Natural Deduction 2

Stéphane Devismes Pascal Lafourcade Michel LÃ©vy
Jean-FranÃ§ois Monin (jean-francois.monin@imag.fr)

UniversitÃ© Joseph Fourier, Grenoble I

February 13, 2015

Plan

Theorem

Theorem 3.3.1

If a formula A is deduced from an environment Γ (i.e., if $\Gamma \vdash A$) then A is a consequence of Γ ($\Gamma \models A$).

Every proof written in an environment Γ is correct !

Theorem

Theorem 3.3.1

If a formula A is deduced from an environment Γ (i.e., if $\Gamma \vdash A$) then A is a consequence of Γ ($\Gamma \models A$).

Every proof written in an environment Γ is correct !

Proof by induction :

- Let Γ a set of formulae.
- Let P a proof of A in this environment.
- Let C_i the conclusion and H_i the context of $i^{\text{Ã¨me}}$ line of P.
- Let Γ, H_i the set of formulae of Γ and of the list H_i.

Show that for every k we have $\Gamma, H_k \models C_k$.

Theorem

Theorem 3.3.1

If a formula A is deduced from an environment Γ (i.e., if $\Gamma \vdash A$) then A is a consequence of Γ ($\Gamma \models A$).

Every proof written in an environment Γ is correct !

Proof by induction :

- Let Γ a set of formulae.
- Let P a proof of A in this environment.
- Let C_i the conclusion and H_i the context of $i^{\text{Ã¨me}}$ line of P.
- Let Γ, H_i the set of formulae of Γ and of the list H_i.

Show that for every k we have $\Gamma, H_k \models C_k$.

For the last line n of the proof : H_n is empty and $C_n = A$

Theorem

Theorem 3.3.1

If a formula A is deduced from an environment Γ (i.e., if $\Gamma \vdash A$) then A is a consequence of Γ ($\Gamma \models A$).

Every proof written in an environment Γ is correct !

Proof by induction :

- Let Γ a set of formulae.
- Let P a proof of A in this environment.
- Let C_i the conclusion and H_i the context of iÃ¨me line of P.
- Let Γ, H_i the set of formulae of Γ and of the list H_i.

Show that for every k we have $\Gamma, H_k \models C_k$.

For the last line n of the proof : H_n is empty and $C_n = A$

Hence, $\Gamma \models A$.

Base case

Suppose that A is derived from Γ by an empty proof.

That is, A is a member of Γ.

Hence $\Gamma \models A$.

Induction hypothesis

Suppose that for every line $i < k$ of the proof we have $\Gamma,\ H_i \models C_i$.

Let us show $\Gamma,\ H_k \models C_k$.

Induction hypothesis

Suppose that for every line $i < k$ of the proof we have $\Gamma,\ H_i \models C_i$.

Let us show $\Gamma,\ H_k \models C_k$.

Three possible cases :

- Line k is « `Suppose` C_k ».
- Line k is « `Hence` C_k ».
- Line k is « C_k ».

Line k is « `Suppose` C_k »

The formula C_k is the last formula of H_k.

Then $H_k \models C_k$.

Then $\Gamma, H_k \models C_k$.

The line k is « Hence C_k »

C_k is the formula $B \Rightarrow D$ where :

- B is the last formula of H_{k-1} and
- D :

.

The line k is ≪ `Hence` C_k ≫

C_k is the formula $B \Rightarrow D$ where :

- B is the last formula of H_{k-1} and
- D :

(1) D is a formula of Γ.

.

(2) D is usable on the previous line.

The line k is « `Hence` C_k »

C_k is the formula $B \Rightarrow D$ where :

- B is the last formula of H_{k-1} and
- D :

(1) D is a formula of Γ. D is then a consequence of $\Gamma,\ H_k$. Since $B \Rightarrow D$ is a consequence of D, we conclude $\Gamma,\ H_k \models C_k$.

(2) D is usable on the previous line.

The line k is « `Hence` C_k »

C_k is the formula $B \Rightarrow D$ where :

- B is the last formula of H_{k-1} and
- D :

(1) D is a formula of Γ. D is then a consequence of Γ, H_k. Since $B \Rightarrow D$ is a consequence of D, we conclude $\Gamma, H_k \models C_k$.

(2) D is usable on the previous line. Hence there exists $i < k$ such that $D = C_i$ and H_i is a prefix of H_{k-1}.
By induction hypothesis, $\Gamma, H_i \models D$.
Since H_i is a prefix of H_{k-1}, we have $\Gamma, H_{k-1} \models D$.
Since B is the last formula of H_{k-1}, $H_{k-1} = H_k, B$ and therefore $\Gamma, H_k, B \models D$, which implies $\Gamma, H_k \models B \Rightarrow D$, that is, $\Gamma, H_k \models C_k$.

Line k is « C_k »

This formula is the conclusion of a rule of table 3.1, applied to its usable premises on the previous line or to the element of Γ.

Line k is « C_k »

This formula is the conclusion of a rule of table 3.1, applied to its usable premises on the previous line or to the element of Γ.

We only consider the rule ∧I, the other cases being similar.

Line k is $\ll C_k \gg$

This formula is the conclusion of a rule of table 3.1, applied to its usable premises on the previous line or to the element of Γ.

We only consider the rule $\wedge$I, the other cases being similar.

The formula C_k is $(D \wedge E)$ and the premises of the rule are D and E.

Line k is ≪ C_k ≫

This formula is the conclusion of a rule of table 3.1, applied to its usable premises on the previous line or to the element of Γ.

We only consider the rule $\wedge$I, the other cases being similar.

The formula C_k is $(D \wedge E)$ and the premises of the rule are D and E.

Since D and E are elements of Γ or usable on the previous line, as in the previous case, using the induction hypothesis, we have : $\Gamma,\ H_{k-1} \models D$ et $\Gamma,\ H_{k-1} \models E$.

Line k is « C_k »

This formula is the conclusion of a rule of table 3.1, applied to its usable premises on the previous line or to the element of Γ.

We only consider the rule $\wedge$I, the other cases being similar.

The formula C_k is $(D \wedge E)$ and the premises of the rule are D and E.

Since D and E are elements of Γ or usable on the previous line, as in the previous case, using the induction hypothesis, we have : $\Gamma,\ H_{k-1} \models D$ et $\Gamma,\ H_{k-1} \models E$.

Since the line k does not change the hypotheses, we have $H_{k-1} = H_k$, hence $\Gamma, H_k \models D$ and $\Gamma, H_k \models E$.

Line k is « C_k »

This formula is the conclusion of a rule of table 3.1, applied to its usable premises on the previous line or to the element of Γ.

We only consider the rule ∧I, the other cases being similar.

The formula C_k is $(D \wedge E)$ and the premises of the rule are D and E.

Since D and E are elements of Γ or usable on the previous line, as in the previous case, using the induction hypothesis, we have : $\Gamma,\ H_{k-1} \models D$ et $\Gamma,\ H_{k-1} \models E$.

Since the line k does not change the hypotheses, we have $H_{k-1} = H_k$, hence $\Gamma, H_k \models D$ and $\Gamma, H_k \models E$.

Since C_k is $(D \wedge E)$, we have : $D,\ E \models C_k$. Therefore $\Gamma,\ H_k \models C_k$.

Plan

Correctness

Completeness

Tactics

Conclusion

Theorem

We prove the completeness of the rules only for formulas containing the following logic symbols : $\bot$, $\wedge$, $\vee$, $\Rightarrow$.

This is enough because additional symbols $\top$, $\neg$ and $\Leftrightarrow$ can be regarded as abbreviations.

Theorem

We prove the completeness of the rules only for formulas containing the following logic symbols : $\bot$, $\wedge$, $\vee$, $\Rightarrow$.

This is enough because additional symbols $\top$, $\neg$ and $\Leftrightarrow$ can be regarded as abbreviations.

Theorem 3.4.1

Let Γ be a finite set of formulae and A a formula, if $\Gamma \models A$ then $\Gamma \vdash A$.

Notations

We define a literal as a variable or an implication between a variable and $\perp$.

Notations

We define a literal as a variable or an implication between a variable and $\perp$.

Let x be a variable, x and $x \Rightarrow \perp$ (which can be abbreviated as $\neg x$) are complementary literals.

Notations

We define a literal as a variable or an implication between a variable and $\bot$.

Let x be a variable, x and $x \Rightarrow \bot$ (which can be abbreviated as $\neg x$) are complementary literals.

Given Γ, a list of formulae, $s(\Gamma)$ denotes the set of formulae of Γ.

Notations

We define a literal as a variable or an implication between a variable and $\bot$.

Let x be a variable, x and $x \Rightarrow \bot$ (which can be abbreviated as $\neg x$) are complementary literals.

Given Γ, a list of formulae, $s(\Gamma)$ denotes the set of formulae of Γ.

To simplify notations, we use the comma for adding an element at the begining or at the end of the list and for concatenating two lists, which can be either lists of formulae or lists of proofs.

Measure

Then measure m of formulae and of lists of formulae is defined as :

- $m(\bot) = 0$,
- $m(x) = 1$ where x is a variable,
- $m(\Rightarrow) = 1$,
- $m(\wedge) = 1$,
- $m(\vee) = 2$,
- $m(A \circ B) = m(A) + m(\circ) + m(B)$,
- the measure of a list of formulae is the sum of the measures of the formulae of the list.

Measure

Then measure m of formulae and of lists of formulae is defined as :

- $m(\bot) = 0$,
- $m(x) = 1$ where x is a variable,
- $m(\Rightarrow) = 1$,
- $m(\wedge) = 1$,
- $m(\vee) = 2$,
- $m(A \circ B) = m(A) + m(\circ) + m(B)$,
- the measure of a list of formulae is the sum of the measures of the formulae of the list.

Since $\neg A$ is an abbreviation of $A \Rightarrow \bot$, we have :
$m(\neg A) = m(A \Rightarrow \bot) = m(A) + 1$.

Measure

Then measure m of formulae and of lists of formulae is defined as :

- $m(\bot) = 0$,
- $m(x) = 1$ where x is a variable,
- $m(\Rightarrow) = 1$,
- $m(\wedge) = 1$,
- $m(\vee) = 2$,
- $m(A \circ B) = m(A) + m(\circ) + m(B)$,
- the measure of a list of formulae is the sum of the measures of the formulae of the list.

Since $\neg A$ is an abbreviation of $A \Rightarrow \bot$, we have :
$m(\neg A) = m(A \Rightarrow \bot) = m(A) + 1$.

For example, let $A = (a \vee \neg a)$. We have $m(\neg a) = 2$, $m(A) = 5$ and $m(A,\ (b \wedge b),\ A) = 13$.

Induction

We define $P(n)$ to be the following property :
given any list of formulae Γ and formula A such that the measure of $\Gamma,\ A$ is n, we have if $s(\Gamma) \models A$ then $s(\Gamma) \vdash A$.

Induction

We define $P(n)$ to be the following property :
given any list of formulae Γ and formula A such that the measure of $\Gamma,\ A$ is n, we have if $s(\Gamma) \models A$ then $s(\Gamma) \vdash A$.

To show that $P(n)$ holds for every integer n, we use « strong » induction :

Induction

We define $P(n)$ to be the following property :
given any list of formulae Γ and formula A such that the measure of $\Gamma,\ A$ is n, we have if $s(\Gamma) \models A$ then $s(\Gamma) \vdash A$.

To show that $P(n)$ holds for every integer n, we use « strong » induction :

Suppose that for every $i < k$, $P(i)$ holds ; then show that $P(k)$ holds as well.
To this effect suppose moreover $m(\Gamma,\ A) = k$ and $s(\Gamma) \models A$, then show : $s(\Gamma) \vdash A$.

Decomposition

Idea : we decompose Γ, A in order to apply the induction hypothesis.

Decomposition

Idea : we decompose Γ, A in order to apply the induction hypothesis.

A is undecomposable if A is $\perp$ or a variable and Γ is undecomposable if Γ is a list of literals or contain the formula $\perp$.

Decomposition

Idea : we decompose Γ, A in order to apply the induction hypothesis.

A is undecomposable if A is $\perp$ or a variable and Γ is undecomposable if Γ is a list of literals or contain the formula $\perp$.

We study three cases :

Case 1 : neither A, nor Γ is decomposable.

Case 2 : A is decomposable.
We decompose A in two sub-formulae B and C.
We obtain the following inequalities : $m(\Gamma,\ B) < m(\Gamma,\ A)$ and $m(\Gamma,\ C) < m(\Gamma,\ A)$.

Case 3 : Γ is decomposable. We permute Γ in order to obtain a list and a decomposable formula.

Case 1 : neither A, nor Γ are decomposable

Then A is $\perp$ or a variable and Γ is a list of literals or contains the formula $\perp$

Case 1 : neither A, nor Γ are decomposable

Then A is $\perp$ or a variable and Γ is a list of literals or contains the formula $\perp$

(a) Either $\perp$ is a formula of Γ ; then A can be derived from $\perp$ by the rule *Efq*, hence $s(\Gamma) \vdash A$.

Case 1 : neither A, nor Γ are decomposable

Then A is $\perp$ or a variable and Γ is a list of literals or contains the formula $\perp$

(a) Either $\perp$ is a formula of Γ ; then A can be derived from $\perp$ by the rule *Efq*, hence $s(\Gamma) \vdash A$.

(b) Or $\perp$ is not a formula of Γ. Hence, Γ is a list of literals and we have two cases :

Case 1 : neither A, nor Γ are decomposable

Then A is $\perp$ or a variable and Γ is a list of literals or contains the formula $\perp$

(a) Either $\perp$ is a formula of Γ ; then A can be derived from $\perp$ by the rule *Efq*, hence $s(\Gamma) \vdash A$.

(b) Or $\perp$ is not a formula of Γ. Hence, Γ is a list of literals and we have two cases :

- $A = \perp$. Since $s(\Gamma) \models A$, the list Γ contains two complementary literals, therefore A can be derived from Γ by the rule $\Rightarrow$E, and consequently $s(\Gamma) \vdash A$ (by the proof $\perp, A$).

Case 1 : neither A, nor Γ are decomposable

Then A is $\perp$ or a variable and Γ is a list of literals or contains the formula $\perp$

(a) Either $\perp$ is a formula of Γ ; then A can be derived from $\perp$ by the rule *Efq*, hence $s(\Gamma) \vdash A$.

(b) Or $\perp$ is not a formula of Γ. Hence, Γ is a list of literals and we have two cases :

- $A = \perp$. Since $s(\Gamma) \models A$, the list Γ contains two complementary literals, therefore A can be derived from Γ by the rule $\Rightarrow$E, and consequently $s(\Gamma) \vdash A$ (by the proof $\perp, A$).
- A is a variable. Since $s(\Gamma) \models A$:

Case 1 : neither A, nor Γ are decomposable

Then A is $\perp$ or a variable and Γ is a list of literals or contains the formula $\perp$

(a) Either $\perp$ is a formula of Γ ; then A can be derived from $\perp$ by the rule *Efq*, hence $s(\Gamma) \vdash A$.

(b) Or $\perp$ is not a formula of Γ. Hence, Γ is a list of literals and we have two cases :

- $A = \perp$. Since $s(\Gamma) \models A$, the list Γ contains two complementary literals, therefore A can be derived from Γ by the rule $\Rightarrow$E, and consequently $s(\Gamma) \vdash A$ (by the proof $\perp, A$).
- A is a variable. Since $s(\Gamma) \models A$:
  - let Γ contains two complementary literals and, as in the previous case, we have $s(\Gamma) \vdash A$

Case 1 : neither A, nor Γ are decomposable

Then A is $\bot$ or a variable and Γ is a list of literals or contains the formula $\bot$

(a) Either $\bot$ is a formula of Γ ; then A can be derived from $\bot$ by the rule *Efq*, hence $s(\Gamma) \vdash A$.

(b) Or $\bot$ is not a formula of Γ. Hence, Γ is a list of literals and we have two cases :

- $A = \bot$. Since $s(\Gamma) \models A$, the list Γ contains two complementary literals, therefore A can be derived from Γ by the rule $\Rightarrow$E, and consequently $s(\Gamma) \vdash A$ (by the proof $\bot, A$).
- A is a variable. Since $s(\Gamma) \models A$:
  - let Γ contains two complementary literals and, as in the previous case, we have $s(\Gamma) \vdash A$
  - let A is element of Γ and in this case we also have $s(\Gamma) \vdash A$ (by empty proof).

Case 2 : A is decomposable into B and C

A is decomposed into $(B \wedge C)$, $(B \vee C)$, or $(B \Rightarrow C)$.

We only study the case $A = (B \wedge C)$, the other cases are similar.

Case 2 : A is decomposable into B and C

A is decomposed into $(B \wedge C)$, $(B \vee C)$, or $(B \Rightarrow C)$.

We only study the case $A = (B \wedge C)$, the other cases are similar.

Since $s(\Gamma) \models A$ and $A = (B \wedge C)$, we have $s(\Gamma) \models B$ and $s(\Gamma) \models C$.

Case 2 : A is decomposable into B and C

A is decomposed into $(B \wedge C)$, $(B \vee C)$, or $(B \Rightarrow C)$.

We only study the case $A = (B \wedge C)$, the other cases are similar.

Since $s(\Gamma) \models A$ and $A = (B \wedge C)$, we have $s(\Gamma) \models B$ and $s(\Gamma) \models C$.

The measures of B and C are strictly less than the measure of A.

Case 2 : A is decomposable into B and C

A is decomposed into $(B \wedge C)$, $(B \vee C)$, or $(B \Rightarrow C)$.

We only study the case $A = (B \wedge C)$, the other cases are similar.

Since $s(\Gamma) \models A$ and $A = (B \wedge C)$, we have $s(\Gamma) \models B$ and $s(\Gamma) \models C$.

The measures of B and C are strictly less than the measure of A.

Hence $m(\Gamma, B) < k$ and $m(\Gamma, C) < k$, and by induction hypothesis, there exist two proofs P and Q such that $s(\Gamma) \vdash P : B$ and $s(\Gamma) \vdash Q : C$.

Case 2 : A is decomposable into B and C

A is decomposed into $(B \wedge C)$, $(B \vee C)$, or $(B \Rightarrow C)$.

We only study the case $A = (B \wedge C)$, the other cases are similar.

Since $s(\Gamma) \models A$ and $A = (B \wedge C)$, we have $s(\Gamma) \models B$ and $s(\Gamma) \models C$.

The measures of B and C are strictly less than the measure of A.

Hence $m(\Gamma, B) < k$ and $m(\Gamma, C) < k$, and by induction hypothesis, there exist two proofs P and Q such that $s(\Gamma) \vdash P : B$ and $s(\Gamma) \vdash Q : C$.

Since A can be derived from B and C by the rule $\wedge$I : $\ll P,\ Q,\ A \gg$ is a proof of A in the environment $s(\Gamma)$ hence $s(\Gamma) \vdash A$.

Case 3 : Γ is decomposable

Six cases may occur :

- Γ is a permutation of the list $(B \wedge C), \Delta$.
- Γ is a permutation of the list $(B \vee C), \Delta$.
- Γ is a permutation of the list $(B \Rightarrow C), \Delta$ or $C \neq \perp$.
- Γ is a permutation of the list $((B \wedge C) \Rightarrow \perp), \Delta$.
- Γ is a permutation of the list $((B \vee C) \Rightarrow \perp), \Delta$.
- Γ is a permutation of the list $((B \Rightarrow C) \Rightarrow \perp), \Delta$.

We only study the first case.

Γ is a permutation of the list $(B \wedge C), \Delta$

Γ is a permutation of the list $(B \wedge C), \Delta$

Γ and $(B \wedge C), \Delta$ have the same measure.

Γ is a permutation of the list $(B \wedge C), \Delta$

Γ and $(B \wedge C), \Delta$ have the same measure.

Since $s(\Gamma) \models A$, $s(B, C, \Delta) \models A$.

Γ is a permutation of the list $(B \wedge C), \Delta$

Γ and $(B \wedge C), \Delta$ have the same measure.

Since $s(\Gamma) \models A$, $s(B, C, \Delta) \models A$.

The sum of the measures of B and C is strictly less than the measure of $B \wedge C$.

Γ is a permutation of the list $(B \wedge C), \Delta$

Γ and $(B \wedge C), \Delta$ have the same measure.

Since $s(\Gamma) \models A$, $s(B, C, \Delta) \models A$.

The sum of the measures of B and C is strictly less than the measure of $B \wedge C$.

Hence $m(B, C, \Delta, A) < m((B \wedge C), \Delta, A) = m(\Gamma, A) = k$, by induction hypothesis, there exist a proof P such that $s(B, C, \Delta) \vdash P : A$.

Γ is a permutation of the list $(B \wedge C), \Delta$

Γ and $(B \wedge C), \Delta$ have the same measure.

Since $s(\Gamma) \models A$, $s(B, C, \Delta) \models A$.

The sum of the measures of B and C is strictly less than the measure of $B \wedge C$.

Hence $m(B, C, \Delta, A) < m((B \wedge C), \Delta, A) = m(\Gamma, A) = k$, by induction hypothesis, there exist a proof P such that $s(B, C, \Delta) \vdash P : A$.

Since B can be derived from $(B \wedge C)$ by the rule $\wedge$E1 and C can be derived from $(B \wedge C)$ by the rule $\wedge$E2 : $\ll B,\ C,\ P \gg$ is a proof of A in the environment $s(\Gamma)$, hence $s(\Gamma) \vdash A$.

Plan

Remark 3.4.2

The proof of completeness is constructive, that is it provides a complete set of tactics to construct the proofs of a formula in an environment.

However, these tactics can lead to long proofs.

Remark 3.4.2

The proof of completeness is constructive, that is it provides a complete set of tactics to construct the proofs of a formula in an environment.

However, these tactics can lead to long proofs.

It is better then to use ≪ optimised ≫ tactics presented in section 3.2.

For example, to prove $B \vee C$:

- First try to prove B
- If failure, then try to prove C
- Otherwise, use tactic 10 (prove C under the hypothesis $\neg B$)

Proof tactics

We wish to prove A in the environment Γ

The 13 following tactics must be used in the following order !

Tactic 1

If $A \in \Gamma$ then the empty proof is obtained.

Tactic 2

If A is the consequence of a rule whose premises are in Γ, then the obtained proof is
$\ll A \gg$.

Tactic 3

If Γ contains a contradiction, that is a formula B and a formula $\neg B$, then the obtained proof is $\ll \bot, A \gg$.

Tactic 4

If A is $B \wedge C$ then :

- prove B : Let P the proof obtained for B,
- prove C : Let Q the proof obtained for C.

The proof obtained for A is $\ll P, Q, A \gg$.

The proofs can fail (if it is asked to prove a formula that is unprovable in the given environment) : if the proof of B or C fails, it is the same for the proof of A. To simplify the remaining, we do not highlight the failure cases anymore, unless they must be followed by another proof.

Tactic 5

If A is $B \Rightarrow C$, then prove C under hypothesis B (B is added to the environment).

Let P, the proof obtained for C.

The proof obtained for A is « `Suppose` B, P, `Hence` A ».

Tactic 6

If A is $B \vee C$, then prove B :
If P is the proof obtained for B, then $\ll P, A \gg$ is the proof obtained for A.

If the proof of B fails then prove C :
If P is the proof obtained for C then $\ll P, A \gg$ is the proof obtained for A.

If the proof of C fails, try the following rules.

Tactic 7

If $B \wedge C$ is in the environment, then prove A starting from formulae B, C, replacing $B \wedge C$ in the environment and let P the result of this proof.

Then $\ll B, C, P \gg$ is a proof of A in the initial environment.

Tactic 8

If $B \vee C$ is in the environment, then :

- prove A in the environment where B replaces $B \vee C$: Let P the obtained proof,
- prove A in the environment where C replaces $B \vee C$: Let Q the obtained proof.

The proof of A is then ≪ `Suppose` B, P, `Hence` $B \Rightarrow A$, `Suppose` C, Q, `Hence` $C \Rightarrow A$, A ≫.

Tactic 9

If $\neg(B \vee C)$ is in the environment, then

- we derive $\neg B$ by the proof $P4$ and
- $\neg C$ by the proof $P5$ (proofs requested in exercise 58).
- Let P the proof of A in the environment where $\neg B$, $\neg C$ replace the formula $\neg(B \vee C)$.

The proof of A is $\ll P4, P5, P \gg$.

Tactic 10

If A is $B \vee C$, then prove C under hypothesis $\neg B$: let P the obtained proof.
$\ll$ `Suppose` $\neg B$, P, `Hence` $\neg B \Rightarrow C \gg$ is a proof of the formula $\neg B \Rightarrow C$ which is equivalent to A.

To obtain the proof of A, simply add the proof $P1$, requested in exercise 58 of A in the environment $\neg B \Rightarrow C$.
The proof obtained from A is therefore $\ll$ `Suppose` $\neg B$, P, `Hence` $\neg B \Rightarrow C$, $P1 \gg$.

Tactic 11

If $\neg(B \wedge C)$ is in the environment, then we deduce from it $\neg B \vee \neg C$ by the proof $P3$ requested in exercise 58 then we reason case by case as follows :

- prove A in the environment where $\neg B$ replaces $\neg(B \wedge C)$: Let P the obtained proof,
- prove A in the environment where $\neg C$ replaces $\neg(B \wedge C)$: Let Q the obtained proof.

The proof of A is $\ll P3$, `Suppose` $\neg B$, P, `Hence` $\neg B \Rightarrow A$, `Suppose` $\neg C$, Q, `Hence` $\neg C \Rightarrow A$, $A \gg$.

Tactique 12

If $\neg(B \Rightarrow C)$ is in the environment, then

- we derive B by the proof $P6$,
- $\neg C$ by the proof $P7$ (proofs requested in exercise 58).
- Let P the proof of A in the environment where B, $\neg C$ replace the formula $\neg(B \Rightarrow C)$.

The proof of A is $\ll P6, P7, P \gg$.

Tactic 13

If $B \Rightarrow C$ is in the environment and if $C \neq \bot$, i.e. if $B \Rightarrow C$ is not $\neg B$, then,
we derive $\neg B \vee C$ in the environment $B \Rightarrow C$ by proof $P2$ from exercise 58, then we reason by cases :

- prove A in the environment where $\neg B$ replaces $B \Rightarrow C$: Let P the obtained proof,
- prove A in the environment where C replaces $B \Rightarrow C$: Let Q the obtained proof.

The proof of A is $\ll P2$, `Suppose` $\neg B$, P, `Hence` $\neg B \Rightarrow A$, `Suppose` C, Q, `Hence` $C \Rightarrow A$, $A \gg$.

Example

Proof of Peirce's formula :

$$((p \Rightarrow q) \Rightarrow p) \Rightarrow p$$

Proof plan

Tactic 5 is compulsory !

Proof Q :

`Suppose` $(p \Rightarrow q) \Rightarrow p$

Q_1 proof or p in the environment $(p \Rightarrow q) \Rightarrow p$

`Hence` $((p \Rightarrow q) \Rightarrow p) \Rightarrow p$

Proof Q_1 necessarily uses tactic 13. Hence this proof is written : In the environment $B \Rightarrow C$ where $B = p \Rightarrow q,\ C = p$.

Plan of the proof of Q_1

Proof Q_1 :

$Q_{11} = P_2$ where P_2 is the proof of $\neg B \vee C$ in the environment $B \Rightarrow C$, see exercise 58

Suppose $\neg B$

Q_{12} proof of $A = p$ in the environment $\neg B$

Hence $\neg B \Rightarrow A$

Suppose C

Q_{13} proof of $A = p$ in the environment C

Hence $C \Rightarrow A$

A

Proof of Q_1

Q_{13} is the empty proof, since $A = C = p$.

Q_{12} is the proof of $C = p$ in the environment $\neg(p \Rightarrow q)$.
Since $\neg A$ is an abbreviation of $A \Rightarrow \bot$, this proof is the proof P_6 requested in exercise 58, where $B = p$ and $C = q$.

By gluing pieces $Q_1,\ Q_{11},\ Q_{12},\ Q_{13}$, we obtain the proof Q.

Below we show how to find the proof Q_{12} without using the tactics.

Proof of Q_{12}

The only rule, which does not lead to a deadlock, is the reduction ad absurdum. Hence this proof is of the form :

Proof Q_{12} of p in the environment $\neg(p \Rightarrow q)$
Suppose $\neg p$
Q_{121} proof of $\perp$ in the environment $\neg(p \Rightarrow q)$, $\neg p$
Hence $\neg\neg p$
p

To obtain a contradiction, hence a proof of $\perp$, $p \Rightarrow q$ must be derived. Hence the proof Q_{121} is :

Suppose p
$\perp$
q
Hence $p \Rightarrow q$
$\perp$

Plan

Automated proofs

To automatically obtain the proofs in the system, one recommends to use the following software (implementing the 13 previous tactics) :

http://teachinglogic.liglab.fr/DN/

People who prefer proofs in the form of trees can use the following software :

http://www-sop.inria.fr/marelle/Laurent.Thery/peanoware/Nd.html

Conclusion : Next course

- First-order logic

Homework : solution using ND

$$(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m) \Rightarrow m \vee p$$

Conclusion

Thank you for your attention.

Questions ?