Workbook Unit 13:

Natural Deduction Proofs (IV)

Overview

This unit

- introduces the remaining inference rules ≡Int, ∨Elim, ~Int and ~Elim
- explains how tautologies are proved in system SD

Prerequisites

You need to have completed Units 10, 11, 12

drp@swps.edu.pl

1. The Biconditional Introduction Rule (≡Int)

If in one subderivation whose assumption is *p* you can derive *r*, and in another subderivation whose assumption is *r* you can derive *p,* then you are allowed to add a line to the derivation with the biconditional *p* ≡ *r*.

i.	| *p*	Assp (≡Int)
j.	| *r*	
k.	| *r*	Assp (≡Int)
l.	| *p*	
➤	*p* ≡ *r*	≡ Int *i–j*, *k–l*

Intuitions

This rule is quite intuitive if you understand the →Int rule and if you understand that a biconditional "p if and of if r" is a conjunction of two conditionals: p if r (r → p) and p only if r (p → r).

Constructing subderivations for rule ≡Int

The ≡Int rule is again a structural rule. Whenever we want to introduce a biconditional into the derivation, we must first construct two sister subderivations and the ≡Int rule tells us exactly what the assumptions of those subderivations will be and what conclusions we will need to derive. Practice constructing such derivations in the exercises below.

Exercises on Applying ≡Int

≡Int.I.a. In all of the following proof schemata, you are asked to apply the ≡Int rule to derive a certain biconditional. Fill in the information missing in step 9. (Note that the point of the exercise is not to actually construct the whole proof!)

(a)

3	| B	Assp (≡Int)
5	| A	
6	| A	Assp (≡Int)
8	| B	
9		≡Int 3–5, 6–8

(b)

3	| C	Assp (≡Int)
5	| ~A	
6	| ~A	Assp (≡Int)
8	| C	
9		≡Int 3–5, 6–8

(c)

3	B → C	Assp (≡Int)
5	C → B	
6	C → B	Assp (≡Int)
8	B → C	
9		≡Int 3–5, 6–8

(d)

3	A ∨ C	Assp (≡Int)
5	~C → A	
6	~C → A	Assp (≡Int)
8	A ∨ C	
9		≡Int 3–5, 6–8

→Int.I.b. In all of the following proof schemata, you are asked to apply the ≡Int rule to derive a biconditional (line 9). To do that we need construct two sister subderivations. Your task is to fill in the assumptions and the conclusions of both subderivations. (As before, the point of the exercise is not to actually construct the whole proof!)

(a)

3		Assp (≡Int)
5		
6		Assp (≡Int)
8		
9	D ≡ A	≡Int 3–5, 6–8

(b)

3		Assp (≡Int)
5		
6		Assp (≡Int)
8		
9	(A → C) ≡ (C → B)	≡Int 3–5, 6–8

(c)

3		Assp (≡Int)
5		
6		Assp (≡Int)
8		
9	~~B ≡ ~(A • ~B)	≡Int 3–5, 6–8

(d)

3		Assp (≡Int)
5		
6		Assp (≡Int)
8		
9	(A ≡ B) ≡ C	≡Int 3–5, 6–8

Examples of Proofs Using ≡Int

Example 1.

To show that the biconditional A ≡ B follows from the conjunction (A → B) • (B → A), we need to construct two subderivations and in each derive its conclusion. Once the derivations are set up, the task is clear. Remember that while you are allowed to repeat information from the mother to each daughter. You are not allowed to repeat information between the sister derivations – the derivation of A from B and from Premise 1 needs to proceed independently from the derivation of B from A and Premise 1. Complete the derivation and check that you have done so correctly with the *Solutions.*

```
1. | (A → B) • (B → A)      Pr.
   |------------------
2. |  | A                  Assp (≡Int)
   |  |------------------
   |  |
   |  | B
   |
   |  | B                  Assp (≡Int)
   |  |------------------
   |  |
   |  | A
   | A ≡ B
```

Example 2.

Prove that (A • C) ≡ (B • D) follows from two premises: A ≡ B and C ≡ D. This proof is again not difficult as long as we construct the subderivations properly:

1.	A ≡ B	Pr.
2.	C ≡ D	Pr.
3.	A • C	Assp (≡Int)
	B • D	
	B • D	Assp (≡Int)
	A • C	
	(A • C) ≡ (B • D)	

Complete the proof and check the results with the *Solutions*.

Exercises on Proofs Using ≡Int

≡Int.II. Construct the following proofs:

(a) Prove that: C ≡ D

1.	A ∨ C	Pr.
2.	D • ~A	Pr.

(b) Prove that: ~A ≡ ~C

1.	~A ∨ C	Pr.
2.	A ∨ ~C	Pr.

(c) Prove that: A ≡ C

1.	A ≡ B	Pr.
2.	B ≡ C	Pr.

(d) Prove that: D

1.	(A ≡ A) → B	Pr.
2.	B ≡ D	Pr.

2. The Disjunction Elimination Rule (∨Elim)

Given a disjunction in an earlier line of a derivation and given that you can derive a certain statement *r* from each of the disjuncts (in two sister subderivations), you are allowed to introduce the statement *r* to the derivation.

i.	$p \vee q$	
j.	│ p	Assp (∨Elim)
k.	│ r	
l.	│ q	Assp (∨Elim)
m.	│ r	
➢	r	∨Elim *i*, *j*–*k*, *l*–*m*

Intuitions

Fill in the conclusion in the schema below:

Annie will either eat the pecan pie or the apple pie.

  │ Suppose that Annie eats the pecan pie.
  │ Annie will destroy her diet.

  │ Suppose that Annie eats the apple pie.
  │ Annie will destroy her diet.

Constructing subderivations for ∨Elim

The ∨Elim rule allows us to use information from a disjunction. Once we know that the need to use a disjunction in a derivation, the rule tells us to construct two sister subderivations and it tells us exactly what the assumptions of those subderivations must be. However, unlike the previous structural rules we have come to know (→Int and ≡Int), the ∨Elim rule does not specify what conclusion we will be deriving save for requiring that it be the same conclusion. What conclusion we will be deriving in each subderivation will be determined by the context of the actual proof we will be doing – by what we will want to derive in each specific proof.

Exercises on Applying ∨Elim

∨Elim.I.a. In all of the following proof schemata, you are asked to apply the ∨Elim rule to derive a certain statement. Fill in the information missing in step 9. (Note that the point of the exercise is not to actually construct the whole proof!)

(a)

2	B ∨ A	
3		B	Assp (∨Elim)
5		C	
6		A	Assp (∨Elim)
8		C	
9		∨Elim 2, 3–5, 6–8

(b)

2	~C ∨ D	
3		~C	Assp (∨Elim)
5		A	
6		D	Assp (∨Elim)
8		A	
9		∨Elim 2, 3–5, 6–8

(c)

2	(D → A) ∨ (A ∨ C)	
3		D → A	Assp (∨Elim)
5		~B ≡ A	
6		A ∨ C	Assp (∨Elim)
8		~B ≡ A	
9		∨Elim 2, 3–5, 6–8

(d)

2	~A ∨ ~B	
3		~A	Assp (∨Elim)
5		B → A	
6		~B	Assp (∨Elim)
8		B → A	
9		∨Elim 2, 3–5, 6–8

∨Elim.I.b. In all of the following proof schemata, you are asked to apply the ∨Elim rule to derive a certain statement (line 9). To do that we need construct two subderivations. Your task is to fill in the assumptions and the conclusions of the subderivations (As before, the point of the exercise is not to actually construct the whole proof!)

(a)

2	D ∨ B	
3		Assp (∨Elim)
5		
6		Assp (∨Elim)
8		
9	C	∨Elim 2, 3–5, 6–8

(b)

2	A ∨ B	
3		Assp (∨Elim)
5		
6		Assp (∨Elim)
8		
9	C ∨ A	∨Elim 2, 3–5, 6–8

(c)

2	C ∨ (A ∨ B)	
3		Assp (∨Elim)
5		
6		Assp (∨Elim)
8		
9	~D ∨ ~B	∨Elim 2, 3–5, 6–8

(d)

2	(A → B) ∨ (B → A)	
3		Assp (∨Elim)
5		
6		Assp (∨Elim)
8		
9	A ∨ B	∨Elim 2, 3–5, 6–8

∨Elim.I.c. In all of the following proof schemata, you are asked to apply the ∨Elim rule to derive a certain statement (line 9). To do that we need construct two subderivations. Your task is to fill in the assumptions and the conclusions of the subderivations (As before, the point of the exercise is not to actually construct the whole proof!)

(a)

2		
3	A	Assp (∨Elim)
5	C	
6	~A	Assp (∨Elim)
8	C	
9		∨Elim 2, 3–5, 6–8

(b)

2		
3	~B	Assp (∨Elim)
5	~~C	
6	~~D	Assp (∨Elim)
8	~~C	
9		∨Elim 2, 3–5, 6–8

(c)

2		
3	A ∨ B	Assp (∨Elim)
5	~B	
6	B ≡ C	Assp (∨Elim)
8	~B	
9		∨Elim 2, 3–5, 6–8

(d)

2		
3	A • B	Assp (∨Elim)
5	B • A	
6	B • A	Assp (∨Elim)
8	B • A	
9		∨Elim 2, 3–5, 6–8

Examples of Proofs Using ∨Elim

Example 3.

Prove that the following reasoning is valid: Either Ann or Betty will go to the cinema. Ann will go to the cinema just in case Cecilia and Danny will both go to the cinema. If Betty goes to the cinema then Danny and George will both go to the cinema. So, Danny will go to the cinema for sure. In symbols (where the first letters of the names abbreviate appropriate simple statements):

$$\begin{array}{c} A \vee B \\ A \equiv (C \bullet D) \\ B \rightarrow (D \bullet G) \\ \hline D \end{array}$$

Such premises might seem hopeless. We could derive D from premise 2 but we would have to have A, which we do not have. We could derive D from premise 3 but we would have to have B, which we do not have. Fortunately, we know that either A or B is true (premise 1). We just need to use this information appropriately. The rule that allows us to use information from a disjunction is the ∨Elim rule. The rule tells us to construct two sister subderivations and it tells us that the assumptions of these subderivations must be A and B, respectively. The rule tells us that as long as we derive the same statement in both subderivations, we will be able to derive that statement in the main derivation. The statement that we ultimately want to derive is D, so let's fill in D as the conclusion of the subderivations. Complete the derivation and check with the *Solutions.*

```
1. | A ∨ B              Pr.
2. | A ≡ (C • D)        Pr.
3. | B → (D • G)        Pr.
   |-----------
4. |  | A               Assp (∨Elim)
   |  |--------
   |  |
   |  | D
   |
   |  | B               Assp (∨Elim)
   |  |--------
   |  |
   |  | D
   | D
```

Exercises on Proofs Using ∨Elim

∨Elim.II. Construct the following proofs:

(a) Prove that: B

1.	~A ∨ B	Pr.
2.	~A → B	Pr.

(b) Prove that: C

1.	A ∨ B	Pr.
2.	(A → C) • (B → C)	Pr.

(c) Prove that: $D \vee G$

1.	$A \vee B$	Pr.
2.	$(A \vee C) \rightarrow D$	Pr.
3.	$G \equiv (\sim A \vee B)$	Pr.

(d) Prove that: $H \bullet B$

1.	$A \bullet B$	Pr.
2.	$A \rightarrow (G \vee H)$	Pr.
3.	$G \equiv H$	Pr.

(e) Prove that: G • H

1.	A • B	Pr.
2.	$A \rightarrow (G \vee H)$	Pr.
3.	$G \equiv H$	Pr.

(f) Prove that: D

1.	$(B \vee A) \rightarrow D$	Pr.
2.	$A \vee (B \vee C)$	Pr.
3.	$C \equiv D$	Pr.

3. *Reductio ad absurdum* arguments: ~Int and ~Elim

Both remaining rules ~Int and ~Elim are based on reductio ad absurdum arguments. This is a well-known form of reasoning, in which one assumes the opposite of what one is trying to prove and shows that this assumption leads to absurdity (contradiction). If one indeed manages to show that the assumption leads to a contradiction then one is justified in concluding that the assumption is false.

The form of reasoning was systematically used by Parmenides in his theory of being. The theory was based on two axioms, which Parmenides thought to be self-evident:

(A_1) Being (i.e. what is) is
(A_2) Nonbeing (i.e. what is not) is not

From those axioms, Parmenides derived four theorems:

(1) Being is ungenerable (does not have a beginning)
(2) Being is imperishable (does not have an end)
(3) Being is continuous (does not have holes)
(4) Being is unchangeable

He used *reductio* arguments to prove each theorem. Here is how he established the first: Suppose the contrary of what I want to prove: suppose that being has a beginning. If so, then there would be some time when the being begins, let's call this time t_0. We should ask: What *was* prior to time t_0? The only available answer is: there had to be nonbeing. But we know from axiom (A_2) that nonbeing cannot be. This means that the supposition that being has a beginning leads to a contradiction, for it leads us to conclude that nonbeing would be (prior to t_0) while we know that nonbeing can not be (according to the truth of reason (A_2)). We must reject the supposition that caused the problem, viz. that being has a beginning, and conclude that being does not have a beginning, that being is ungenerable.

3.1. Negation Introduction Rule (~Int)

> If in a subderivation with *p* as an assumption, you manage to derive both a statement and its negation, then you may add the negation ~*p* to the derivation.
>
> | | | |
> |---|---|---|
> | *i.* | \| *p* | Assp (~Int) |
> | *j.* | \| *r* | |
> | *k.* | \| ~*r* | |
> | ➤ | ~*p* | ~Int *i*–*j*, *i*–*k* |

Intuitions

Fill in the missing steps in Parmenides' reasoning:

1.	Being has a beginning	(Pr.)
2.	There is a moment m_p at which being begins.	(2)
3.	Prior to m_p there is	(3)
4.	Non-being is not	(A_2)
5.		~Int (1-3, 1-4)

„A statement and its negation", or directly contradictory statements

When we talked about Parmenides' theory of being, we spoke of a contradiction or absurdity. As we shall see later, the pair of statements A ∨ ~A and B • ~B is also contradictory. The rule ~Int requires, however, that we have directly contradictory statements, i.e. a statement and its negation. Here are examples of such pairs:

A ∨ ~A	~(A ∨ ~A)
A • ~A	~(A • ~A)
A	~A
~A	~~A
A ∨ B	~(A ∨ B)

Statements *p* and *q* are **directly contradictory** just in case either *q* is the negation of *p* or *p* is the negation of *q*.

Constructing Subderivations for ~Int

The ~Int rule specifies clearly what the assumption of the subderivation must be. It does not specify, however, what the pair of contradictory statements is. Therein lies the difficulty in using the ~Int rule. The direct contradiction we seek might be related to the assumption but it may very well be completely unrelated to it.

Exercises on Applying ~Int

~Int.I.a. Complete the missing statements in the directly contradictory pairs, keeping the convention that statements in the second column (~*p*) are negations of the statements in the first column (*p*).

p	~*p*
A	
A ∨ B	
~(A • B)	
~A ∨ B	
~~A	
~B	

p	~*p*
	~(C • ~D)
	~(~A ≡ ~(A • B))
	~~(C → B)
	~(~A ≡ C)
	~~C
	~(~B • B)

~Int.I.b. In all of the following proof schemata, you are asked to apply the ~Int rule to derive a certain negation. Fill in the information missing in step 9. (Note that the point of the exercise is not to actually construct the whole proof!)

(a)

3	B	Assp (~Int)
5	~C	
8	C	
9		~Int 3–5, 3–8

(b)

3	A ∨ B	Assp (~Int)
5	~C	
8	~~C	
9		~Int 3–5, 3–8

(c)

3	~B	Assp (~Int)
5	~C ∨ D	
8	~(~C ∨ D)	
9		~Int 3–5, 3–8

(d)

3	~A • ~B	Assp (~Int)
5	~C	
8	~~C	
9		~Int 3–5, 3–8

~Int.I.c. In all of the following proof schemata, you are asked to apply the ~Int rule to derive a certain negation. Fill in the information missing in steps 3 and 5. (Note that the point of the exercise is not to actually construct the whole proof!)

(a)

3		Assp (~Int)
5		
8	A	
9	~C	~Int 3–5, 3–8

(b)

3		Assp (~Int)
5		
8	B	
9	~~A	~Int 3–5, 3–8

(c)

3		Assp (~Int)
5		
8	C • B	
9	~(A ∨ B)	~Int 3–5, 3–8

(d)

3		Assp (~Int)
5		
8	C	
9	~~(A → B)	~Int 3–5, 3–8

Examples of Proofs Using ~Int

Example 4.

Prove that the following reasoning is valid: "If Ann and Betty go to the cinema then so will Celia. Betty went to the cinema but Celia did not. So, Ann did not go to the cinema either". In symbols (using the same convention as before):

$$\begin{array}{c} (A \bullet B) \rightarrow C \\ B \bullet {\sim}C \\ \hline {\sim}A \end{array}$$

Note that we cannot use any elimination strategy to derive ~A. In order to derive ~A by elimination, ~A would have to be a component in what we have, but it is not. Let's try to think whether we cannot apply an introduction strategy to derive ~A – here we will be introducing a negation (~Int). In order to do that we need to construct an appropriate subderivation with A as assumption:

1.	$(A \bullet B) \rightarrow C$	Pr.
2.	$B \bullet {\sim}C$	Pr.
3.	│ A	Assp (~Int)
	│	
	│	
	~A	

Note that ~Int does not tell us what to derive except that what we need to derive is a direct contradiction. In such a case a good rule of thumb is to see whether we can find the beginnings of a contradiction already in the premises, i.e. if we can use an elimination strategy to derive a contradiction. This is indeed so in our case. The premises contain as components the statements C and ~C. If we could only get them to stand on their own lines in the subderivation, we would have all we need to complete our derivation. Complete the proof and check with the *Solutions* that you've done so correctly.

Exercises on Proofs Using ~Int

~Int.II. Construct the following proofs:

(a) Prove that: ~A

1.	A → B	Pr.
2.	A → ~B	Pr.

(b) Prove that: ~~C

1.	C • B	Pr.
2.	A	Pr.

(c) Prove that: ~(A • B)

1.	A → C	Pr.
2.	B → ~~D	Pr.
3.	~C • ~D	Pr.

(d) Prove that: ~(A ≡ B)

1.	(A ≡ B) → C	Pr.
2.	~(C ∨ A)	Pr.

3.2. The Negation Elimination Rule (~Elim)

If in a subderivation with ~*p* as an assumption, you manage to derive both a statement and its negation, then you may add the statement *p* to the derivation.

i.	| | ~*p*	Assp (~Elim)
j.	| | *r*	
k.	| | ~*r*	
➤	| *p*	~Elim *i*–*j*, *i*–*k*

Intuitions

Complete Parmendies' reasoning:

1.	Being is not continuous	(Pr.)
2.	There is a hole *h* in being.	(1)
3.	At any moment m_h of the duration of the hole *h*, there is	(2)
4.	Non-being is not.	(A_2)
5.		~Elim (1-3, 1-4)

Constructing subderivations for ~Elim

The ~Elim is very similar to ~Int. It specifies exactly what the assumption must be, it does not specify exactly what direct contradiction to derive. However, there is a sense in which it is easier to apply ~Int than ~Elim. If you compare the statements that the rules justify, you will see that the statement justified by ~Int will always be a negation (~*p*) whereas the statement justified by ~Elim can be anything (*p*). This means that there is no clear indication when to apply ~Elim. Moreover, because the rule requires us to venture onto an often difficult hunt for a direct contradiction, it should be used very sparingly. This is why the rule should be used only as the last resort – when everything else fails.

Ćwiczenia na stosowanie reguły ~Elim

~Elim.I.a. In all of the following proof schemata, you are asked to apply the ~Elim rule to derive a certain statement. Fill in the information missing in step 9. (Note that the point of the exercise is not to actually construct the whole proof!)

(a)

3	~A	Assp (~Elim)
5	~C	
8	C	
9		~Elim 3–5, 3–8

(b)

3	~~B	Assp (~Elim)
5	~C	
8	~~C	
9		~Elim 3–5, 3–8

(c)

3	~(A → B)	Assp (~Elim)
5	~C ∨ D	
8	~(~C ∨ D)	
9		~Elim 3–5, 3–8

(d)

3	~~(~A • ~B)	Assp (~Elim)
5	~C	
8	~~C	
9		~Elim 3–5, 3–8

~Elim.I.b. In all of the following proof schemata, you are asked to apply the ~Int rule to derive a certain negation. Fill in the information missing in steps 3 and 5. (Note that the point of the exercise is not to actually construct the whole proof!)

(a)

3		Assp (~Elim)
5		
8	A • B	
9	~C	~Elim 3–5, 3–8

(b)

3		Assp (~Elim)
5		
8	B ∨ C	
9	A	~Elim 3–5, 3–8

(c)

3		Assp (~Elim)
5		
8	~C • B	
9	~A ≡ B	~Elim 3–5, 3–8

(d)

3		Assp (~Elim)
5		
8	A → B	
9	A → B	~Elim 3–5, 3–8

Examples of Proofs Using ~Elim

Example 5.

Prove that A follows from two premises ~A → B and B → A. It might look like there is nothing simpler to do – we just need to use an elimination strategy. We can get A from the second premise (apply →Elim), but we would need B. We could get B from the first premise, but we would have to have ~A, which we do not, however. So the elimination strategy fails. Since we want to derive a simple proposition, we cannot use an introduction strategy. All that remains is ~Elim. Let's set up the subderivation:

1.	~A → B	Pr.
2.	B → A	Pr.
3.	~A	Assp (~Elim)
	A	

Now the task is to derive a direct contradiction. Again, let's review the premises – perhaps we can find a direct contradiction as components of the premises. Indeed we can: as long as we derive A, we will have a direct contradiction of A and ~A. Complete the proof and check that you have done so correctly with the *Solutions*. (Note that you should reiterate ~A in the subderivation at some point to derive the second term of the contradiction).

Examples on Proofs Using ~Elim

~Elim.II. Construct the following proofs:

(a) Prove that: A

1.	~~A • C	Pr.
2.	B • ~~D	Pr.

(b) Prove that: ~C

1.	~C ∨ (B ∨ ~C)	Pr.
2.	~B	Pr.

(c) Prove that: A • B

1.	~(A • B) → (C • D)	Pr.
2.	~D	Pr.

(d) Prove that: C

1.	~C → ~B	Pr.
2.	~A • (~B → B)	Pr.

(h) Prove that: ~(A ∨ B) [difficult]

1.	A → C	Pr.
2.	B → ~~D	Pr.
3.	~C • ~D	Pr.

What You Need to Know and Do

- You need to know the inference rules and be able to apply them
- You need to be able to construct proofs using the inference rules