, C\ \text{(is axiom)}}{A, (\neg A) \underline{\vee} B \vdash B, C}$$
$$\dfrac{}{A, (\neg A) \underline{\vee} B \vdash B \underline{\vee} C}$$

$$\begin{array}{lll} & \overset{\text{(is a)}}{\downarrow} & \\ \text{(If)} & & \\ \downarrow\ S & : & \text{sequent: there is a proof tree with axiom-sequents as leaves} \end{array}$$

S holds !, then: all leaves hold !

Thus all leaves are axioms

Hence, the tree is a proof tree

3.2.4.5 Number of logical connectives over sequents

$$\underbrace{L(S)}_{n} : \text{ the number of logical connectives}$$

if S holds, then there exists P proof tree for S

$n = 0$: S is atom-sequent: S is axiom!

$n > 0$: for all S' with $L(S) < n$, there exists P' proof tree for S'.
(holds)

S holds

$L(S) > 0$ there exists l logical connectives in S

by inspecting inference rules, there exists R : $\dfrac{S_1[S_2]}{S}$ $\begin{matrix}(\vdash l)\\ (l \vdash)\end{matrix}$

S_1, S_2 hold: $L(S_1) < n, L(S_2) < n$ (because l is eliminated)

By induction hypothesis there are $P_1[,\ P_2]$ proof trees for $S_1[,\ S_2]$

Thus we have P : $\dfrac{\begin{matrix}P_1 & \left[\begin{matrix}P_2\\ S_2\end{matrix}\right]\\ S_1 & \end{matrix}}{S}$ (proof tree for S)

For a sequent $\Phi \vdash \Psi$, we have seen that

$$\begin{matrix}\neg A_1 \vee \cdots \vee \neg A_n \vee B_1 \vee \cdots \vee B_m\\ \Phi = \{A_1, \ldots, A_n\},\ \Psi = \{B_1, \ldots, B_m\}\end{matrix}$$

and the interesting part of the proof ♣ was:

$$\left\{\begin{matrix}A_2 = \cdots = A_n = \mathbb{T}\\ B_1 = \cdots = B_m = \mathbb{F}\end{matrix}\right. \qquad A_1 =_{\sqrt{}} \mathbb{T} \qquad A_1 = \mathbb{F}$$

So:

$$\text{if } \Phi, \Psi \subseteq \Theta \cup \{\mathbb{T}, \mathbb{F}\}$$

$$\Phi \vdash \Psi \text{ holds, then } \Phi \vdash \Psi \in \mathscr{A} \text{ with } \left\{\begin{matrix}\Phi \cap \Psi \neq \emptyset\\ \mathbb{F} \in \Phi\\ \mathbb{T} \in \Psi\end{matrix}\right.$$

"Short calculus" for $\mathscr{L}$ with $\{\mathbb{T}, \mathbb{F}\}$:

$$\mathscr{A}' = \{\Phi \vdash \Psi \,|\, \Phi \cap \Psi \neq \{\} \text{ or } \mathbb{F} \in \Phi \text{ or } \mathbb{T} \in \Psi\}$$

3.2.4.6 Extending the calculus

"Short calculus" is nice, but one also wants other rules, so we add this rule to our calculus:

$$\varphi \Rightarrow \psi \equiv \neg\varphi \vee \psi$$

$$\frac{\Phi, \varphi' \vdash \Psi}{\Phi, \varphi \vdash \Psi} \quad \text{if} \quad \varphi \equiv \varphi' \ (\equiv\vdash), \text{ similar } (\vdash\equiv)$$

The calculus is still correct (because each rule is correct) and complete.

Now let's add implication:

implication rule (1st version)	$\dfrac{\Phi, \neg\varphi \vee \psi \vdash \Psi}{\Phi, \varphi \Rightarrow \psi \vdash \Psi}\ (\Rightarrow\vdash)$	$\dfrac{\Phi \vdash \Psi, \neg\varphi \vee \psi}{\Phi \vdash \Psi, \varphi \Rightarrow \psi}\ (\vdash\Rightarrow)$

We see that we can obtain a simplified version, which is still correct, because it was created by applying the existing correct rules:

creating by applying existing correct rules:	$\dfrac{\dfrac{\Phi \vdash \Psi, \varphi}{\Phi, \neg\varphi \vdash \Psi} \qquad \Phi, \psi \vdash \Psi}{\dfrac{\Phi, \neg\varphi \vee \psi \vdash \Psi}{\Phi, \varphi \Rightarrow \psi \vdash \Psi}}(\Rightarrow\vdash)$	$\dfrac{\dfrac{\Phi, \varphi \vdash \Psi, \psi}{\Phi \vdash \Psi, \neg\varphi, \psi}}{\dfrac{\Phi \vdash \Psi, \neg\varphi \vee \psi}{\Phi \vdash \Psi, \varphi \Rightarrow \psi}}(\vdash\Rightarrow)$

Simplyfied version:

implication rule (2nd version)	$\dfrac{\Phi \vdash \Psi, \varphi \quad \Phi, \psi \vdash \Psi}{\Phi, \varphi \Rightarrow \psi \vdash \Psi}\ (\Rightarrow\vdash)$	$\dfrac{\Phi, \varphi \vdash \Psi, \psi}{\Phi \vdash \Psi, \varphi \Rightarrow \psi}\ (\vdash\Rightarrow)$

This is an advantage of starting with a simple calculus and then adding rules.

One can also see it in the way, that one starts with a big calculus and then eliminates the rules that can be inferred by other rules, so one gets a "minimal" calculus.

Introduction of modus ponens

$$\frac{\overbrace{\Phi, \varphi \vdash \Psi, \varphi}^{\text{axiom}} \qquad \Phi, \varphi, \psi \vdash \Psi}{\Phi, \varphi, \varphi \Rightarrow \psi \vdash \Psi}$$

so this is eliminated, so that the rule for modus ponens is:

$$\frac{\Phi, \varphi, \psi \vdash \Psi}{\Phi, \varphi, \varphi \Rightarrow \psi \vdash \Psi}$$

Note: Homework 3.4 is to formulate two rules for equivalence $\Leftrightarrow$, and prove their correctness ("eliminate" them) using other rules.

Notion 1

This relation $\Phi \vdash_{\mathscr{C}} \Psi$ holds iff there exists a proof of $\Phi \vdash \Psi$ in $\mathscr{C}$

Note: $\vdash_{\mathscr{C}}$ means a "syntactical logical consequence" or also "syntactical entailment"

Notion 2

$$\varphi \quad \models \quad \psi$$

What we want is

$$\Phi \vdash_{\mathscr{C}} \Psi \quad \begin{array}{c}\overset{\text{"correctness"}}{\Rightarrow} \\ \underset{\text{"completeness"}}{\Leftarrow}\end{array} \quad \bigwedge \Phi \models \bigvee \Psi$$

Chapter 4

Fourth Lecture on 3.11.2005

4.1 Recapitulation of Homework 2

4.2 Proof system which is more suited for computer implementation

4.2.1 Proof using unsatisfiability

It is tedious to obtain the CNF in this form:

$$
\begin{gathered}
(A \Rightarrow P), (B \Rightarrow P) \models (A \vee B) \Rightarrow P \\
((A \Rightarrow P) \wedge (B \Rightarrow P)) \Rightarrow ((A \vee B) \Rightarrow P) \text{ is valid} \\
\neg((\neg A \vee P) \wedge (\neg B \vee P)) \vee (\neg(A \vee B) \vee P) \\
\neg(\neg A \vee P) \vee \neg(\neg B \vee P) \vee (\neg(A \vee B) \vee P) \\
(A \wedge \neg P) \vee (B \wedge \neg P) \vee ((\neg A \wedge \neg B) \vee P) \\
(A \wedge \neg P) \vee (B \wedge \neg P) \vee ((\neg A \vee P) \wedge (\neg B \vee P)) \\
\underline{(A \vee (B \wedge \neg P)) \wedge (\neg P \vee (B \wedge \neg P))} \vee \underline{(\neg A \vee (\neg B \vee P)) \wedge (P \vee (\neg B \vee P))}
\end{gathered}
$$

Another way is (similar to *proof by contradiction*):

showing:	$\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n \Rightarrow \psi$	is valid
by showing:	$\neg((\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n) \Rightarrow \psi)$	is unsatisfiable
	$\neg(\neg(\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n) \vee \psi)$	is unsatisfiable
	$\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n \wedge \neg\psi$	is unsatisfiable

$\mathscr{B}_{\Rightarrow}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$

Applying this to the above example:

showing:	$((A \Rightarrow P) \wedge (B \Rightarrow P)) \Rightarrow ((A \vee B) \Rightarrow P)$	is valid
by showing:	$(A \Rightarrow P) \wedge (B \Rightarrow P) \wedge \underbrace{\neg((A \vee B) \Rightarrow P)}_{\neg(\neg(A \vee B) \vee P)}$	is unsatisfiable
	$(\neg A \vee P) \wedge (\neg B \vee P) \wedge (A \vee B) \wedge \neg P$	is unsatisfiable
	$(\overline{A} \vee P) \wedge (\overline{B} \vee P) \wedge (A \vee B) \wedge \overline{P}$	in CNF !

CNF: conjunction of clauses, each clause is a disjunctive set of literals.

This could also be written this way:

$$\bigwedge \left\{ \bigvee \left\{ \overline{A}, P \right\}, \bigvee \left\{ \overline{B}, P \right\}, \bigvee \{A, B\}, \bigvee \left\{ \overline{P} \right\} \right\}$$

4.2.2 Example proof

For an arbitrary $\mathcal{I}$:

$$\begin{array}{lll} \left\langle \overline{A} \vee P \right\rangle_{\mathcal{I}} = \mathbb{T} & \text{, then (by (1)):} & \left\langle \overline{A} \right\rangle_{\mathcal{I}} = \mathbb{T}, \langle A \rangle_{\mathcal{I}} = \mathbb{F} \\ \overline{B} \vee P & \text{, then} \ldots & \langle B \rangle_{\mathcal{I}} = \mathbb{F} \\ A \vee B & \xrightarrow[\text{contradiction}]{} & \langle A \vee B \rangle_{\mathcal{I}} = \mathbb{F} \\ \left\langle \overline{P} \right\rangle_{\mathcal{I}} = \mathbb{T} & \text{, then } \langle P \rangle_{\mathcal{I}} = \mathbb{F} \ (1) & \end{array}$$

4.2.3 New scheme to write this proof

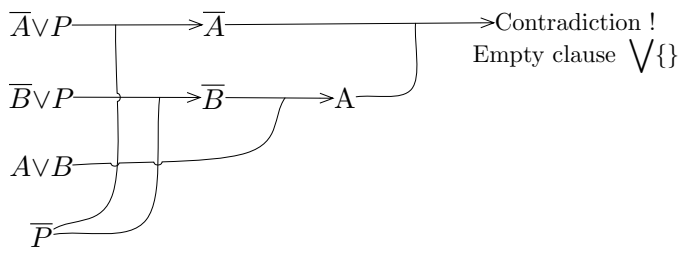


Figure 4.1: New schema to write this proof

Sequent calculus: "Gentzen style" $\Phi \vdash \Psi$

"Hilbert style": Set of assumptions Φ / ... / ... / ... / ... / ... / Ψ — new assumptions — inference rules — to show $\Phi \models \Psi$ — obtain a new formula from "known" other ones

Note: in "Hilbert style", the set of assumptions is never changed

4.2.3.1 Rules

Note: In the following rules C_1 etc. denote Clauses, whereas L denotes a literal, $\square$ denotes the empty clause

$$\left.\begin{array}{l} C_1 : L \vee C_1' \\ C_2 : \overline{L} \end{array}\right\} \ \mapsto \ C_1'$$

Can be used to delete the literal where the opposite is present:

$$\left.\begin{array}{l} A \vee B \\ \overline{A} \vee B \end{array}\right\} \ \mapsto \ B \qquad \left.\begin{array}{l} A \vee \overline{B} \\ \overline{A} \vee \overline{B} \end{array}\right\} \ \mapsto \ \overline{B} \qquad \mapsto \ \square$$

"Resolution inference rule":

$$\left.\begin{array}{l} C_1 : L \vee C_1' \\ C_2 : \overline{L} \vee C_2' \end{array}\right\} \quad \mapsto \quad \underbrace{C_1' \vee C_2'}_{\begin{array}{c}\text{"resolvent of the} \\ \text{clauses } C_1, C_2\text{"}\end{array}}$$

For example:

$$\left.\begin{array}{l} A \vee B \vee C \\ \overline{A} \vee B \vee P \vee Q \end{array}\right\} \quad \mapsto \quad B \vee C \vee P \vee Q$$

4.2.4 Correctness of the resolution rule

$$L \vee C_1', \overline{L} \vee C_2' \quad \models \quad C_1' \vee C_2'$$

Proof:

For an arbitrary interpretation $\mathcal{I}$

$$\langle L \vee C_1' \rangle_{\mathcal{I}} = \mathbb{T} = \left\langle \overline{L} \vee C_2' \right\rangle_{\mathcal{I}} \quad \text{cases} \quad \left\{ \begin{array}{ll} \langle L \rangle_{\mathcal{I}} = \mathbb{T} : \left(\text{otherwise } \left\langle \overline{L} \vee C_2' \right\rangle_{\mathcal{I}} = \mathbb{F}\right) & \left\langle C_2' \right\rangle_{\mathcal{I}} = \mathbb{T} \\ \left\langle \overline{L} \right\rangle_{\mathcal{I}} = \mathbb{T} : \left(\text{otherwise } \left\langle L \vee C_1' \right\rangle_{\mathcal{I}} = \mathbb{F}\right) & \left\langle C_1' \right\rangle_{\mathcal{I}} = \mathbb{T}, \text{ thus } \left\langle C_1' \vee C_2' \right\rangle_{\mathcal{I}} = \mathbb{T} \end{array} \right.$$

$$\begin{array}{lll} C_1 \wedge C_2 & \not\equiv & C_1' \vee C_2' \text{ !} \\ C_1 \wedge C_2 & \equiv & C_1 \wedge C_2 \wedge (C_1' \vee C_2') \end{array}$$

"resolution principle" $\stackrel{\wedge}{=}$ inference rule

"resolution method" $\stackrel{\wedge}{=}$ applying resolution principly until you reach the empty clause

$$\left.\begin{array}{l} A \vee B \\ \overline{A} \vee \overline{B} \end{array}\right\} \quad \mapsto \quad \underbrace{\overline{B} \vee B}_{\mathbb{T}}$$

4.2.5 Prove completeness

Completeness: every time it is unsatisfiable, will I get the empty clause ?

4.2.5.1 Semantic Tree

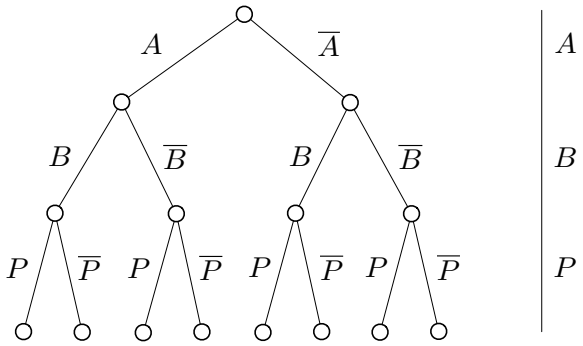


Figure 4.2: Semantic tree

$$\{A, \overline{B}, \overline{P}\} \quad : \quad \begin{cases} A \leftarrow \mathbb{T} \\ B \leftarrow \mathbb{F} \\ P \leftarrow \mathbb{F} \end{cases}$$

Clauses (compare tree in figure 4.1 on page 38):

1. $\overline{A} \vee P$
2. $\overline{B} \vee P$
3. $A \vee B$
4. $\overline{P}$
5. $\overline{A}$
6. $\overline{B}$
7. A
8. $\bigvee\{\}$

Using these clauses, one can close the nodes, which do not satisfy the clauses. In the following figure, closed nodes are shown crossed out and annoted the corresponding clause:

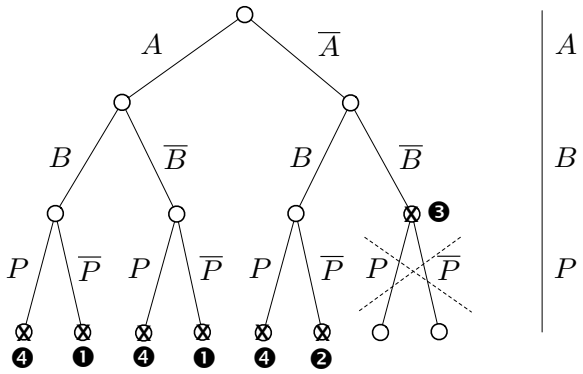


Figure 4.3: Semantic tree with closed nodes (1)

The closed nodes form a border within the tree, so that the areas below the closed nodes are no more reachable:

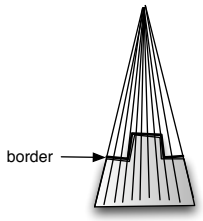


Figure 4.4: Border within the semantic tree caused by closed nodes

There must be at least one node, where both brothers are closed:

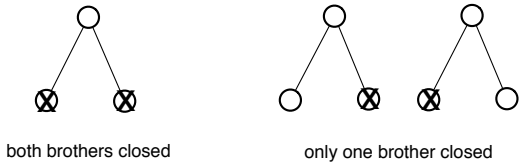


Figure 4.5: Both brothers closed versus one brother closed

So when both brothers are closed, then

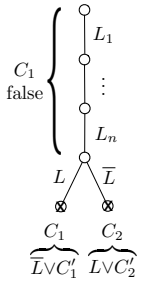


Figure 4.6: Consequence if both brothers are closed

We see that this principle applies, when we iterate over the clauses again (starting from our tree of figure 4.3 on the previous page):

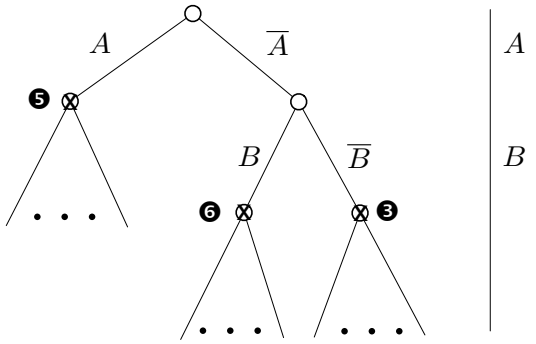


Figure 4.7: Semantic tree with closed nodes (2)

And finally

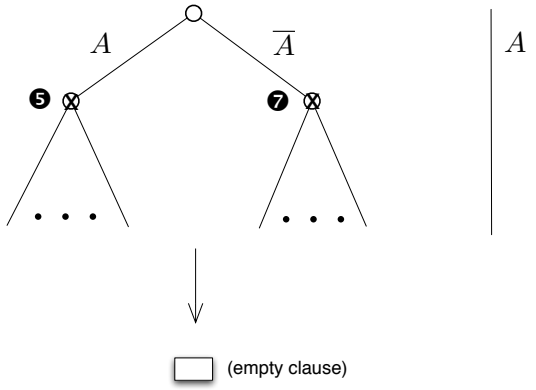


Figure 4.8: Semantic tree with closed nodes (3 and 4)

4.2.5.2 Replacement for tree

Instead of the tree, we will use the following steps:

1. $P \vee Q \vee R \ \rightarrow Q \vee R$
2. $\overline{P} \vee R \quad \nearrow$
3. $\overline{Q}$
4. $\overline{R}$
5. (1,2) $Q \vee R$
6. (1,3) $P \vee R$
7. (1,4) $P \vee Q$
8. (2,4) $\overline{P}$
9. (5,3) R
10. (5,4) Q
11. (7,3) P
12. (9,4) $\square$

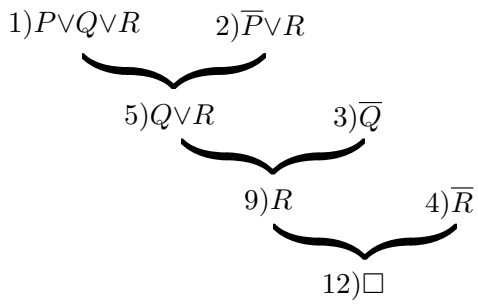


Figure 4.9: Tree for replacement

$$\text{"unit propagation"}: \quad \left.\begin{matrix} \vee \overline{P} \end{matrix}\right\} \quad \begin{matrix} \vee \overline{A} \\ \overline{B} \\ A \vee B \end{matrix} \quad \left\{\begin{matrix} \overline{B} \\ B \end{matrix}\right. \quad \rightarrow \quad \square$$

$$\begin{matrix} L: & C \vee \overline{L} & \mapsto & C \\ & C \vee L & \mapsto & \text{removed !} \end{matrix}$$

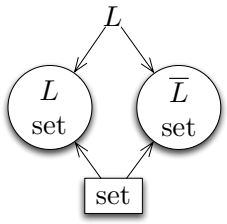


Figure 4.10: Davis Putnam

$$\text{initial set} \quad \begin{cases} P \vee Q \\ \overline{P} \vee Q \\ P \vee \overline{Q} \\ \overline{P} \vee \overline{Q} \end{cases} \quad \begin{matrix} P \\ \nearrow \\ \searrow \\ \overline{P} \end{matrix} \quad \begin{cases} Q \\ \overline{Q} \end{cases} \quad \to \square \qquad \begin{cases} Q \\ \overline{Q} \end{cases} \quad \to \square$$

$$\begin{cases} P \vee Q \\ \overline{P} \vee Q \\ R \end{cases} \quad \begin{matrix} \nearrow \\ \\ \searrow \end{matrix} \quad \begin{matrix} P : Q & \nearrow & \emptyset \\ & \searrow & \emptyset \\ \overline{P} : Q & \nearrow & \emptyset \\ & \searrow & \emptyset \end{matrix}$$

$$\begin{array}{ccc} & R \leftarrow \mathbb{T} & \\ \swarrow & & \searrow \\ \mathbb{T} \to P & & \overline{P} \leftarrow \mathbb{T} \\ \downarrow & & \downarrow \\ \mathbb{T} \to Q & & Q \leftarrow \mathbb{T} \end{array}$$

$(P \wedge Q \wedge R) \vee (\overline{P} \wedge Q \wedge R)$ ... Davis-Putnam method also generates a NF

Note: Homework 4.1 was: Can we see the "short calculus" $(\neg, \wedge, \vee)$ as a normal form algorithm for

- sequents (what would be a good normal form ?) (could be a formula, or conjunction of sequents
- propositional formulae ? (DNF)

4.3 First-order Predicate Logic

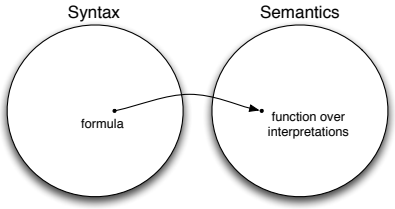


Figure 4.11: Syntax and Semantics

Previously, we had the formula: $(\varphi_1, \ldots, \varphi_n \models \psi)$ iff $(\varphi_1 \wedge \cdots \wedge \varphi_n \Rightarrow \psi)$ is valid

One could rewrite this to be more precise:

$$\forall_{\varphi_1,\ldots,\varphi_n,\psi:Formulae} isSemanticalLogicalConsequence\,(SetOf\,(\varphi_1,\ldots,\varphi_n),\psi) \Leftrightarrow \\ \Leftrightarrow isValid\,(Quote\,(Implication\,(And\,(\varphi_1,\ldots,\varphi_n),\psi)))$$

Or even further:

$$\forall_{\varphi_1,\ldots,\varphi_n,\psi}\,(IsFormula\,(\varphi_1) \wedge \cdots \wedge IsFormula\,(\varphi_n) \wedge IsFormula\,(\psi)) \Rightarrow \\ (isSemanticalLogicalConsequence\,(SetOf\,(\varphi_1,\ldots,\varphi_n),\psi) \Leftrightarrow \\ \Leftrightarrow isValid\,(Quote\,(Implication\,(And\,(\varphi_1,\ldots,\varphi_n),\psi))))$$

Note: $\forall_{\varphi_1,\ldots,\varphi_n,\psi}$ is a "shortcut" for $\forall_{\varphi_1}\forall_{\varphi_2}\ldots\forall_{\varphi_n}\forall_{\psi}$

Formulae:

- terms
  - variables, constants
  - function symbols
- quantifiers
- logical connectives
- predicates

4.3.1 Definition of syntax

$$\text{2 Languages} \quad \begin{cases} \text{Language of terms } \mathscr{L}_T \\ \text{Language of formulae } \mathscr{L}_F \end{cases}$$

4.3.1.1 Language of terms

$$\mathscr{L}_T \quad \begin{cases} \vartheta \in \Theta, c \in \mathscr{C} : \text{ are terms} \\ f \in \mathscr{F}, \quad t_1, \ldots, t_n : \text{ terms, then} \\ \qquad\qquad f(t_1, \ldots, t_n) \text{ is term} \\ \text{(these are all !)} \end{cases}$$

variables constants
$\downarrow$ $\downarrow$

- Θ , $\mathscr{C}$ are (infinite) sets of symbols
- $\mathscr{F}$: set of "functional symbols"
  $\mathscr{F} = \bigcup_{n\in\mathbb{N}} \mathscr{F}_n$
  - each $\mathscr{F}_n$ is infinite
  - $\mathscr{F}_n \cap \mathscr{F}_m = \emptyset$ (disjoint)
  - $f \in \mathscr{F}_n$: "f has arity n"
    - if $n = 0$, we don't write $f()$, but f, which is a constant
    - so $\mathscr{C} = \mathscr{F}_0$

4.3.1.2 Language of formulae

$$\mathscr{L}_F \quad \begin{cases} p \in \mathscr{P}, \quad t_1, \ldots, t_m : \text{ terms, then} \\ \qquad\qquad P(t_1, \ldots, t_m) \text{ is formula ("atom")} \\ \varphi, \psi \text{ formulae}, \vartheta \in \Theta : \quad \forall_\vartheta \varphi, \exists_\vartheta \varphi, \\ \qquad\qquad\qquad\qquad \neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \\ \qquad\qquad\qquad\qquad \varphi \Rightarrow \psi, \varphi \Leftrightarrow \psi \text{ are formulae} \\ \text{(these are all !)} \end{cases}$$

- $\mathscr{P}$ "predicate symbols"
  - $\mathscr{P} = \bigcup_{m\in\mathbb{N}} \mathscr{P}_m$, each $\mathscr{P}_m$ infinite,
  - $\mathscr{P}_n \cap \mathscr{P}_m = \emptyset$ (disjoint)
  - $p \in \mathscr{P}_m$: "p has arity m"
    - if $m = 0$, we don't write $P()$, but P
    - $P \rightsquigarrow$ propositional variable, so propositional logic is a subset of first-order predicate logic

4.3.1.3 Example

For every two points, there is one and only one line through the two points

$$\forall_x \forall_y P(x) \wedge P(y) \quad \Rightarrow \quad \exists_z ! \, (L(z) \wedge T(z, x, y))$$

$P(x)$: "x is point"
$L(z)$: "z is line"
$T(z, x, y)$: "z passes through x and y"

Schema for "there exists exactly one":

$$\exists_x !P(x) \quad : \quad \exists_x P(x) \wedge \left(\forall_y P(y) \Rightarrow \underbrace{(x = y)}_{\text{more precise: } Equal(x,y)} \right)$$

Written without $\exists!$-Usage, and demonstrating terms "scope" and "bound":

$$\forall_x \forall_y \left(P(x) \wedge P(y) \Rightarrow \exists_z \left(\underbrace{(L(z) \wedge T(z,x,y)) \wedge \forall_t \left(\underbrace{(L(t) \wedge T(t,x,y)) \Rightarrow (t = z)}_{\text{scope of the } \forall_t \text{ quantifier}} \right)}_{\substack{\text{scope of the } \exists \text{ quantifier,} \\ \text{variable } z \text{ is bound in this scope}}} \right) \right)$$

$$\begin{array}{lccl} \exists_x & x & \leq & y \\ & \uparrow & & \uparrow \\ & \text{bound} & & \text{free} \\ & & & \uparrow \\ & & & \text{which } y \text{ ?} \\ & & & \uparrow \\ & & & \text{no meaning is assigned to } y \end{array}$$

Only list has a variable arity

- $\langle 1, a, f(2), 3, \diamond \rangle$
- $\langle x, 5 \rangle$
- $List\,(\langle 1, a, f(2), 3, \diamond \rangle)$
- $List\,(\langle x, 5 \rangle)$
- $\forall_{x,y} \forall_{x',y'} \langle x, y \rangle = \langle x', y' \rangle \Rightarrow x = x', y = y'$

In First-order predicate logic, you cannot have this flexible arity

4.3.2 Semantics

$$f_{\forall_x \exists_y x \leq y} \quad : \quad \underbrace{\mathscr{I}}_{\text{interpretations}} \rightarrow \{\mathbb{T}, \mathbb{F}\}$$

$$\mathcal{I} \quad : \quad \begin{cases} \text{"domain"} \ldots & D \neq \emptyset \\ \text{constant symbol} \ldots & c_{\mathcal{I}} \in D \\ \text{functional symbol} \ldots \text{ (arity } n\text{)} & f_{\mathcal{I}} : D^n \rightarrow D \\ \text{predicate symbol} \ldots \text{ (arity } m\text{)} & p_{\mathcal{I}} : D^m \rightarrow \{\mathbb{T}, \mathbb{F}\} \end{cases}$$

Example:

$$\forall_x \exists_y x \leq y$$

$$\mathcal{I}: \begin{cases} D = \{0,1\} \\ \leq_{\mathcal{I}}: \end{cases}$$

x\y	0	1
0	$\mathbb{T}$	$\mathbb{T}$
1	$\mathbb{F}$	$\mathbb{T}$

$\left\langle \forall_x \exists_y x \leq y \right\rangle_{\mathcal{I}} = \mathbb{T}$ iff foreach $d \in D$:

$\left\langle \exists_y x \leq y \right\rangle^{\mathcal{I}}_{\{x \leftarrow 0\}}$ ~~$\left\langle \exists_y 0 \leq y \right\rangle_{\mathcal{I}}$~~ $\left\langle \exists_y x \leq y \right\rangle^{\mathcal{I}}_{\{x \leftarrow d\}}$

$\langle x \leq y \rangle^{\mathcal{I}}_{\{x \leftarrow 0, y \leftarrow 0\}}$ ~~$\left\langle \exists_y 1 \leq y \right\rangle_{\mathcal{I}}$~~

$$\leq_{\mathcal{I}} \left(\langle x \rangle^{\mathcal{I}}_{\{x \leftarrow 0, y \leftarrow 0\}}, \langle y \rangle^{\mathcal{I}}_{\{x \leftarrow 0, y \leftarrow 0\}} \right)$$
$$\leq_{\mathcal{I}} (0,0) = \mathbb{T}$$

$$\left\langle \exists_y x \leq y \right\rangle^{\mathcal{I}}_{\{x \leftarrow 1\}} = \cdots = \mathbb{T}$$

Chapter 5

Fifth Lecture on 10.11.2005

5.1 Example: predicate logic formula, interpretation and truth evaluation

$$\forall_x \left(P\left(x\right) \Rightarrow Q\left(f\left(x\right), a\right)\right)$$

f.ex. $\mathcal{I}$:

$D = \{1, 2\}$

$a_{\mathcal{I}} = 1 \quad (\in D)$

$f_{\mathcal{I}} : D \to D \quad \begin{cases} f_{\mathcal{I}}\left(1\right) = 1 \\ f_{\mathcal{I}}\left(2\right) = 1 \end{cases}$

$P_{\mathcal{I}} : D \to \{\mathbb{T}, \mathbb{F}\} \quad \begin{cases} P_{\mathcal{I}}\left(1\right) = \mathbb{T} \\ P_{\mathcal{I}}\left(2\right) = \mathbb{F} \end{cases}$

$Q_{\mathcal{I}} : D^2 \to \{\mathbb{T}, \mathbb{F}\}$

$Q_{\mathcal{I}}$	1	2
1	$\mathbb{T}$	$\mathbb{F}$
2	$\mathbb{F}$	$\mathbb{T}$

$$\begin{array}{rcll} \left\langle \forall_x \left(P\left(x\right) \Rightarrow Q\left(f\left(x\right), a\right)\right)\right\rangle_{\mathcal{I}} & = & \mathbb{T} & \\ & \text{iff} & & \\ \left\langle P\left(x\right) \Rightarrow Q\left(f\left(x\right), a\right)\right\rangle^{\mathcal{I}}_{\{x \leftarrow d\}} & = & \mathbb{T} & \text{(for each } d \in D\text{)} \end{array}$$

So for each element of the domain there is a case:

- Case $d = 1$:

$$\begin{array}{rcl} \left\langle P\left(x\right) \Rightarrow Q\left(f\left(x\right), a\right)\right\rangle^{\mathcal{I}}_{\{x \leftarrow 1\}} & = & \mathscr{B}_{\Rightarrow}\left(\left\langle P\left(x\right)\right\rangle^{\mathcal{I}}_{\{x \leftarrow 1\}}, \left\langle Q\left(f\left(x\right), a\right)\right\rangle^{\mathcal{I}}_{\{x \leftarrow 1\}}\right) \\ & = & \mathscr{B}_{\Rightarrow}\left(P_{\mathcal{I}}\left(\left\langle x\right\rangle^{\mathcal{I}}_{\{x \leftarrow 1\}}\right), Q_{\mathcal{I}}\left(\left\langle f\left(x\right)\right\rangle^{\mathcal{I}}_{\{x \leftarrow 1\}}, \left\langle a\right\rangle^{\mathcal{I}}_{\{x \leftarrow 1\}}\right)\right) \\ & = & \mathscr{B}_{\Rightarrow}\left(P_{\mathcal{I}}\left(\left\langle x\right\rangle^{\mathcal{I}}_{\{x \leftarrow 1\}}\right), Q_{\mathcal{I}}\left(f_{\mathcal{I}}\left(\left\langle x\right\rangle^{\mathcal{I}}_{\{x \leftarrow 1\}}\right), \left\langle a\right\rangle^{\mathcal{I}}_{\{x \leftarrow 1\}}\right)\right) \\ & = & \mathscr{B}_{\Rightarrow}\left(P_{\mathcal{I}}\left(1\right), Q_{\mathcal{I}}\left(f_{\mathcal{I}}\left(\left\langle x\right\rangle^{\mathcal{I}}_{\{x \leftarrow 1\}}\right), a_{\mathcal{I}}\right)\right) \\ & = & \mathscr{B}_{\Rightarrow}\left(\mathbb{T}, Q_{\mathcal{I}}\left(f_{\mathcal{I}}\left(1\right), 1\right)\right) = \mathscr{B}_{\Rightarrow}\left(\mathbb{T}, Q_{\mathcal{I}}\left(1, 1\right)\right) \\ & = & \mathscr{B}_{\Rightarrow}\left(\mathbb{T}, \mathbb{T}\right) = \mathbb{T} \end{array}$$

- Case $d = 2$:

$$\langle P(x) \Rightarrow Q(f(x), a)\rangle^{\mathcal{I}}_{\{x \leftarrow 2\}} \quad = \quad \ldots = \mathbb{T}$$

Problem: infinite Domains ! this requires more clever techniques (because no explicit definition of the interpretation is possible)

Also note that writing $P(1) \Rightarrow Q(f(1), a)$ is not correct, because the elements of D are not in our alphabet.

So one uses the trick, that one adds the $d \in D$ (of the respective interpretation) to the alphabet on-the-fly. This must be done for each respective interpretation.

5.2 Different domains ("types")

isString	Len	isString["a string"] = $\mathbb{T}$
isReal	Abs	isString[14] = $\mathbb{F}$

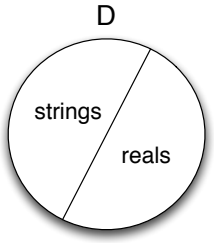


Figure 5.1: Classes of the domain strings and reals

$$\begin{aligned} \forall_s isString(s) &\Rightarrow Len(S) = 1 + Len(Tail(S)) \\ s &\neq EmptyString \\ \forall_r isReal(r) &\Rightarrow ((r \geq 0 \Rightarrow Abs(r) = r) \wedge (r < 0 \Rightarrow Abs(r) = -r)) \end{aligned}$$

Abs("Something") $\rightarrow$ is no "mistake", only cannot evaluate

like in Mathematics, $\frac{1}{0}$ also is no syntactical "mistake", only cannot evaluate

In C-like syntax:

```
Real Abs(Real r) {
    if (r≥0) then r
    else (-r)
}
```

5.3 Truth evaluation $\langle\varphi\rangle_{\mathcal{I}}$

5.3.1 Formula

$\mathbb{T}, \mathbb{F}$	
$P(t_1, \ldots, t_n)$	$\langle P(t_1, \ldots, t_n)\rangle_\alpha^{\mathcal{I}} = P_{\mathcal{I}}\left(\langle t_1\rangle_\alpha^{\mathcal{I}}, \ldots, \langle t_n\rangle_\alpha^{\mathcal{I}}\right)$
$\neg\varphi$, $\quad \varphi\Omega\psi$ $\quad\quad\uparrow$ $\in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$	$\langle\varphi\Omega\psi\rangle_\alpha^{\mathcal{I}} = \mathscr{B}_\Omega\left(\langle\varphi\rangle_\alpha^{\mathcal{I}}, \langle\psi\rangle_\alpha^{\mathcal{I}}\right)$ $\langle\neg\varphi\rangle_\alpha^{\mathcal{I}} = \mathscr{B}_\neg\left(\langle\varphi\rangle_\alpha^{\mathcal{I}}\right)$
$\forall_\vartheta\varphi$ $\quad(\vartheta:$ variable symbol)	$\langle\forall_\vartheta\varphi\rangle_\alpha^{\mathcal{I}} = \mathbb{T}$ iff (for each $d \in D, \langle\varphi\rangle_{\alpha\cup\{\vartheta\leftarrow d\}}^{\mathcal{I}} = \mathbb{T}$) (of the interpretation $\mathcal{I}$)
$\exists_\vartheta\varphi$ $\quad(\vartheta:$ variable symbol)	$\langle\exists_\vartheta\varphi\rangle_\alpha^{\mathcal{I}} = \mathbb{T}$ iff (for some $d \in D, \langle\varphi\rangle_{\alpha\cup\{\vartheta\leftarrow d\}}^{\mathcal{I}} = \mathbb{T}$) (of the interpretation $\mathcal{I}$)

$\forall_x P(x) \Rightarrow \exists_y Q(x, y)$
$\langle\ldots\rangle_{\{\}}^{\mathcal{I}} \ \ldots\ \langle\ldots\rangle_{\{x\leftarrow d\}}^{\mathcal{I}} \ \ldots\ \langle\ldots\rangle_{\{x\leftarrow d, y\leftarrow d\}}^{\mathcal{I}}$

But what if x is already bound:

$\forall_x P(x) \Rightarrow \exists_x Q(x, x)$
$\langle\ldots\rangle_{\{\}}^{\mathcal{I}} \ \ldots\ \langle\ldots\rangle_{\{x\leftarrow d\}}^{\mathcal{I}} \ \ldots\ \langle\ldots\rangle^{\mathcal{I}}_{\{x\leftarrow d, x\leftarrow d\}}$
$\uparrow$
which x ?

"$\check{\bigcup}$" : modify the assignment !
(replace any other $\{\vartheta \leftarrow \ldots\}$!

All variables must be bound.

All free variables are present in the assignment α

5.3.2 Term

ϑ $\quad(\in$ variable symbol set)	$\langle\vartheta\rangle_\alpha^{\mathcal{I}} = \langle\vartheta\rangle_{\{\ldots,\vartheta\leftarrow d,\ldots\}}^{\mathcal{I}} = d$ (assume that there is such an assignment $\vartheta \leftarrow d$)
c $\quad(\in$ constant symbol set)	$\langle c\rangle_\alpha^{\mathcal{I}} = c_{\mathcal{I}}$
f $\quad(\in$ functional symbol set)	$\langle f(t_1, \ldots, t_n)\rangle_\alpha^{\mathcal{I}} = f_{\mathcal{I}}\left(\langle t_1\rangle_\alpha^{\mathcal{I}}, \ldots, \langle t_n\rangle_\alpha^{\mathcal{I}}\right)$

5.3.3 Equivalence

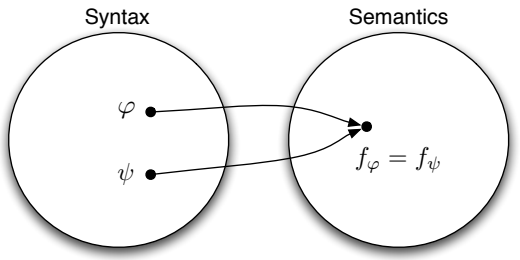


Figure 5.2: Syntax, Semantics and semantical equivalence

Then we can define equivalence of two formulae φ and ψ as that they've got the same semantical function.
$\Phi \models \Psi$ semantical logical consequence

$$\begin{array}{lrcl} \diamond: & \neg(\forall_x \varphi) & \equiv & \exists_x(\neg\varphi) \\ & \neg(\exists_x \varphi) & \equiv & \forall_x(\neg\varphi) \end{array}$$

$$\begin{array}{lrcl} \clubsuit: & \forall_x(\varphi \wedge \psi) & \equiv & (\forall_x\varphi) \wedge (\forall_x\psi) \\ & \exists_x(\varphi \vee \psi) & \equiv & (\exists_x\varphi) \vee (\exists_x\psi) \end{array}$$

$$\left.\begin{array}{l} \forall_x(\varphi \vee \psi) \equiv (\forall_x\varphi) \vee \psi \\ \exists_x(\varphi \wedge \psi) \equiv (\exists_x\varphi) \wedge \psi \end{array}\right\} \begin{array}{c} \text{if } x \mathring{\notin} \psi \\ \text{"does not occur"} \end{array}$$

Substitution:

$$\begin{array}{rcl} \forall_x\varphi & \equiv & \forall_y\varphi_{x\leftarrow y} \\ \multicolumn{3}{c}{\text{if } y \mathring{\notin} \varphi} \\ \exists_x\varphi & \equiv & \exists_y\varphi_{x\leftarrow y} \end{array}$$

We try to prove $\diamond$:

$$\begin{array}{rcll} \text{For arbitary interpretation } \mathcal{I}: & & & \\ \langle\neg\forall_x\varphi\rangle^{\mathcal{I}} = \langle\neg\forall_x\varphi\rangle^{\mathcal{I}}_{\{\}} & = & \mathscr{B}_\neg\left(\langle\forall_x\varphi\rangle^{\mathcal{I}}_{\{\}}\right) & \\ & = \mathbb{F} \quad \text{iff} & \left(\langle\forall_x\varphi\rangle^{\mathcal{I}}_{\{\}} = \mathbb{T}\right) \text{ iff} & \\ & & \text{for each } d \in D: \langle\varphi\rangle^{\mathcal{I}}_{\{x\leftarrow d\}} = \mathbb{T} & \\ & = \mathbb{T} \quad \text{iff} & \ldots & \end{array}$$

$$\left(\langle\neg\exists_x\varphi\rangle^{\mathcal{I}}_{\{\}} = \mathbb{F}\right) \quad \text{iff} \quad \text{for some } d \in D: \langle\varphi\rangle^{\mathcal{I}}_{\{x\leftarrow d\}} = \mathbb{T}$$

Note: Homework 5.1 will be to prove $\clubsuit$, Homework 5.2 will be to find an example where this equivalence does not hold: $\exists_x\varphi \wedge \psi \neq \exists_x\varphi \wedge \exists_x\psi$.

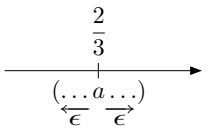
5.4 Example to see how people prove things

The situation:

$$\text{A sequence } f \text{ is convergent: } \underbrace{\exists_{a\in\mathbb{R}}\forall_{\substack{\epsilon\in\mathbb{R}\\ \epsilon>0}}\exists_{n\in\mathbb{N}}\forall_{\substack{p\in\mathbb{N}\\ p>n}} |f(p)-a|<\epsilon}_{\varphi}$$

$f : \mathbb{N} \to \mathbb{R}$
$f(n) = \frac{2n+3}{3n+1}$

n	$f(n)$
0	3/1
1	5/4
2	7/7
3	9/10
⋮	⋮



"in our syntax":

$$\exists_a (a \in \mathbb{R}) \wedge \forall_\epsilon (\epsilon \in \mathbb{R} \wedge \epsilon > 0) \Rightarrow \exists_n (n \in \mathbb{N}) \wedge \forall_p (p \in \mathbb{N} \wedge p > n) \Rightarrow |f(p) - a| < \epsilon$$

Note that $|f(p) - a| < \epsilon$ could also be formulated as $Less(Abs(Minus(f(p), a)), \epsilon)$

$$\forall_f isSequence(f) \Rightarrow (isConvergent(f) \Leftrightarrow \varphi)$$

Note that *isConvergent* is a predicate over a function, so this is "second order" predicate logic, where one can have predicates and functions applied to first order formulae. The general term is "higher order" predicate logic.

If f, g convergent, then $f \oplus g$ is also convergent:

$$\forall_{f,g}\forall_{n\in\mathbb{N}} \quad \underbrace{(f \oplus g)(n)}_{Plus(f,g)(n)\text{ "currying"}} = f(n) + g(n)$$

} higher order

$$\forall_{\substack{f,g\\ IsSequence(f)\wedge\\ IsSequence(g)}} (IsConvergent(f) \wedge IsConvergent(g)) \Rightarrow IsConvergent(f \oplus g)$$

Note: f, g are arbitary but fixed here

$$f,g: \quad IsSequence(f), \quad IsSequence(g)$$

$$\exists_{a\in\mathbb{R}}\forall_{\substack{\epsilon\in\mathbb{R}\\ \epsilon>0}}\exists_{n\in\mathbb{N}}\forall_{\substack{p\in\mathbb{N}\\ p>n}} |f(p)-a|<\epsilon$$

Assume: $\begin{cases} IsConvergent(f)\ \nearrow \\ IsConvergent(g)\ \searrow \end{cases}$

$$\exists_{a\in\mathbb{R}}\forall_{\substack{\epsilon\in\mathbb{R}\\ \epsilon>0}}\exists_{n\in\mathbb{N}}\forall_{\substack{p\in\mathbb{N}\\ p>n}} |g(p)-a|<\epsilon$$

Prove: $IsConvergent(f\oplus g)$

$$\exists_{a\in\mathbb{R}}\forall_{\substack{\epsilon\in\mathbb{R}\\ \epsilon>0}}\exists_{n\in\mathbb{N}}\forall_{\substack{p\in\mathbb{N}\\ p>n}} \left| \underbrace{(f\oplus g)(p)}_{\substack{\text{replace with:}\\ f(p)+g(p)}} - a \right| < \epsilon$$

Used for proof $\begin{cases} \forall_x IsSequence(f) \Rightarrow (IsConvergent(f) \Leftrightarrow \varphi) \\ \xrightarrow[\text{instantiation}]{} \quad IsSequence(g) \Rightarrow \underbrace{\left(isConvergent(g) \Leftrightarrow \varphi_{f\leftarrow g}\right)}_{\forall_x\varphi \rightsquigarrow \varphi_{x\leftarrow t}} \\ \frac{\cdots\vdash\varphi}{\varphi\Leftrightarrow\psi\vdash\Psi} \end{cases}$

A_f : "take" $a_1\in\mathbb{R}$:	$\forall_{\substack{\epsilon\in\mathbb{R}\\ \epsilon>0}} \ldots f\cdots < \frac{\epsilon_0}{2}$
A_g : "take" $a_2\in\mathbb{R}$:	$\forall_{\substack{\epsilon\in\mathbb{R}\\ \epsilon>0}} \ldots g\cdots < \frac{\epsilon_0}{2}$
G : "use" $a\leftarrow a_1+a_2$	$\forall_{\substack{\epsilon\in\mathbb{R}\\ \epsilon>0}} \exists_{n\in\mathbb{N}} \lvert(f(p)+g(p))-(a_1+a_2)\rvert<\epsilon_0$ $\epsilon_0\in\mathbb{R}$ $a_1+a_2\in\mathbb{R}$ $\quad \epsilon_0>0$ assumptions

inference rules (assumptions)	inference rules (goals)
$\forall_x\varphi \vdash\rightsquigarrow \varphi_{x\leftarrow t} \vdash$	$\vdash \forall_x\varphi \rightsquigarrow\vdash \varphi_{x\leftarrow a}$ (a "is new") $(a\mathring{\notin}\varphi)$
$\exists_x\varphi \vdash\rightsquigarrow \varphi_{x\leftarrow a} \vdash$ (a "is new") $(a\mathring{\notin}\varphi)$	$\vdash \exists_x\varphi \rightsquigarrow\vdash \varphi_{x\leftarrow t}$

Note: Homework 5.3 was to finish this proof

5.5 Sequents

inference rules (assumptions)	inference rules (goals)
$\dfrac{\Phi,\varphi_{x\leftarrow a}\vdash\Psi}{\Phi,\exists_x\varphi\vdash\Psi}$ $(\exists\vdash)$ where $a\mathring{\notin}\varphi,\Phi,\Psi$	$\dfrac{\Phi\vdash\Psi,\psi_{x\leftarrow t}}{\Phi\vdash\Psi,\exists_x\psi}$ $(\vdash\exists)$
$\dfrac{\Phi,\varphi_{x\leftarrow t}\vdash\Psi}{\Phi,\forall_x\varphi\vdash\Psi}$ $(\forall\vdash)$	$\dfrac{\Phi\vdash\Psi,\psi_{x\leftarrow a}}{\Phi\vdash\Psi,\forall_x\psi}$ $(\vdash\forall)$ where $a\mathring{\notin}\psi,\Phi,\Psi$

Note: first we will study the proof system (resolution) and we will infer as a consequence the proof for correctness and completeness

5.6 Prenex

A "prenex formula" is of the form:

$$\underbrace{\left(\forall_x \exists_y \exists_z \forall_t \dots\right)}_{\text{all quantifiers}} \quad \underbrace{\varphi}_{\text{quantifier-free formula}}$$

The quantifier-free formula part can be transformed into CNF

$$\left.\begin{array}{r}\text{"prenex formula"}\\ \text{(with the quantifier free formula part in) CNF}\end{array}\right\} \quad \text{"prenex normal form"}$$

So, a formula in *prenex normal form* somehow looks like:

$$\left(\forall_x \exists_y \exists_z \forall_t \dots\right)((\cdots \vee \cdots \vee \dots) \wedge \cdots \wedge (\cdots \vee \cdots \vee \dots))$$

5.7 Skolem transformation

A "skolem transformation" is
$$\begin{array}{ccc} \exists_x P(x) & \rightsquigarrow & P(a) \\ \text{if this is sat} \rightarrow & \text{then} & \rightarrow \text{this is sat} \\ \text{this is sat} \leftarrow & \text{then} & \leftarrow \text{if this is sat} \end{array}$$

Assume that we have an interpretation $\mathcal{I}$ that satisfies $\exists_x P(x)$

$$\mathcal{I}: \left\{\begin{array}{l} D \\ P_{\mathcal{I}} : D \to \{\mathbb{T}, \mathbb{F}\} \\ \text{for some } d \in D : P_{\mathcal{I}}(d) = \mathbb{T} \end{array}\right. \quad \mapsto \quad \mathcal{I}': \left\{\begin{array}{ll} D & \\ P_{\mathcal{I}'} = P_{\mathcal{I}} & \text{so } P_{\mathcal{I}'}(a_{\mathcal{I}}) = \mathbb{T} \\ a_{\mathcal{I}'} = d & \end{array}\right.$$

Note: Homework 5.4 will be to prove the other direction $\underbrace{P(a)}_{\text{assume}} \to \underbrace{\exists_x P(x)}_{\text{prove}}$

$$\begin{array}{rcl} (\exists_x \varphi \text{ satisfiable}) & \text{iff} & (\varphi_{x \leftarrow a} \text{ satisfiable}) \\ \text{and more interesting:} & & \\ (\exists_x \varphi \text{ unsatisfiable}) & \text{iff} & (\varphi_{x \leftarrow a} \text{ unsatisfiable}) \end{array}$$

$$\begin{array}{ccc} \underline{\forall_x \exists_y P(x,y)} & \leftarrow\!\shortmid & \underline{\forall_x P(x, f(x))} \\ \mathcal{I}' \dots (D, P_{\mathcal{I}}) & & \mathcal{I} \dots (D, P_{\mathcal{I}}, f_{\mathcal{I}}) \\ \text{for each } d \in D & & \text{for each } d \in D \\ \quad \text{(there is } d' \in D, & & \quad P_{\mathcal{I}}(d, f_{\mathcal{I}}(d)) = \mathbb{T} \\ \quad P_{\mathcal{I}}(d,d') = \mathbb{T}) & & \\ \quad \text{take } d' = f_{\mathcal{I}}(d) & & \end{array}$$

Note: Homework 5.5 was to show the other direction $\mapsto$. Hint: Take an interpretation which is sat on the LHS and show that it is sat RHS.

$$\Big(\forall_x \exists_y \varphi \text{ unsatisfiable}\Big) \quad \text{iff} \quad \Big(\forall_x \varphi_{y \leftarrow f(x)} \text{ unsatisfiable}\Big) \text{ where } f \overset{\circ}{\notin} \varphi$$

and more generally:

$$\Big(\forall_{x_1} \ldots \forall_{x_n} \exists_y \varphi\Big) \quad \rightsquigarrow \quad \Big(\forall_{x_1} \ldots \forall_{x_n} \varphi_{x \leftarrow f(x_1,\ldots,x_n)}\Big)$$

So, by *skolem transformation*, one can eliminate all the $\exists$-quantors.

$$\Big(\forall_x \exists_y \forall_z \forall_t \exists_w\Big) \quad \underbrace{((\cdots \vee \cdots \vee \ldots) \wedge \cdots \wedge (\cdots \vee \cdots \vee \ldots))}_{CNF}$$

$$\downarrow \text{"skolem transform"}$$

$$(\forall_{x_1} \ldots \forall_{x_n}) \quad \underbrace{((\cdots \vee \cdots \vee \ldots) \wedge \cdots \wedge (\cdots \vee \cdots \vee \ldots))}_{CNF}$$

One does not need to write the quantifiers $(\forall_{x_1} \ldots \forall_{x_n})$ any more ! You simply assume that all are universally quantified.

5.8 Resolution

"Resolution principle": $\left.\begin{array}{c} L \vee C_1 \\ \overline{L} \vee C_2 \end{array}\right\} \quad C_1 \vee C_2$

For example:

$$\left.\begin{array}{c} \forall_x P(x) \Rightarrow Q(x) \\ P(a) \end{array}\right\} \quad Q(a) \qquad\qquad \left.\begin{array}{c} \overline{P}(x) \vee Q(a) \\ P(a) \end{array}\right\} \quad Q(a)$$

If $(L_1\sigma = L_2\sigma)$, $\left.\begin{array}{c} L_1 \vee C_1 \\ L_2 \vee C_2 \end{array}\right\} \quad C_1\sigma \vee C_2\sigma$

For example:

$$\left.\begin{array}{lcl} \overline{P}(x,a) \vee Q(x) & \overset{\text{instantiate}}{\underset{\{x \leftarrow b\}}{\longrightarrow}} & \overline{P}(b,a) \vee Q(b) \\ P(b,y) \vee R(y) & \overset{}{\underset{\{y \leftarrow a\}}{\longrightarrow}} & P(b,a) \vee R(a) \end{array}\right\} \quad Q(b) \vee R(a)$$

Substitution $\sigma = \{x \leftarrow b, y \leftarrow a\}$

$$\nearrow \text{ Correctness: } \begin{array}{c} L_1 \vee C_1, \overline{L_2} \vee C_2 \models C_1\sigma \vee C_2\sigma \\ (\text{where } L_1\sigma = L_2\sigma) \end{array} \quad \left\{\begin{array}{l} \forall_\vartheta \varphi \models \quad \forall_{\vartheta_1} \ldots \forall_{\vartheta_n} \varphi_{\vartheta \leftarrow t} \\ \qquad \{\vartheta_1, \ldots, \vartheta_n\} = FreeVars(t) \\ \forall_{x_1} \ldots \forall_{x_n} \Big((L_1 \vee C_1) \wedge \Big(\overline{L_2} \vee C_2\Big)\Big) \Rightarrow (C_1 \vee C_2) \\ \updownarrow \\ \quad (\forall_{x_1} \ldots \forall_{x_n} L_1 \vee C_1) \\ \quad \Big(\forall_{x_1} \ldots \forall_{x_n} \overline{L_2} \vee C_2\Big) \end{array}\right. \models \quad (\forall_{x_1} \ldots \forall_{x_n} C_1 \vee C_2)$$

$\searrow$ Completeness: If φ unsatisfiable, then $\varphi \vdash_{Res} \square$ (the empty clause)

Roughly sketch the proof

$$\begin{array}{ll} d \in D & \langle \varphi \rangle^{\mathcal{I}}_{\{\vartheta \leftarrow d\}} = \mathbb{T} \\ & \uparrow \\ & \langle t \rangle^{\mathcal{I}}_{\{\vartheta_1 \leftarrow d_1, \ldots, \vartheta_n \leftarrow d_n\}} \end{array}$$

Chapter 6

Sixth Lecture on 24.11.2005

6.1 Recapitulation of the Lecture until now

- Propositional Logic
- Predicate Logic

For those, we have discussed:

- Syntax, Semantics
  - $\mathscr{L}, \mathcal{I}$
  - $\langle\varphi\rangle_{\mathcal{I}}$, $\langle\varphi\rangle^{\mathcal{I}}_{\{x\leftarrow d,\dots\}}$
- Proof systems
  - sequent calculus ("natural deduction") DNF
    - "short calculus"
      - $\neg, \vee, \wedge$: completeness
      - Elimination of inference rules by using other
    - $\forall, \exists$
  - resolution (clauses, normal form)
    - correctness, completeness
    - Now, one should know
      - how to transform into NF (skolem / prenex)
      - how to do proof with resolution

6.2 Questions and Answers

6.2.1 Q1: Difference Correctness / Completeness

With the sequent calculus:

- *Correctness:* if $\left(\begin{array}{c}\text{(obtain a proof tree)}\\ \downarrow \\ \text{syntactically}\\ \downarrow \\ \varphi \qquad \text{proved}\end{array}\right) \xrightarrow{\text{then}} (\varphi \text{ is valid})$
- *Completeness:* Whenever $(\varphi \text{ is valid}) \xleftarrow{\text{then}} (\text{you are able to prove } \varphi)$

And with resolution:

- *Correct:* Whenever you can obtain $\square$, you know that it is unsatisfiable
- *Complete:* Whenever it is unsatisfiable, you will obtain $\square$

Sequent calculus:

$\frac{S_1 S_2}{S}$ Correctness/Completeness: Because both directions ($\uparrow\downarrow$) work for each rule

6.2.2 Q2: Allquantors

inference rules (assumptions)	inference rules (goals)
$\frac{\Phi, \varphi_{x\leftarrow t} \vdash \Psi}{\Phi, \forall_x \varphi \vdash \Psi}$ $(\forall \vdash)$	$\frac{\Phi \vdash \Psi, \psi_{x\leftarrow a}}{\Phi \vdash \Psi, \forall_x \psi}$ $(\vdash \forall)$ where $a \mathring{\notin} \psi, \Phi, \Psi$

For example (see example with convergence):

$$\begin{array}{c}\text{replace } \epsilon \\ \text{with a new term} \\ \downarrow\end{array} \quad \begin{array}{lcll} \ldots, \forall_\epsilon \ldots \text{ if } \ldots| < \epsilon & \vdash & \forall_\epsilon \ldots \text{ if } f \oplus g \ldots| < \epsilon & \leftarrow \text{in order to prove it for all it is enough } \ldots \\ \text{if } \ldots| < \frac{\epsilon_0}{2} & \vdash & \ldots \text{ if } f \oplus g \ldots| < \epsilon_0 & \leftarrow \ldots \text{ to prove it for one constant which somehow looks strange} \end{array}$$

The clou of the proof is how define this term t.

6.2.3 Q3: Predictate logic proof tree

f.ex. the convergence example can also be written as a proof tree

Example showing the essence:

$$\frac{\frac{\text{modus ponens}}{P(a) \Rightarrow P(f(a)), P(f(a)) \Rightarrow P(f(f(a))), P(a) \vdash P(f(f(a)))}}{\forall_x P(x) \Rightarrow P(f(x)), P(a) \vdash P(f(f(a)))}$$

This corresponds to $A \Rightarrow B, B \Rightarrow C, A \vdash C$

or

$$\frac{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \vdash P(f(f(a)))}{\forall_x P(x) \Rightarrow P(f(x)), P(a) \vdash \exists_x P(f(f(x)))} \text{ (replace with } a\text{)} \quad \boxed{\leftrightarrow \quad \frac{\vdash \psi_{x \leftarrow t}}{\vdash \exists_x \psi} \text{ (replace with term)}}$$

or

$$\frac{\forall_x P(x) \Rightarrow P(f(x)), P(a) \vdash \forall_x P(f(f(x)))}{\forall_x P(x) \Rightarrow P(f(x)), \forall_x P(x) \vdash \forall_x P(f(f(b)))\ P(f(f(b)))}$$

6.3 Resolution

6.3.1 Resolution principle

The *resolution principle*, which is an inference rule:

$$\left.\begin{array}{l} L_1 \vee C_1 \\ L_2 \vee C_2 \end{array}\right\} \quad L_1\sigma = L_2\sigma \vdash C_1\sigma \vee C_2\sigma$$

$$\left.\begin{array}{c} \left(\begin{array}{c} \text{formulae which are not ground are} \xrightarrow{[\forall_x]} \\ \text{implicitly universally quantified} \end{array}\right) \quad \begin{array}{c} P(x, f(a)) \vee Q(x) \\ \overline{P}(b, y) \vee R(y) \end{array} \end{array}\right\} \quad \underbrace{Q(b) \vee R(f(x))}_{\text{"they are ground"}}$$

$$\text{Substitution } \sigma = \{x \leftarrow b, y \leftarrow f(x)\}$$

6.3.2 Resolution method

The *resolution method* is a proof system:

- Correct
- Complete

"Apply the resolution principle until you obtain the empty clause".

6.3.2.1 Correct

$$\frac{L_1\sigma \vee C_1\sigma}{\overline{L_2}\sigma \vee C_2\sigma}$$

Forall $d \in D : \langle P(x) \rangle^{\mathcal{I}}_{\{x \leftarrow d\}} = \mathbb{T}$

$$\begin{array}{ccc} \forall_x P(x) & \models & \forall_y P(f(y)) \\ & x \leftarrow f(y) & \\ P(x) & \vdash & P(f(y)) \end{array}$$

$$\begin{array}{l} \forall_x P(x) \vee Q(x) \\ \forall_x \overline{P}(x) \vee R(x) \end{array} \quad \models \quad \forall_x Q(x) \vee R(x)$$

$$\text{For any } d \in D : \langle Q(x) \vee R(x) \rangle^{\mathcal{I}}_{\{x \leftarrow d\}} = \mathbb{T}$$

$$\begin{array}{c} \mathscr{B}_\vee \left(\left\langle \overline{P}(x) \right\rangle^{\mathcal{I}}_{\{x \leftarrow d\}}, \langle Q(x) \rangle^{\mathcal{I}}_{\{x \leftarrow d\}} \right) \\ = \mathscr{B}_\vee \left(\mathscr{B}_\vee \left(\langle P(x) \rangle^{\mathcal{I}}_{\{x \leftarrow d\}} \right), \langle Q(x) \rangle^{\mathcal{I}}_{\{x \leftarrow d\}} \right) \\ \cdots \end{array}$$

6.3.2.2 Complete

For doing this, we will introduce a special domain ("Herbrand universe")

$$\begin{array}{rcl} H & = & \{a, f(x), f(f(x)), \dots\} \text{ (set of ground terms, which is } \infty_{\text{enumerable}}) \\ H_0 & = & \{ \text{ constants } \} \\ H_{k+1} & = & H_k \cup \{ \text{ terms obtained by applying all functional symbols to all elements of } H_k \} \\ H & = & \bigcup_{k \in \mathbb{N}} H_k \text{ (can be constructed in a systematic way)} \\ \mathcal{I} : & & \left\{ \begin{array}{l} H \\ f_I : H \to H \qquad f_I(t) = "f(" \smile t \smile ")" \quad \text{(where } \smile \text{ denotes a concatenation of strings)} \\ P_{\mathcal{I}} : H \to \{\mathbb{T}, \mathbb{F}\} \end{array} \right. \\ M & = & \{P(a), P(f(a)), P(f(f(a))), \dots\} \text{ ("atom set": set of ground atoms, which is } \infty_{\text{enumerable}}) \end{array}$$

An interpretation $\mathcal{I}$ is a list of ground literals (some are positive, some are not).

One can write all possible interpretations as a tree.

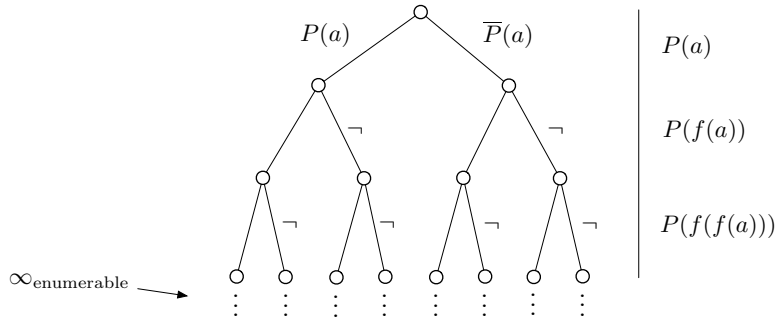


Figure 6.1: "Semantic tree"

$$\begin{array}{rcl} H\text{-satisfiable} & \text{iff} & \text{satisfiable} \\ \mathcal{I}_H & \Leftarrow & (\text{over some } \mathcal{I}/D) \end{array}$$

Take M and evaluating ground terms $M = \left\{ \begin{array}{cc} & \\ "P(a)" & , \quad "P(f(a))" \quad , \dots \\ \uparrow & \uparrow \\ P_{\mathcal{I}}(a_{\mathcal{I}}) = \mathbb{T} & P_{\mathcal{I}}(f_{\mathcal{I}}(a_{\mathcal{I}})) = \mathbb{T} \end{array} \right\}$

So practically it is sufficient so speak about terms "$P(a)$" instead of $P_I(a_I) = \mathbb{T}$

$$\begin{cases} P(a) & (1) \\ \overline{P}(x) \vee P(f(x)) \\ \overline{P}(f(f(a))) \end{cases}$$

So we can close the nodes

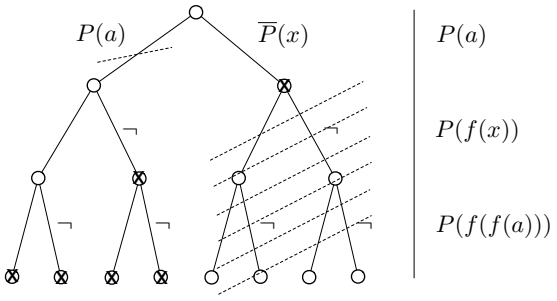


Figure 6.2: Semantic tree with closed nodes

$\Rightarrow$ The semantic tree is closed

Now if it would not be closed, one would have an infinite path somewhere where all clauses evaluate to true.

Conclusion: If φ is unsat, then the semantic tree is closed

$$\boxed{\begin{array}{c} A \\ \overline{A} \\ \overline{A} \vee \overline{B} \\ \overline{B} \vee C \end{array}} \quad \begin{cases} \bullet\ P(a) \\ \quad \overline{P}(x) \vee P(f(x)) \quad \swarrow\ x \leftarrow a \quad \nwarrow\ x \leftarrow f(a) \\ \bullet\ \overline{P}(f(f(a))) \end{cases}$$
$$\bullet\ \overline{P}(a) \vee P(f(a))$$
$$\bullet\ \overline{P}(f(a)) \vee P(f(f(a)))$$

Herbrand Theorem: If a formula is unsatisfiable, then there exists an unsatisfiable set of instances of the clauses in the formula

1. It is enough to reason about ground instances
2. Each time you introduce a new instance, you prove if it is unsatisfiable py propositional logic

Now this is not really an algorithm, because if it is satisfiable, it would never terminate, therefore this is a *semi-decidable procedure*, so undecidable.

There is no better method for 1st order predicate logic.

But, there are decidable classes of 1st order predicate logic.

But that's not really an limitation in real life from the practical view, because in practice it's more important if the answer is here in some time, so an decidable procedure, where the answer is here in 200 years is not so helpful ($\Rightarrow$ efficiency is more important as decidability)

6.4 Lifting lemma

ground resolution proof	$\mapsto$	resolution proof
if φ unsat then $\varphi \overset{\{Res\}}{\vdash} \square$		if $\varphi \not\overset{\{Res\}}{\vdash}$, then φ is sat
if S is valid, then $\overset{\{ND\}}{\vdash} S$		if $\varphi \not\overset{\text{"consistent"}\ \{Res\}}{\vdash} \mathbb{F}$, then φ is sat
		$\downarrow$ (there exists an interpretation ("model"))

Natural deduction: "S is provable by ND calculus"

Gödels completeness theorem

$$\frac{\Phi, \varphi_{x\leftarrow a} \vdash \Psi}{\Phi, \exists_x \varphi \vdash \Psi}$$
$$\downarrow$$
$$\frac{\Phi, \varphi_{x\leftarrow a}, \neg\Psi \vdash \mathbb{F}}{\Phi, \exists_x \varphi, \neg\Psi \vdash \mathbb{F}}$$

If $(\Phi \wedge \varphi_{x\leftarrow a})$ unsat, then $(\Phi \wedge \exists_x \varphi)$ unsat
(even iff, so this is an "reversible rule"!)

$$\frac{\Phi \vdash \varphi_{x\leftarrow a}}{\Phi \vdash \forall_x \varphi} \qquad \frac{\Phi, (\neg\varphi)_{x\leftarrow a} \vdash}{\Phi, \underbrace{\neg\forall_x \varphi}_{\exists_x (\neg\varphi)} \vdash}$$

not reversible $\not\uparrow \downarrow$ $\quad \dfrac{\Phi, \varphi_{x\leftarrow t} \vdash}{\Phi, \forall_x \varphi \vdash} \qquad \dfrac{\Phi, \forall_x \varphi, \varphi_{x\leftarrow t} \vdash}{\Phi, \forall_x \varphi \vdash} \uparrow\downarrow$

So the rule is correct $\downarrow$ If $\Phi, \varphi_{x\leftarrow t} \vdash \mathbb{F}$ (unsatisfiable !) then $\Phi, \forall_x \varphi \vdash \mathbb{F}$ (because $\forall_x \varphi \vdash \varphi_{x\leftarrow t}$)

6.5 Predicate logic with equality

= is a predicate symbol

($\diamond$) properties:
- equivalence
  - reflexive
  - symmetric
  - transitive
- "replacement rule"
  - if $t_1 = t_2$ then $\tau_{x\leftarrow t_1} = \tau_{x\leftarrow t_2}$
  - $a = b \vdash f(a) = f(b)$
  - $\tau : f(x)$

6.5.1 Explicit treatment

We used this in a lot of proofs: $A \wedge B = B \wedge A \ \vdash\ C \Rightarrow (A \wedge B) = C \Rightarrow (B \wedge A)$

f.ex. $\forall_x x = x$

f.ex. $\forall_x \forall_y x = y \ \Rightarrow\ f(x) = f(y)$
$\forall_x \forall_y x = x' \ \Rightarrow\ g(x, y, z) = g(x', y, z)$

"explicit treatment of equality"

6.5.2 Implicit treatment

Interpretation for an equality formula

$$I \quad = \quad \begin{cases} D \\ a_I, f_I, P_I \\ =_I \ \ldots \text{ has the properties of equalities (as in } \diamond) \end{cases}$$

"implicit treatment of equality"

6.5.3 Inference rules

$$\frac{\Phi, t_1 = t_2, \tau_{x \leftarrow t_1} = \tau_{x \leftarrow t_2} \vdash \Psi}{\Phi, t_1 = t_2 \vdash \Psi}$$

6.5.4 Programming language

replacement $t_1 = t_2 \vdash \tau_{x \leftarrow t_1} = \tau_{x \leftarrow t_2}$

$\nearrow$

$\searrow$

substitution $\forall_x \varphi \vdash \varphi_{x \leftarrow t}$

Those two principles can be used as a programming language

6.5.4.1 Example with GCD

"Conditional rewriting":

$$\begin{cases} GCD(x, y) = GCD(y, x - y) & (\text{if } x \geq y) \\ GCD(x, y) = GCD(x, y - x) & (\text{if } y > x) \\ GCD(x, 0) = x \end{cases}$$

Can be instantiated, f.ex.:

$$\begin{aligned} GCD(15, 12) &= GCD(12, 3) = GCD(3, 9) = GCD(3, 6) \\ &= GCD(3, 3) = GCD(3, 0) = 3 \end{aligned}$$

This is also a proof, sequence of expressions obtained by applying certain transformations.

6.5.5 Proof equalities

f.ex.:

$$\left(a^2 + 1\right) - \left(b^2 + 1\right) \quad = \quad (a + b) * (a - b)$$

We identify *rewrite rules* so that we obtain a *normal form*

In this case of polynomials those are the rules for distributivity and rules for sorting, so that we obtain the same syntactical form.

There is a general method called *Knuth-Bendix completion algorithm* (unfortunatly only semi-decidable)

Hint (of the editor): see `http://mathworld.wolfram.com/Knuth-BendixCompletionAlgorithm.html`

6.6 Logic Programming: PROLOG

One can declare facts: $B(A, J)$

And one can declare rules:

$$\cancel{\forall_{x,z}} B(x, y) \Leftarrow \cancel{\exists_z} B(x, z) \wedge B(z, y)$$
(quantifiers implicit)

These rules are like *horn clauses*, have the form $P_1 \wedge \cdots \wedge P_n \Rightarrow Q$

Query:

$$\begin{array}{rl} ?\ B(A, C) & x \leftarrow A, y \leftarrow C \text{ (substitution)} \\ & \text{(unification)} \\ B(x, z) \wedge B(z, y) & \\ B(A, J) \wedge B(J, C) & \end{array}$$