Workbook Unit 12:

Natural Deduction Proofs (III)

Overview

This unit

- introduces the inference rules →Int
- introduces the concept of subderivations
- explains the constrains on using subderivations

Prerequisites

You need to have completed Units 10 and 11

drp@swps.edu.pl

1. The Conditional Introduction Rule (→Int)

If in a subderivation whose assumption is p, you can derive r then you are allowed to add a line to the derivation (i.e. to the «mother» derivation of the subderivation) with the conditional $p \to r$.

i.	p	Assp. (→Int)
j.	r	
➢	$p \to r$	→ Int i–j

1.1. Intuitions

Many inference rules in the SD system use the so-called subderivations. Subderivations are intuitively speaking derivations that you might think of as being carried out on a side to aid the main derivation.

In the case of the →Int rule, we can think of the subderivation as a kind of «think box», in which we *suppose* that the antecedent of the conditional we want to derive is true and in which we are in fact able to derive the consequent of the conditional.

Consider the following example argument:

(1) If Chris fails logic then he will not think much of himself and he will think that he studied too little.
(2) If Chris thinks that he studied too little, he will feel pangs of conscience.
(3) If Chris does not think much of himself, he will lose self-confidence.
(4) If in addition to losing self-confidence Chris will experience pangs of conscience, he will become depressed.
(5) If Chris becomes depressed, he will fail all his other courses.

So, if Chris fails logic then he will fail all his other courses.

If we want to show that the conclusion follows from the premises, we can reason on the supposition that Chris fails logic:

Suppose that (6) Chris fails logic.
From (1) and (6), it follows that (7) Chris will not think much of himself and that he will think that he studied too little.
From (7) and (2), it follows that (8) Chris will feel pangs of conscience.
From (7) and (3), it follows that (9) Chris will lose self-confidence.
From (8), (9) and (4), it follows that (10) Chris will become depressed.
From (10) and (5), it follows that (11) Chris will fail all his other courses.

So we have shown that on the assumption that Chris fails logic, he will fail all his other courses. In other words, we have shown the following conditional to be true:

If Chris fails logic then he will fail all his other courses.

In the SD system, there is a graphical way of organizing such additional proofs that we can carry out as if on the side. Subderivations contain such proofs. We can use this graphical method without yet using the inference rules of SD to organize this reasoning in the following way:

1.	If Chris fails logic then he will not think much of himself and he will think that he studied too little.	Pr.
2.	If Chris thinks that he studied too little, he will feel pangs of conscience.	Pr.
3.	If Chris does not think much of himself, he will lose self-confidence	Pr.
4.	If in addition to losing self-confidence Chris will experience pangs of conscience, he will become depressed.	Pr.
5.	If Chris becomes depressed, he will fail all his other courses.	Pr.
6.	Chris fails logic.	Assp.
7.	Chris will not think much of himself and that he will think that he studied too little.	(1), (6)
8.	Chris will feel pangs of conscience.	(7), (2)
9.	Chris will lose self-confidence.	(7), (3)
10.	Chris will become depressed.	(8), (9), (4)
11.	Chris will fail all his other courses.	(10), (5)
12.	If Chris fails logic then he will fail all his other courses.	(6)–(11)

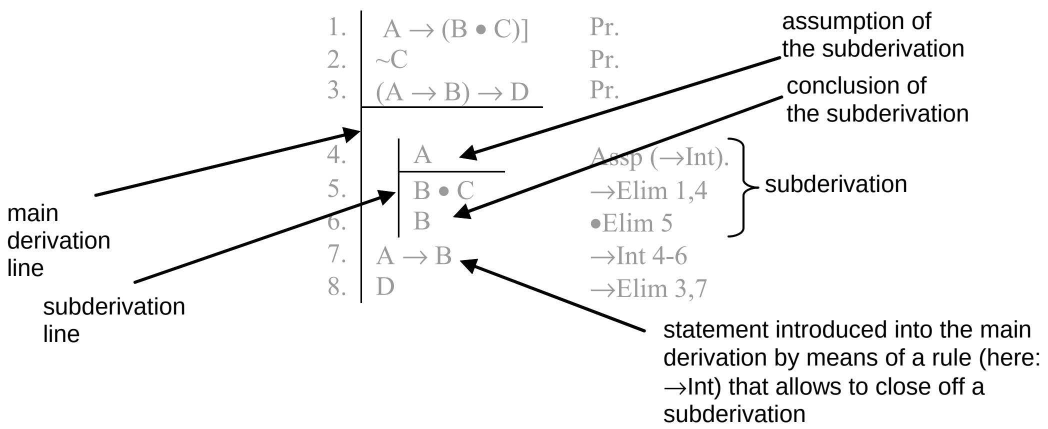
The subderivation is contained in lines 6-11. All statements in the subderivation are stand next the subderivation line, not next to the main derivation line. This is to mark the fact that the statements derived in the subderivation are derived under the assumption that «governs» the subderivation, in our case, under the assumption that Chris fails logic.

1.2. Subderivations

We will say more about subderivations soon. At this point, it is enough that you have a general idea what the function of subderivations is. You should also know that each subderivation:

- has its own derivation line
- has exactly *one* assumption, which is underlined by an assumption line
- can contain an unlimited number of steps
- the last step in the subderivation is called the conclusion of that subderivation

In addition, all the inference rules that require subderivations also tell us how to close subderivations and introduce a certain kind of statement into the mother derivation of that subderivation. You should not try to understand the following proof – just use it as a point of reference for the terminology we have introduced:



1.3. Constructing subderivations for →Int

The first thing that you need to learn is to construct the subderivations for the →Int rule. The →Int rule tells us

1. that we need to construct a subderivation
2. what the assumption of that subderivation is (always the antecedent of the conditional we want to derive), and
3. what conclusion we must derive in that subderivation (always the consequent of the conditional we want to derive).

Before you proceed any further, you must do the following exercises and check with the *Solutions* that your answers are correct:

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 conditional. Fill in the information missing in step 9. (Note that the point of the exercise is not to actually construct the whole proof!)

		Pr.
3	B	Assp. (→Int)
8	A	
9		→ Int 3–8

		Pr.
3	~B	Assp. (→Int)
8	A	
9		→ Int 3–8

		Pr.
3	C	Assp. (→Int)
8	~B	
9		→ Int 3–8

		Pr.
3	A ≡ C	Assp. (→Int)
8	B	
9		→ Int 3–8

		Pr.
3	C	Assp. (→Int)
8	B → A	
9		→ Int 3–8

		Pr.
3	C → A	Assp. (→Int)
8	B	
9		→ Int 3–8

		Pr.
3	~(A → B)	Assp. (→Int)
8	C	
9		→Int 3-8

		Pr.
3	~A	Assp. (→Int)
8	B → C	
9		→ Int 3–8

		Pr.
3	A ∨ ~A	Assp. (→Int)
8	B	
9		→ Int 3–8

		Pr.
3	B → C	Assp. (→Int)
8	C → D	
9		→ Int 3–8

→Int.I.b. In all of the following proof schemata, you are asked to apply the →Int rule to derive a conditional (line 9). To do that we need construct a subderivation, which has already been constructed. Your task is to fill in the assumption of the subderivation (step 3) and the conclusion that would have be derived in step 8. (As before, the point of the exercise is not to actually construct the whole proof!)

		Pr.
3		Assp. (→Int)
8		
9	A → B	→ Int 3–8

		Pr.
3		Assp. (→Int)
8		
9	~A → B	→ Int 3–8

		Pr.
3		Assp. (→Int)
8		
9	A → ~B	→ Int 3–8

		Pr.
3		Assp. (→Int)
8		
9	(A ∨ C) → B	→ Int 3–8

		Pr.
3		Assp. (→Int)
8		
9	A → (B → C)	→ Int 3–8

		Pr.
3		Assp. (→Int)
8		
9	(A → B) → C	→ Int 3–8

		Pr.
3		Assp. (→Int)
8		
9	(A ≡ ~B) → (~B ≡ A)	→ Int 3–8

		Pr.
3		Assp. (→Int)
8		
9	~(A→B)→(~A→B)	→ Int 3–8

1.4. Proofs Using →Int

The use of structural inference rules such as →Int really does change the dynamic of the proving process. For now, we could basically think on the side and our task was to find a strategy of applying the inference rules to get what we wanted. We will still do that but in addition we will need to be constructing subderivations. In order not to get confused when doing so, we will need to actually put the conclusion that we want to derive at the bottom of the proof (without line number or justification yet) and slowly work to justify the proof. This does not sound like much but it is one point where a lot of students find themselves at a loss when applying →Int. I will therefore guide you step by step in the first example. Try to actually draw the proof on a piece of paper and check that we are doing the same thing.

Example 1

Prove that A → C follows from A → (B • C)

Here is an instance of that reasoning „If Alice goes to the party then she will meet Ben and Chris' future wife. So, if Alice goes to the party then she will meet Chris' future wife).

```
1. | A → (B • C)      Pr.          Prove:  A → C
   |______________
   |
   |
   |
```

In this case, you will need to apply the →Int rule – your conclusion is a conditional and you only have a conditional to work with. As I've just said, in the case of structural rules, the first thing that you should do, is to write the statement you want to derive at the bottom of your proof. How much space should you leave? "Ample" is the answer. There is no ready-made rule for that. With practice, you will kind of figure out how much. It is certainly always better to leave more space than less. So we need to rewrite our proof like that:

```
1. | A → (B • C)      Pr.
   |______________
   |
   |
   |
   | A → C
```

Note! the fact that we have written the statement A → C at the bottom of the derivation line does not mean that we have derived it. We will not have derived it until we can justify it. We still cannot do that. The statement is there just to structure our proof – to remind us what we need to derive.

The conclusion we want to derive (A → C) is a conditional. We cannot derive it from anywhere above (there is no rule that we could use to derive it from the only premise we have, viz. premise 1), so we will need to apply a conditional introduction rule to get it. Now, we know that in order to apply the →Int rule, the first thing that we need to do is to construct a subderivation. Let's draw one in (note that the subderivation we draw must begin right below premise 1 and end right above the conditional that we want to derive:

1.	A → (B • C)	Pr.
	|	
	|	
	A → C	

The →Int tells us exactly what must be the assumption of the subderivation and what must be the conclusion that we will have to derive in the subderivation. Let us fill in this information. The assumption of the subderivation will be the antecedent of the conditional that we want to derive, i.e. the antecedent of A → C, in our case: A. The →Int tells us also that we can add this assumption to our proof, number the step and justify it as "Assp. (→Int)".

The last thing that we need to do is to fill in the conclusion that we will have to derive in the subderivation. The conclusion of the subderivation required by →Int will always be the consequent of the conditional that we want to derive, i.e. the consequent of A → C, in our case: C.

1.	A → (B • C)	Pr.
2.	| A	Assp. (→Int)
	| C	
	A → C	

Note that unlike in the case of the assumption of the subderivation, the →Int rule does not tell us how to justify the conclusion of the subderivation. It only tells us what we will have to derive in the subderivation in order to apply →Int further.

It is at this point that students make a very common error. In the setting up of the subderivation your reference point is the conditional that you want to derive, i.e. A → C, not the conditional that are given, i.e. A → (B • C). If you set up the subderivation with the conditional that you are given (here A → (B • C)) in mind, you will only be able to prove that conditional again, which is not your task.

1.	A → (B • C)	Pr.
2.	| A	Assp. (→Int)
	| B • C	
	A → C	

Let’s look our set-up again.

1.	A → (B • C)	Pr.
2.	A	Assp. (→Int)
	C	
	A → C	

We are ultimately supposed to prove A → C. However, after we set up the proof for →Int, our task as it were got simplified. We now have to prove that C using two pieces of information: premise 1, which is A → (B • C), and the assumption that the rule allowed us to add in line 2, i.e. A. But surely that task now is trivial.

Since we have the conditional A → (B • C) in line 1 and the matching antecedent A in line 2, we can derive the consequent by means of →Elim:

1.	A → (B • C)	Pr.
2.	A	Assp. (→Int)
3.	B • C	→Elim 1, 2
	C	
	A → C	

Given that we derived the conjunction B • C, we can derive C by means of •Elim (note that you will be just adding the line number and the justification to the line where you have written C already in the set-up for the →Int rule):

1.	A → (B • C)	Pr.
2.	A	Assp. (→Int)
3.	B • C	→Elim 1, 2
4.	C	•Elim 3
	A → C	

Now we managed to derive C in the subderivation. In other words, we have managed to do what the →Int rule required of us: we have constructed a subderivation with the antecedent of A → C as an assumption, we have managed to derive in that subderivation the consequent of A → C, so the rule →Int allows us now to *close off* the subderivation and introduce the conditional A → C into the main derivation.

1.	A → (B • C)	Pr.
2.	A	Assp. (→Int)
3.	B • C	→Elim 1, 2
4.	C	•Elim 3
5.	A → C	→Int 2-4

Note that the →Int rule references the whole subderivation, i.e. lines 2 through 4.

Example 2.

Prove that (A • C) → ~D follows from A → (B • ~D):

```
1. | A → (B • ~D)      Pr.
   |----------------
   |
   |
   |
   | (A • C) → ~D
```

The natural reaction to this proof is confusion: "Where did the C come from?". Before we get confused any further, let's do what we have been told. The conclusion we want is a conditional, it does not look like we can derive it by any elimination rules from the premises (here: one premise), so the only strategy we can pursue is by using an introduction rule (here: →Int). The →Int rule tells us to set up a subderivation with the antecedent of the conditional we want to derive as an assumption and the consequent of that conditional as the conclusion, which we will have to derive:

```
1. | A → (B • ~D)      Pr.
   |----------------
2. |  | A • C          Assp. (→Int)
   |  |-------------
   |  |
   |  | ~D
   | (A • C) → ~D
```

Now our task is much simpler. We need to derive the conclusion ~D from two statements: the conditional A → (B • ~D) and the conjunction A • C. We reason as usual. ~D is a conjunct of the conjunction B • ~D, which is hidden in the consequent of the conditional in line 1. We will need to get the conjunction to stand on its own. We can do that by applying →Elim rule to the conditional in line 1 – but we will need to match its antecedent. The antecedent of the conditional in line 1 is A, which we do not have on its own line but we can derive from line 2 by means of •Elim.

Complete the proof and check that you have done so correctly with the *Solutions.*

Example 3.

Prove that from the premise A → (B → C) we can derive the conclusion (A • B) → C

```
1. | A → (B → C)      Pr.
   |-----------------
   |  |
   |  |--------------
   |  |
   |  |
   |  |
   |  |
   | (A • B) → C
```

Our task is to derive the conditional (A • B) → C. We will need to apply the →Int rule, which means that we will have to set up an appropriate subderivation. Try to fill it in in the proof above and complete the proof. Then read the commentary and check that your derivation looks just like the one in the *Solutions*.

Remembering that it is the conditional that we want to derive that provides us with the information how to set up the derivation, we can insert A • B and justify it as an assumption of the subderivation and we know that we will have to derive C as the conclusion of the subderivation. Now our task is to derive C from two statements A → (B → C) and A • B. The wanted conclusion C is hidden within a consequent of a conditional that is itself the consequent of the conditional in line 1. We know a rule that allows us to derive the consequent of a conditional (→Elim) as long as we have the antecedent of that conditional. So given that we have A → (B → C) we will be able to derive its consequent if we can match the antecedent, if we have A on its own line. We do not but we can derive it by •Elim from the assumption of the subderivation. The application of →Elim will allow us to derive B → C. We can then derive C by →Elim if we have B. We can easily get B by •Elim from the assumption of the subderivation.

Exercises on Proofs with →Int

→Int.II. Complete the following proofs. First, you have to complete the preparation of the subderivation filling in the assumption in line 3 and the wanted conclusion in line 5. You should then fill in the missing step in line 4 and justify all steps:

1.	A → (B ∨ C)	Pr.
2.	~B	Pr.
3.		Assp. (→Int)
4.		
5.		
6.	A → C	

1.	A → (B ≡ ~D)	Pr.
2.	~D	Pr.
3.		Assp. (→Int)
4.		
5.		
6.	A → B	

1.	~A → (C → B)	Pr.
2.	C	Pr.
3.		Assp. (→Int)
4.		
5.		
6.	~A → B	

1.	B ∨ (B ∨ C)	Pr.
2.	~B	Pr.
3.		Assp. (→Int)
4.		
5.		
6.	A → C	

1.	A ≡ (B ∨ C)	Pr.
2.	~D	Pr.
3.		Assp. (→Int)
4.		
5.		
6.	(A ≡ ~D) → (B ∨ C)	

1.	(A ≡ B) ∨ (B ∨ C)	Pr.
2.	~(A ≡ B)	Pr.
3.		Assp. (→Int)
4.		
5.		
6.	~B → C	

2. Nested Subderivations

Each subderivation is a derivation, so there is no reason why subderivations could not contain further subderivations. Before we look at proofs that use subderivations of subderivations, let's introduce some terminology that will help us think about the nested subderivations. We will use a familial metaphor to help us navigate these concepts.

We have already said that we can think of the derivation D that contains a subderivation S as the «mother» derivation with respect to S. The subderivation could be thought of as the «daughter» of D. If the subderivation S contains another subderivation S′, S′ is the «daughter» of S and the «granddaughter» of D, so D is the «grandmother» of S′. Subderivations can also have sisters. If two subderivations have the same mother then they are sister derivations. These relations are illustrated in Figure 1.

We also have to introduce the notion of **mother-line** of a derivation S, which is a set of all derivations that are the mother of S, the grandmother of S, the greatgrandmother of S, the greatgreatgrandmother of S and so on. It should be noted the mother-line of S does not include the sisters of S or the sisters of the mother of S and so on. The **daughter-line** of a derivation S is in turn a set of all derivations that are the daughters of S, the granddaughters of S, the greatgranddaughters of S, and so on. The **sister-line** of a derivation S is in turn a set of all derivations that are the sisters of S and their daughter-lines.

Note that the notion of a "mother derivation" is a relative notion. A derivation is a mother derivation always with respect to *some* derivation. If a derivation does not have a mother derivation then it is the main derivation.

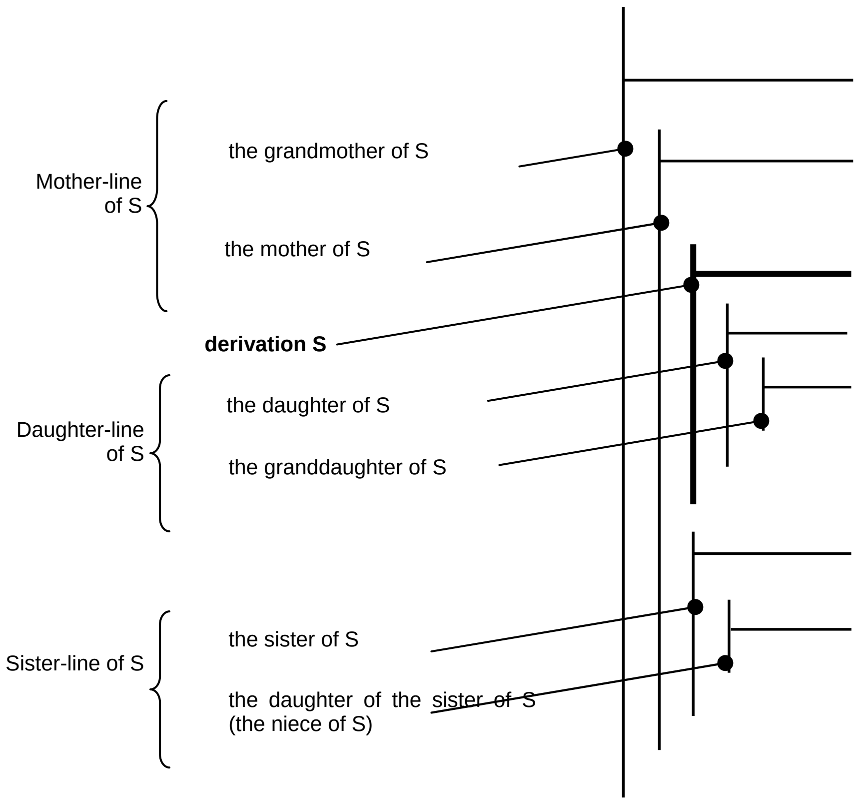


Figure 1.

Definitions

These concepts can be defined more precisely. Let the notion of a subderivation of a given derivation be a primitive notion.

Derivation *D* jest the **mother derivation** of derivation *d* iff *d* is a subderivation of *D*.

Derivation *D′* belongs to the **mother-line** of derivation *D* iff:

(0) *D′* is the mother derivation of *D*

(1) *D′* is the mother derivation of derivation D_x, where D_x belongs to the mother-line of *D*.

Derivation *D′* belongs to the **daughter-line** of derivation *D* iff:

(0) *D′* is a subderivation of *D*

(1) *D′* is the subderivation of derivation D_x, where D_x belongs to the daughter-line of *D*

Derivation *D′* belongs to the **sister-line** of derivation *D* iff:

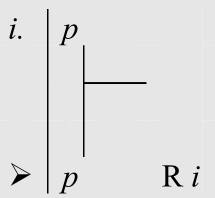
(0) there is a derivation D_m, which is the mother derivation of both *D′* and *D*

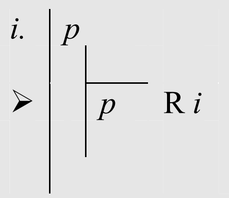
(1) there is a derivation Dm, which is the mother derivation of both D_x and *D*, where D_x belongs to the mother-line of *D′*.

3. Reiteration Rule (R)

The Reiteration Rule is rarely used explicitly. Most of the time, it is used implicitly – it basically tells us what information can be used in subderivations.

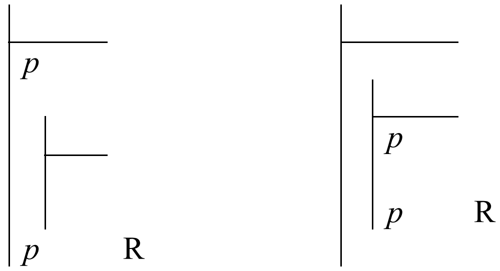
If *p* occurs in an earlier line of a given derivation or of some derivation belonging to the mother-line of a given derivation then you may add a line to the derivation with *p* (standing on its own).



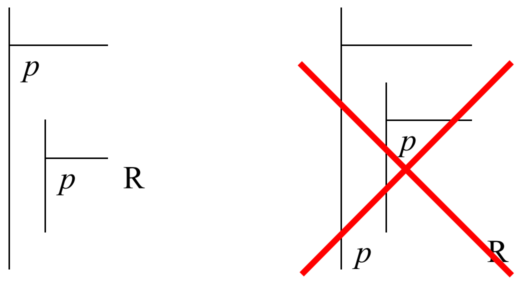


The Reiteration rule governs the transfer of information between derivations and subderivations. To help us apply the rule, we can think of it as composed of three principles:

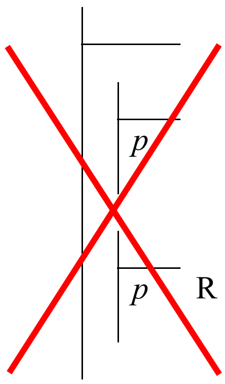
Principle 1: ***Everybody knows her own secrets***. – You can repeat any statement within a given derivation.



Principle 2: ***Mothers convey their secret to daughters*** (who convey their secrets to their own daughters, etc.), however ***daughters never tell their secrets to their mothers,*** grandmothers, etc.



Principle 3: ***Sisters never share any secrets with each other***. This is because sister derivations may have different additional assumptions (different «fathers»).



Let's consider two derivation schemata (a) and (b) in Fig. 2. Which of the lines is justified by the Reiteration Rule?

(a) (b)

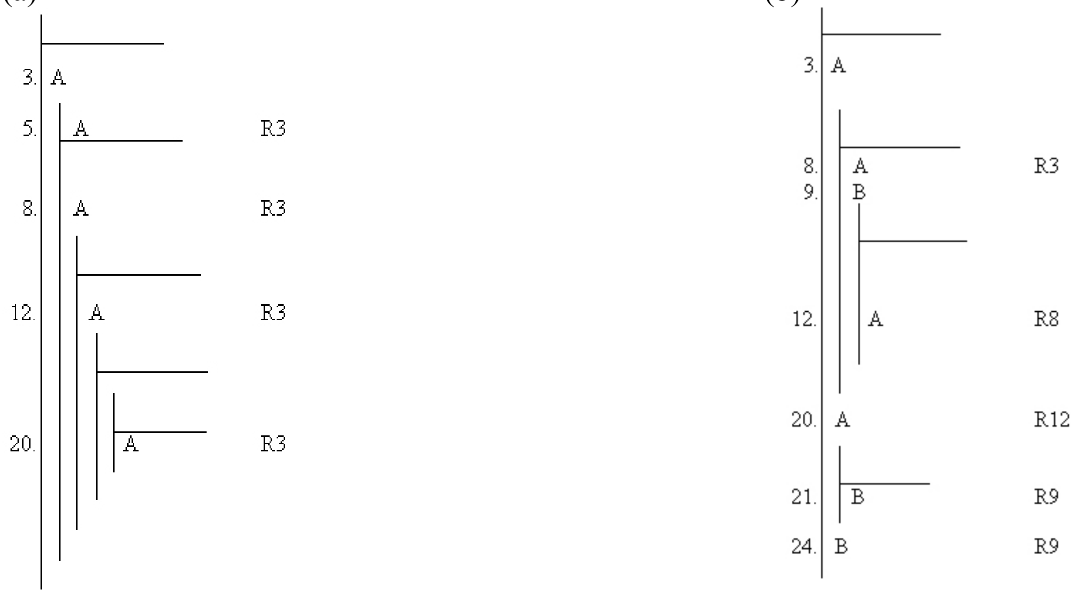


Fig. 2.

In the derivation schema (a), the only incorrect step is in line 5 since the Reiteration Rule may not be used to justify an assumption of a subderivation.

In the derivation schema (b), there are more incorrect applications of R:

In line 20: R is applied to A standing in a subderivation of a subderivation, but granddaughters do not tell secrets to their grandmothers (Principle 2). Of course, if the justification of line 20 read "R3", the step would be justified.

In line 21: B is repeated from line 9, which is a line in a sister derivation, and sisters don't tell each other secrets (Principle 3).

In line 24: B is repeated from line 9, which is a line in a subderivation, and daughters don't share secrets with their mothers (Principle 2).

Using Rule R implicitly and explicitly

Rule R can be used explicitly but it is often quite tedious. The rule is therefore usually applied implicitly, which means that we use information from those lines in a proof that are accessible to rule R, i.e. which could be repeated and justified explicitly by rule R. Consider two versions of the same proof. In the first version, rule R is used implicitly, while it is used explicitly in the second version:

1.	$(D \vee A) \rightarrow (C \rightarrow B)$	Pr.
2.	$C \bullet \sim A$	Pr.
3.	$\quad D$	Pr.
4.	$\quad D \vee A$	$\vee$Int 3
5.	$\quad C \rightarrow B$	$\rightarrow$Elim 1,4
6.	$\quad C$	$\bullet$Elim 2
7.	$\quad B$	$\rightarrow$Elim 5,6
8.	$D \rightarrow B$	$\rightarrow$ Int 3–7

1.	$(D \vee A) \rightarrow (C \rightarrow B)$	Pr.
2.	$C \bullet \sim A$	Pr.
3.	$\quad D$	Pr.
4.	$\quad D \vee A$	$\vee$Int 3
5.	$\quad (D \vee A) \rightarrow (C \rightarrow B)$	R1
6.	$\quad C \rightarrow B$	$\rightarrow$Elim 4, 5
7.	$\quad C \bullet \sim A$	R2
8.	$\quad C$	$\bullet$Elim 7
9.	$\quad B$	$\rightarrow$Elim 6,8
10.	$D \rightarrow B$	$\rightarrow$ Int 3–9

Both proofs are correct. We will adopt the convention of using rule R implicitly most of the time.

There are some proofs, however, where rule R must be used explicitly, which is illustrated by the following example:

Example 4.

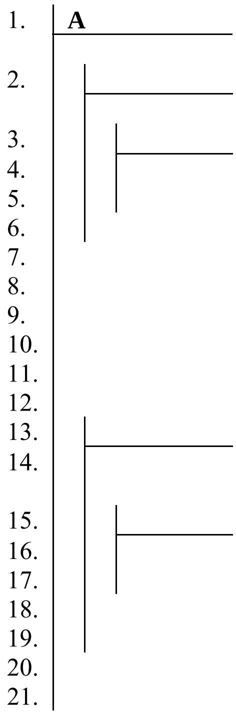
Prove that $C \rightarrow A$ follows from A. To derive a conditional, we need to construct a subderivation for $\rightarrow$Int, with the antecedent of the wanted $C \rightarrow A$ (i.e. C) as the assumption and its consequent (i.e. A) as the conclusion of the suderivation:

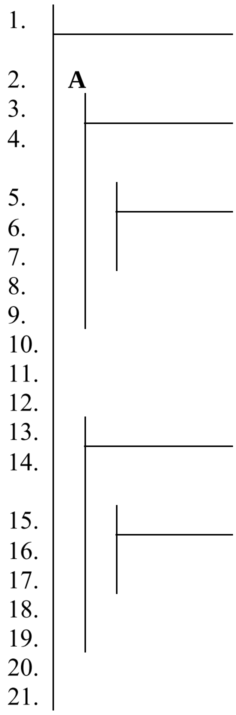
1.	A	Pr.
2.	$\quad C$	Assp. ($\rightarrow$Int)
	$\quad A$	
	$C \rightarrow A$	

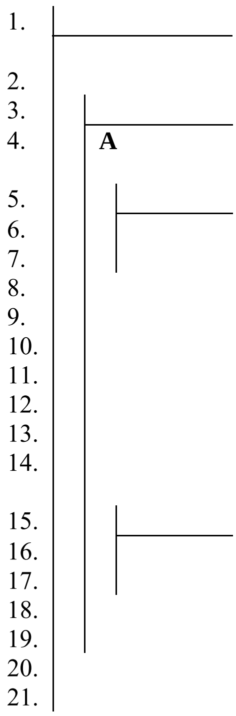
How can we derive A from the premise (A in line 1) and the additional assumption (C in line 2)? Simply by applying rule R – but this time we must do so explicitly lest the conclusion of the derivation be unjustified. (See *Solutions.*)

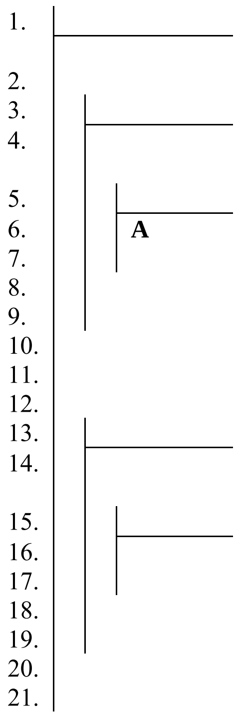
Exercises on Applying Rule R

Ex. R.I. In each of the following proof schemata, show all the lines where you can repeat (using rule R) the indicated statement A. (Remember that assumptions of subderivations are always justified as assumptions. Assume also that each of the subderivations is closed by →Int rule in the line that immediately follows the subderivation.)









Exercises on Proofs Using Rule R

Ex. R.II. Construct the following proofs using rule R *explicitly*:

Prove: $C \rightarrow (A \bullet B)$

1.	$A \bullet B$	Pr.

Prove: $A \bullet A$

1.	A	Pr.

Prove: $C \rightarrow A$

1.	$A \equiv B$	Pr.
2.	$B \equiv C$	Pr.

Prove: $[B \bullet (D \bullet G)] \rightarrow (A \bullet C)$

1.	$A \equiv B$	Pr.
2.	$C \equiv D$	Pr.

4. Examples of Proofs with Nested Subderivations

Example 5.

Prove that the conditional A → (B → C) follows from the premise (A • B) → C

1.	(A • B) → C	Pr.	Prove: A → (B → C)

We have already seen that the reverse implication holds (Example 3). We begin as always by constructing a subderivation whose additional assumption is the antecedent of the conditional we want to derive, and the conclusion – the consequent of the conditional we want to derive.

1.	(A • B) → C	Pr.
2.	A	Assp. (→Int)
	B → C	
	A → (B → C)	

How can we derive B → C from (A • B) → C and A? It is not really clear. So, it might be useful to repeat our reasoning. We want to derive a conditional, it might be worthwhile to apply the →Int rule. It requires us to construct a subderivation, so we must construct yet another subderivation. In assigning the assumption and the conclusion of this subderivation, we need to pay attention to the conditional we now want to derive (i.e. the conditional B → C). Its antecedent (B) will be the assumption of the second (nested) subderivation, while its consequent (C) will be the conclusion of that derivation. We thus have the following structure:

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

Now – despite the temporary vertigo that you may experience – the task is much simpler. We need to derive C from three statements: (A • B) → C, A and B. But that is now easy. We can derive C by →Elim as long as we have A • B, which we can in turn easily get using the •Int rule.

```
1. | (A • B) → C          Pr.
   |-----------
2. | | A                  Assp. (→Int)
   | |----------
3. | | | B                Assp. (→Int)
   | | |--------
4. | | | A • B            •Int 2,3
5. | | | C                →Elim 1,4
6. | | B → C
7. | A → (B → C)
```

We have thus obtained and justified statement C, which means that we can close the "granddaughter" derivation (of the main derivation) and add the conditional B → C to her mother derivation (the daughter derivation of the main derivation):

```
1. | (A • B) → C          Pr.
   |-----------
2. | | A                  Assp. (→Int)
   | |----------
3. | | | B                Assp. (→Int)
   | | |--------
4. | | | A • B            •Int 2,3
5. | | | C                →Elim 1,4
6. | | B → C              →Int 3–5
7. | A → (B → C)
```

We have thus obtained and justified statement B → C, which means that we can close the "daughter" derivation (of the main derivation) and add the conditional A → (B → C) to her mother derivation (i.e. to the main derivation):

```
1. | (A • B) → C          Pr.
   |-----------
2. | | A                  Assp. (→Int)
   | |----------
3. | | | B                Assp. (→Int)
   | | |--------
4. | | | A • B            •Int 2,3
5. | | | C                →Elim 1,4
6. | | B → C              →Int 3–5
7. | A → (B → C)          →Int 2–6
```

Exercises on Applying →Int

Ex. →Int.I.c. In all of the following proof schemata, you are asked to apply the →Int rule twice to derive a certain conditional whose consequent is also a conditional. Fill in the information missing in steps 8 and 9. (Note that the point of the exercise is not to actually construct the whole proof!)

		Pr.
3.	A	Assp. (→Int)
4.	B	Assp. (→Int)
7.	C	
8.		→ Int 4–7
9.		→ Int 3–8

		Pr.
3.	~A	Assp. (→Int)
4.	B	Assp. (→Int)
7.	~C	
8.		→ Int 4–7
9.		→ Int 3–8

		Pr.
3.	B ∨ C	Assp. (→Int)
4.	~B	Assp. (→Int)
7.	C	
8.		→ Int 4–7
9.		→ Int 3–8

		Pr.
3.	~B	Assp. (→Int)
4.	B ∨ C	Assp. (→Int)
7.	C	
8.		→ Int 4–7
9.		→ Int 3–8

		Pr.
3.	~B	Assp. (→Int)
4.	B	Assp. (→Int)
7.	B ∨ C	
8.		→ Int 4–7
9.		→ Int 3–8

		Pr.
3.	A → B	Assp. (→Int)
4.	D	Assp. (→Int)
7.	C	
8.		→ Int 4–7
9.		→ Int 3–8

Ex. →Int.I.d. In all of the following proof schemata, you are asked to apply the →Int rule twice to derive a conditional (line 9). To do that we need construct two nested subderivations, which have already been constructed. Your task is to fill in the assumptions of both subderivations (steps 3 and 4) and their respective conclusions (steps 7 and 8). (As before, the point of the exercise is not to actually construct the whole proof!)

		Pr.
3.		Assp. (→Int)
4.		Assp. (→Int)
7.		
8.		→ Int 4–7
9.	B → (A → C)	→ Int 3–8

		Pr.
3.		Assp. (→Int)
4.		Assp. (→Int)
7.		
8.		→ Int 4–7
9.	~C → (~B → ~A)	→ Int 3–8

		Pr.
3.		Assp. (→Int)
4.		Assp. (→Int)
7.		
8.		→ Int 4–7
9.	(B•C)→[A→(C•A)]	→ Int 3–8

		Pr.
3.		Assp. (→Int)
4.		Assp. (→Int)
7.		
8.		→ Int 4–7
9.	B → [A → (C → A)]	→ Int 3–8

		Pr.
3.		Assp. (→Int)
4.		Assp. (→Int)
7.		
8.		→ Int 4–7
9.	B → [(A → C) → C]	→ Int 3–8

		Pr.
3.		Assp. (→Int)
4.		Assp. (→Int)
7.		
8.		→ Int 4–7
9.	(B → C) → (A → C)	→ Int 3–8

Exercises on Proofs Using →Int

Ex. →Int.III. Complete the following proofs. First, you have to complete the preparation of the subderivations filling in the assumptions in lines 2 and 3 and the wanted conclusion in line 5 and 6. You should then fill in the missing step in line 4 and justify all steps:

1.	(A • B) → C	Pr.
2.		
3.		
4.		
5.		
6.		
7.	B → (A → C)	

1.	(A ∨ G) → C	Pr.
2.		
3.		
4.		
5.		
6.		
7.	A → (B → C)	

1.	(A → C) • G	Pr.
2.		
3.		
4.		
5.		
6.		
7.	A → (B → C)	

1.	A → (B → C)	Pr.
2.		
3.		
4.		
5.		
6.		
7.	B → (A → C)	

1.	G ≡ (A ∨ B)	Pr.
2.		
3.		
4.		
5.		
6.		
7.	G → (~A → B)	

1.	C ∨ (B ∨ A)	Pr.
2.		
3.		
4.		
5.		
6.		
7.	~C → (~B → A)	

5. Further Examples of Proofs

Example 6.

Prove that statement B follows from the following three premises. (Try to do the proof on your own, then read on to check that you've done it right):

1.	$A \to B$	Pr.	Prove: B
2.	$B \to C$	Pr.	
3.	$(A \to C) \to B$	Pr.	

Statement B occurs in all three premises. We know immediately that we will not be able to derive it from line 2 since B is the antecedent of a conditional. We don't have a rule that would allow us to derive the antecedent of a conditional (for a good reason too since you would not want to be able to infer *p* from $p \to r$; to convince yourself that you really would not, substitute 'I owe my logic teacher $1mln' for *p* and 'I should pay $1mln to my logic teacher' for *r*). So, we need to choose whether to get B from line 1 or from line 3.

Consider line 1. We could get B from line 1 if we had A, but we don't have A. Moreover, we have no hope of getting A because it is in the antecedent of a conditional, which is hidden in the antecedent of another conditional in the third premise.

Consider line 3. We could get B from line 3 if we had $A \to C$, but we do not have it. Can we get $A \to C$? Well, we can try applying the →Int rule:

1.	$A \to B$	Pr.
2.	$B \to C$	Pr.
3.	$(A \to C) \to B$	Pr.
4.	A	Assp. (→Int)
	C	
	$A \to C$	
	B	

Now, our immediate aim is to derive C in the just constructed subderivation, which given our additional assumption is indeed quite simple – we just need to apply the →Elim rule twice:

1.	$A \rightarrow B$	Pr.
2.	$B \rightarrow C$	Pr.
3.	$(A \rightarrow C) \rightarrow B$	Pr.
4.		A	Assp. (→Int)
5.		B	→Elim 1,4
6.		C	→Elim 2,5
	$A \rightarrow C$	
	B	

Since we have derived C from A, we can close off the subderivation and introduce the conditional A → C in the main derivation. This is what we needed to get B:

1.	$A \rightarrow B$	Pr.
2.	$B \rightarrow C$	Pr.
3.	$(A \rightarrow C) \rightarrow B$	Pr.
4.		A	Assp. (→Int)
5.		B	→Elim 1,4
6.		C	→Elim 2,5
7.	$A \rightarrow C$	→Int 4–6
8.	B	→Elim 3, 7

6. Proofs and Subderivations

Subderivations are "side" proofs and must always be closed off and the information contained in them must be introduced into the mother–derivation by means of an appropriate structural inference rule (such as →Int).

Consider once again the above proof (Example 6). In this proof, we are to derive B. But note that we do derive B already in step 5:

1.	$A \rightarrow B$	Pr.	Prove that B
2.	$B \rightarrow C$	Pr.	
3.	$(A \rightarrow C) \rightarrow B$	Pr.	
4.		A	Assp. (→Int)	
5.		B	→Elim 1,4	

~~The end~~

We cannot yet think that the proof is finished at this point. Our aim is to derive a given statement in the *main* derivation. This is not the same as deriving the statement in a subderivation – for when a statement such as B is derived within a subderivation, it might be that its derivation crucially depends on the assumption added to the proof. The statement we aim to derive must be derived in the main derivation. The above proof is completed in 8 steps.

→Int.IV. Construct the following proofs:

(a) Prove: B → (A → C)

1.	A → (B → C)	Pr.

(b) Prove: A → (B → C)

1.	B → (A → C)	Pr.

(c) Prove: C

1.	(A → B) ≡ C	Pr.
2.	B • D	Pr.

(d) Prove: ~A → [~B → (C • D)]

1.	C ∨ A	Pr.
2.	D ∨ B	Pr.

(e) Prove: C

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

(f) Prove: $A \rightarrow (B \bullet C)$

1. $A \rightarrow C$ Pr.
2. $A \rightarrow B$ Pr.

(g) Prove: $(B \rightarrow C) \rightarrow (A \rightarrow C)$

1. $A \rightarrow B$ Pr.

(h) Prove: $B \rightarrow (A \bullet C)$

1. $A \bullet (B \rightarrow C)$ Pr.

(i) Prove: $A \rightarrow (B \rightarrow C)$

1.	$C \bullet D$	Pr.

(j) Prove: $A \rightarrow (B \rightarrow (D \rightarrow C))$

1.	$C \bullet D$	Pr.

(k) Prove: $A \rightarrow (D \rightarrow C)$

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

(l) Prove: $(A \vee B) \rightarrow C$

1.	$(A \rightarrow C) \bullet {\sim}B$	Pr.
2.	$(B \vee C) \vee D$	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