

Chapter 3. Abbreviation

The language \mathcal{L} is rather austere. It will be convenient to supplement its alphabet with some additional symbols: \supset , \wedge , \equiv , \top and \perp , which we refer to, respectively, as the conditional sign, the conjunction sign, the biconditional sign, tee and eet. It will also be convenient to have some unsubscripted symbols to play the role of sentence letters.

There are at least three ways these new symbols might be regarded. First, they might be taken as purely metalinguistic devices for naming expressions of \mathcal{L} in an economical way. ' $(\mathbf{p}_1 \supset \mathbf{p}_2)$ ', for example, would be another name for the formula $(\neg \mathbf{p}_1 \vee \mathbf{p}_2)$. Second, they might be regarded as connectives and sentence letters of an independent object language that are, in many contexts, provably eliminable. On this view \supset and \vee would each have its own formal and informal interpretations, but $(\mathbf{p}_1 \supset \mathbf{p}_2)$ and $(\neg \mathbf{p}_1 \vee \mathbf{p}_2)$ would be shown to be truth-functionally equivalent. We adopt a third view. The new connectives are to be regarded as object language abbreviations. \supset and \vee are expressions of a single language \mathcal{L}^+ that contains \mathcal{L} . A binary relation of **abbreviating** or **standing for** is defined on the expressions of \mathcal{L}^+ , and expressions not contained in \mathcal{L} are regarded as inheriting many of their properties, including formulahood and truth under interpretation, from the expressions they abbreviate. $(\mathbf{p}_1 \supset \mathbf{p}_2)$, on this view, is not interpreted directly, but inherits the truth conditions of $(\neg \mathbf{p}_1 \vee \mathbf{p}_2)$. Details are presented below, with proofs of the central results left to the problems.

The alphabet of \mathcal{L}^+ is obtained by adding \supset , \wedge , \equiv , \top , \perp , \mathbf{p} , \mathbf{q} , \mathbf{r} , and \mathbf{s} to the alphabet of \mathcal{L} . Expressions of \mathcal{L}^+ that are not expressions of \mathcal{L} are regarded as putative abbreviations for expressions of \mathcal{L} . There are certain basic rules of abbreviation that tell us when one expression directly abbreviates another. Direct abbreviation, which we symbolize by $>$, is defined in the present case by the following conditions. In clauses (iv)-(vi), ' \mathbf{A} ' and ' \mathbf{B} ' range over the formulas of \mathcal{L} (though we will subsequently consider the effects of allowing the variables in those clauses to range over larger classes of expressions.)

definition 3.1

- (i) $\mathbf{p} > \mathbf{p}_1$; $\mathbf{q} > \mathbf{p}_2$; $\mathbf{r} > \mathbf{p}_3$; $\mathbf{s} > \mathbf{p}_4$;
- (ii) $\top > (\neg \mathbf{p}_1 \vee \mathbf{p}_1)$;
- (iii) $\perp > \neg \top$;
- (iv) $(\mathbf{A} \supset \mathbf{B}) > (\neg \mathbf{A} \vee \mathbf{B})$;
- (v) $(\mathbf{A} \wedge \mathbf{B}) > \neg(\mathbf{A} \supset \neg \mathbf{B})$;
- (vi) $(\mathbf{A} \equiv \mathbf{B}) > ((\mathbf{A} \supset \mathbf{B}) \wedge (\mathbf{B} \supset \mathbf{A}))$;

There are two general ways in which the basic rules of abbreviation may be extended. First, an expression may be taken to abbreviate not only on its own but also in the context of a larger expression. We therefore say

definition 3.1b An expression \mathbf{E} subdirectly abbreviates \mathbf{F} ($\mathbf{E} \triangleright \mathbf{F}$) if \mathbf{F} is the result of replacing a single occurrence of an expression in \mathbf{E} with the expression that it directly abbreviates.

Second, an abbreviation of an expression is also taken to abbreviate what that expression

abbreviates. So we say:

definition 3.1c **E abbreviates F** ($\mathbf{E} \gg \mathbf{F}$) if **F** is generated from **E** by the relation of subdirect abbreviation, i.e., if there is a sequence $\mathbf{E}_1, \dots, \mathbf{E}_n$, $n > 1$, of expressions beginning with **E** and ending with **F**, such that each expression subdirectly abbreviates its successor.

For example the following generation record shows that $(\mathbf{p} \equiv \top) \gg \neg(\neg(\neg\mathbf{p}_1 \vee (\neg\mathbf{p}_1 \vee \mathbf{p}_1)) \vee \neg(\neg(\neg\mathbf{p}_1 \vee \mathbf{p}_1) \vee \mathbf{p}_1))$.

1. $(\mathbf{p} \equiv \top)$
2. $(\mathbf{p}_1 \equiv \top)$ 1 clause i
3. $(\mathbf{p}_1 \equiv (\neg\mathbf{p}_1 \vee \mathbf{p}_1))$ 2 clause ii
4. $((\mathbf{p}_1 \supset (\neg\mathbf{p}_1 \vee \mathbf{p}_1)) \wedge ((\neg\mathbf{p}_1 \vee \mathbf{p}_1) \supset \mathbf{p}_1))$ 3 clause vi
5. $((\neg\mathbf{p}_1 \vee (\neg\mathbf{p}_1 \vee \mathbf{p}_1)) \wedge ((\neg\mathbf{p}_1 \vee \mathbf{p}_1) \supset \mathbf{p}_1))$ 4 clause iv
6. $((\neg\mathbf{p}_1 \vee (\neg\mathbf{p}_1 \vee \mathbf{p}_1)) \wedge (\neg(\neg\mathbf{p}_1 \vee \mathbf{p}_1) \vee \mathbf{p}_1))$ 5 clause iv
7. $\neg((\neg\mathbf{p}_1 \vee (\neg\mathbf{p}_1 \vee \mathbf{p}_1)) \supset \neg(\neg(\neg\mathbf{p}_1 \vee \mathbf{p}_1) \vee \mathbf{p}_1))$ 6 clause v
8. $\neg(\neg(\neg\mathbf{p}_1 \vee (\neg\mathbf{p}_1 \vee \mathbf{p}_1)) \vee \neg(\neg(\neg\mathbf{p}_1 \vee \mathbf{p}_1) \vee \mathbf{p}_1))$ 7 clause iv.

Abbreviation should not be confused with "simplification". The formula $\neg\neg\mathbf{A}$ simplifies to **A**, but neither expression abbreviates the other.

Notice that no expression of \mathcal{L} directly abbreviates anything, and that therefore no expression of \mathcal{L} abbreviates anything. An expression that abbreviates nothing will be called *primitive*. Expressions of \mathcal{L} are not the only primitive expressions-- \supset (, for example, is primitive because none of the clauses in 3.1 apply to it. That is why we called expressions proper to \mathcal{L}^+ "putative" abbreviations rather than abbreviations simpliciter.

The sequence of "disabbreviation" steps in the example above is not the only such sequence possible. The steps of lines 5 and 6, which eliminate two "parallel" occurrences of \supset from the formula on line 4, could have been performed in reverse order. The requirement that the variables in clauses (iv)-(vi) be formulas of \mathcal{L} , however, strongly constrains the order of disabbreviation. For example, the occurrence of \wedge on line 4 cannot be eliminated before the occurrence of \supset . This constraint on the order of disabbreviation facilitates the proof that abbreviation is "well-behaved" in the sense described in the following result.

Theorem 3.2

- a. \gg is a well-founded relation;
- b. Every expression of \mathcal{L}^+ is either itself a primitive expression or abbreviates exactly one primitive expression.

Notice that a and b describe independent properties. The result (a) that every sequence of disabbreviations starting with an expression **E** must terminate after a finite number of steps does not imply (b) that they must terminate at the same expression. Likewise, the result that every sequence of disabbreviations of **E** that terminates does so at the same expression does not preclude the existence of other sequences that fail to terminate.

If **A** is a non-primitive expression of \mathcal{L}^+ , the b part of theorem 3.2 allows us to speak without ambiguity of the unabbreviated form of **A**; and we shall extend this notion to all the expressions of \mathcal{L}^+ by stipulating that the unabbreviated form of a formula **A** of \mathcal{L} is **A** itself.

The basic principle governing the use of abbreviations is that they are taken to stand in for

the expressions they abbreviate; it is as if they were these expressions. As a consequence of this principle, abbreviations are assumed to acquire the properties of the expressions that they abbreviate. So, in particular, the definition of a formula of \mathcal{L}^+ will not be given in the usual manner, but instead we shall say that an expression is a formula of \mathcal{L}^+ if it is either a formula of \mathcal{L} or an abbreviation of such a formula.

definition 3.3

- i) \mathbf{A} is a formula of \mathcal{L}^+ if \mathbf{A} is a formula of \mathcal{L} .
- ii) If \mathbf{A} abbreviates \mathbf{A}' then \mathbf{A} is a formula if \mathbf{A}' is.

This is an inductive definition. If \mathbf{A} abbreviates some expression \mathbf{A}' , then whether \mathbf{A} is a formula depends on whether \mathbf{A}' is a formula, which may in turn depend on whether some \mathbf{A}'' that \mathbf{A}' abbreviates is a formula, and so on. By theorem 3.2, this process must eventually terminate in a primitive formula whose formulahood is determined by clause i. To be certain that abbreviations acquire the "non-formulahood", as well as the formulahood, of the formulas they abbreviate we must show that the **ifs** in clauses i and ii of definition 3.3 can be strengthened to **iff**.

Theorem 3.4

- i) If \mathbf{A} is primitive then \mathbf{A} is a formula of \mathcal{L}^+ iff \mathbf{A} is a formula of \mathcal{L} .
- ii) If \mathbf{A} abbreviates \mathbf{A}' then \mathbf{A} is a formula iff \mathbf{A}' is.

Instead of taking formulahood to be an *acquired* property of expressions of \mathcal{L}^+ , which they inherit from the expressions they abbreviate, it would also be quite natural to regard it as an *autonomous* property, which they have in their own right. In that case we would give direct formation rules like 1.1 for the formulas of \mathcal{L}^+ :

- (i) $\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s}$, and $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots$ are formulas
- (ii) \top and \perp are formulas;
- (iii) If \mathbf{A} is a formula so is $\neg\mathbf{A}$;
- (iv) If \mathbf{A} and \mathbf{B} are formulas so are $(\mathbf{A} \vee \mathbf{B})$, $(\mathbf{A} \wedge \mathbf{B})$, $(\mathbf{A} \supset \mathbf{B})$ and $(\mathbf{A} \equiv \mathbf{B})$;

Each approach results in the same expressions of \mathcal{L}^+ being formulas. Since we take formulahood to be an acquired notion, we consider it to be a theorem (rather than a definition) that clauses i-iv determine the class of formulas. But if we took formulahood to be autonomous, it would be a theorem (and not a definition) that formulas are expressions that abbreviate formulas. More precise statements and proofs of both results are outlined in the problems.

The equivalence of the two accounts of formula makes it appropriate for us to regard \supset , \wedge , and \equiv , like \vee , as binary connectives, which combine with two formulas to produce another formula, and \top and \perp as "zero-ary connectives", which combine with zero formulas to produce a formula. Terminology for describing formulas is extended to \mathcal{L}^+ in the standard way. $(\mathbf{A} \supset \mathbf{B})$ is a conditional with antecedent \mathbf{A} and consequent \mathbf{B} . $(\mathbf{A} \wedge \mathbf{B})$ is a conjunction with left conjunct \mathbf{A} and right conjunct \mathbf{B} . $(\mathbf{A} \equiv \mathbf{B})$ is a biconditional.

Given the notion of a formula of \mathcal{L}^+ (whether acquired or autonomous) it is possible to loosen the constraints on disabbreviation. Let us call the abbreviation relation that results from allowing the variables in definition 3.1 to range over all the formulas of \mathcal{L}^+ , free abbreviation. According to this notion of abbreviation, it is permissible, in the disabbreviation sequence

discussed above, to eliminate the \wedge after line 4, because its conjuncts, $((\mathbf{p}_1 \supset (\neg \mathbf{p}_1 \vee \mathbf{p}_1))$ and $((\neg \mathbf{p}_1 \vee \mathbf{p}_1) \supset \mathbf{p}_1)$ are formulas of the extended language. The reader may easily verify that disabbreviating in this order will not change the "outcome" of the process. More generally, it can be shown that free abbreviation is well-behaved and consistent with ordinary abbreviation in the sense made precise below.

Theorem 3.5 Every chain of free disabbreviation starting from an expression \mathbf{E} of \mathcal{L}^+ terminates, after a finite number of steps, in the (ordinary) unabbreviated form of \mathbf{E} .

This result ensures that if \mathbf{A} freely abbreviates \mathbf{B} then \mathbf{A} and \mathbf{B} share all properties acquired from the formulas they abbreviate. Thus it is safe to employ free abbreviations in any context in which ordinary ones can be employed, and indeed, it is safe to ignore the distinction between ordinary and free abbreviation altogether. We leave as an exercise the formulation of examples to show that it is not safe to allow the variables in clauses (iv)-(vi) of definition 3.1 to range over arbitrary expressions of \mathcal{L}^+ .

Truth in \mathcal{L}^+ , like formulahood, can be treated as an acquired property or an autonomous one. The definition below and the two subsequent results reflect the first option.

definition 3.6 If \mathbf{A} abbreviates \mathbf{A}' then $\alpha \models \mathbf{A}$ if $\alpha \models \mathbf{A}'$.

Like 3.3, this is an inductive definition. The truth of \mathbf{A} depends on the truth of some \mathbf{A}' that \mathbf{A} abbreviates, and so on. Theorem 3.5 guarantees that this chain of dependence must stop at a formula of \mathcal{L} , for which truth value is already defined. To be certain that abbreviations acquire the falsity, as well as the truth, of the formulas they abbreviate we must show:

Theorem 3.7 If \mathbf{A} abbreviates \mathbf{A}' then $\alpha \models \mathbf{A}$ iff $\alpha \models \mathbf{A}'$.

One method of proving 3.7 is outlined in the problems.

The next result shows that the clauses for \neg and \vee in the original definition of \models yield the appropriate clauses for the connectives of \mathcal{L}^+ .

Theorem 3.8 For any valuation α and any formulas \mathbf{A} and \mathbf{B} of \mathcal{L}^+ ,

(i) If \mathbf{A} is $\mathbf{p}, \mathbf{q}, \mathbf{r}$, or \mathbf{s} then $\alpha \models \mathbf{A}$ iff $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$, or \mathbf{p}_4 , respectively, are members of α ; If \mathbf{A} is a sentence letter of \mathcal{L} then $\alpha \models \mathbf{A}$ iff $\mathbf{A} \in \alpha$;

(ii) $\alpha \models \neg \mathbf{A}$ iff it is not the case that $\alpha \models \mathbf{A}$;

(iii) $\alpha \models \top$;

(iv) $\alpha \not\models \perp$;

(v) $\alpha \models (\mathbf{A} \vee \mathbf{B})$ iff either $\alpha \models \mathbf{A}$ or $\alpha \models \mathbf{B}$;

(vi) $\alpha \models (\mathbf{A} \supset \mathbf{B})$ iff either $\alpha \not\models \mathbf{A}$ or $\alpha \models \mathbf{B}$;

(vii) $\alpha \models (\mathbf{A} \wedge \mathbf{B})$ iff $\alpha \models \mathbf{A}$ and $\alpha \models \mathbf{B}$;

(viii) $\alpha \models (\mathbf{A} \equiv \mathbf{B})$ iff either both $\alpha \models \mathbf{A}$ and $\alpha \models \mathbf{B}$ or neither $\alpha \models \mathbf{A}$ nor $\alpha \models \mathbf{B}$.

Proof According the remarks above, the formulas of \mathcal{L}^+ are generated from the sentence letters and \top and \perp by forming negations, disjunctions, conditionals, conjunctions and biconditionals. We proceed by induction on formulas as so generated.

Basis. If \mathbf{A} is $\mathbf{p}, \mathbf{q}, \mathbf{r}$, or \mathbf{s} then by theorem 3.7 $\alpha \models \mathbf{A}$ iff $\alpha \models \mathbf{p}_i$ for the appropriate \mathbf{p}_i . By definition 2.1 this holds iff $\mathbf{p}_i \in \alpha$. If \mathbf{A} is any other sentence letter, it is a formula of \mathcal{L} , so there is nothing to prove. If $\mathbf{A} = \top$ then by definition 3.6 $\alpha \models \mathbf{A}$ iff $\alpha \models \neg \mathbf{p}_1 \vee \mathbf{p}_1$. But this holds for any α . If $\mathbf{A} = \perp$ then by theorem 3.7 $\alpha \models \mathbf{A}$ iff $\alpha \models \neg(\neg \mathbf{p}_1 \vee \mathbf{p}_1)$. This holds for no α .

Induction step. We do the case for negation. The other cases are similar and are left as exercises. Let \mathbf{A}^* be the unabbreviated form of \mathbf{A} . Then $\alpha \models \neg \mathbf{A}$ iff $\alpha \models (\neg \mathbf{A})^*$ (by theorem 3.7) iff $\alpha \models \neg \mathbf{A}^*$ iff $\alpha \not\models \mathbf{A}^*$ (by the truth definition for \mathcal{L}) iff $\alpha \not\models \mathbf{A}$ (by theorem 3.7).

In light of these results, it is natural to read \supset as **only if** (or $\mathbf{A} \supset \mathbf{B}$ as **if \mathbf{A} then \mathbf{B}**), \wedge as **and**, and \equiv as **if and only if**. However, strictly speaking, the interpretation of formulas containing these new connectives should be given by the interpretation of the formulas that they abbreviate.

Under an autonomous treatment of truth, the clauses of theorem 3.8 would constitute a truth-definition of a regular sort for \mathcal{L}^+ . 3.6 and 3.7 could then be proved as theorems. Under any assignment, either approach would result in the same formulas being true and being false.

Although we prefer to take most properties of the expressions of \mathcal{L}^+ to be acquired, there are some that it will be important to regard as autonomous. Under an autonomous treatment of logical form, $\mathbf{p} \wedge \mathbf{q}$ would represent a relatively simple form, one involving only one connective, whereas under the abbreviatory approach described here it indirectly represent a relatively complex form, one directly represented by $\neg(\neg \mathbf{p} \vee \neg \mathbf{q})$. It seems clear that the former more faithfully represents what we take to be the form of a conjunctive sentence or a conjunctive proposition. Our understanding of their form, after all, does not involve the analysis of the conjunction in terms of disjunction and negation. So, to the extent that we wish to represent the form of such sentences or propositions we should adopt an autonomous approach.

We employ two additional devices of abbreviation. First, we permit conjunctions, disjunctions, and conditionals of three or more formulas. These are to be disabbreviated by grouping to the right. For example, $\neg(\mathbf{p} \vee \mathbf{q} \vee \mathbf{r} \vee \mathbf{s})$ is understood to abbreviate $\neg(\mathbf{p} \vee (\mathbf{q} \vee (\mathbf{r} \vee \mathbf{s})))$, and $(\mathbf{p} \supset \mathbf{q} \supset \mathbf{r})$ to abbreviate $(\mathbf{p} \supset (\mathbf{q} \supset \mathbf{r}))$. Second, we often omit the outermost brackets of a formula. For example, $\neg \mathbf{p}_1 \vee \mathbf{p}_3$ abbreviates $(\neg \mathbf{p}_1 \vee \mathbf{p}_3)$. This last abbreviatory device is of a somewhat different character from all the others. Until now an abbreviation could be used, with the same effect, in any context. But we clearly cannot omit, without ambiguity, the outermost brackets of a formula that is embedded within another. The new devices do, however, preserve the important properties expressed in theorem 3.5 and problem 5.

Exercises and problems

1[e]. (i) Find the unabbreviated form of the following formulas.

- a. $\mathbf{p} \supset (\mathbf{q} \wedge \mathbf{p})$
- b. $(\mathbf{q} \supset \perp) \vee (\perp \supset \mathbf{q})$
- c. $(\mathbf{q} \equiv \mathbf{s}) \supset \mathbf{r}$
- d. $\mathbf{r} \supset (\mathbf{q} \equiv \mathbf{s})$

*(ii) What is the quickest way to disabbreviate c? What is the quickest way to disabbreviate d? What general rule should one follow to obtain the quickest disabbreviation? What does **quickest** mean here?

2[e]. Suppose that the sentence letters of \mathcal{L} were regarded, not as individual symbols, but as expressions containing \mathbf{p} and a numerical subscript. Show that the abbreviation relation would not be well-founded.

3[e]. Show that there are expressions of \mathcal{L}^+ that abbreviate more than one expression of \mathcal{L} [Hint: Consider "formulas" with extra sets of parentheses].

4[e]. Do one of the remaining cases in the proof of theorem 3.8.

5[e]. Find distinct formulas \mathbf{A} , \mathbf{B} , \mathbf{B}' such that $\mathbf{A} \gg \mathbf{B}$ and $\mathbf{A} \gg \mathbf{B}'$. Now find a fourth formula \mathbf{C} such that $\mathbf{B} \gg \mathbf{C}$ and $\mathbf{B}' \gg \mathbf{C}$.

6[e]. Suggest an informal semantics for the language of \mathcal{L}^+ that respects the formal semantics given here. Make your account accurate enough to avoid ambiguities in the scope of English connecting words.

7[p]. (Unique unabbreviated form) Prove theorem 3.5. Hints: First prove that the particular disabbreviation procedure described in the text always terminates, and that the resulting expression is a formula of \mathcal{L} . It follows that to prove theorem 3.5 it is sufficient to prove:

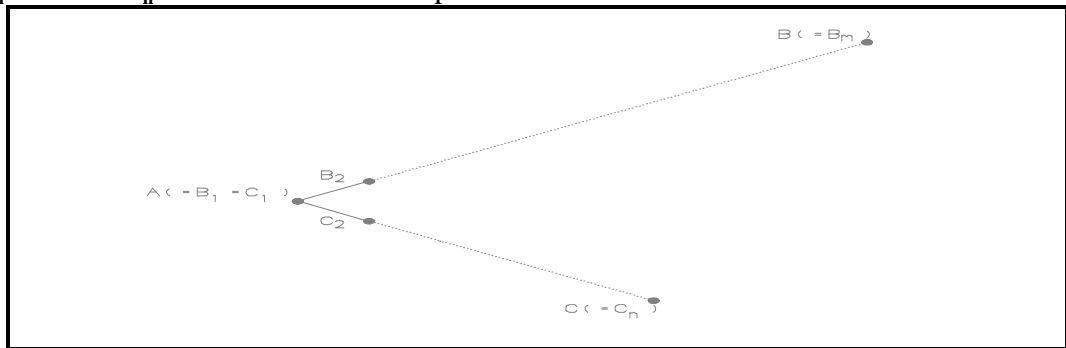
(*) If $\mathbf{A} \gg \mathbf{B}$ and $\mathbf{A} \gg \mathbf{C}$ then, for some \mathbf{D} , $\mathbf{B} \gg \mathbf{D}$ and $\mathbf{C} \gg \mathbf{D}$.

(where $\mathbf{A} \gg \mathbf{B}$ indicates that $\mathbf{A} \gg \mathbf{B}$ or $\mathbf{A}=\mathbf{B}$).

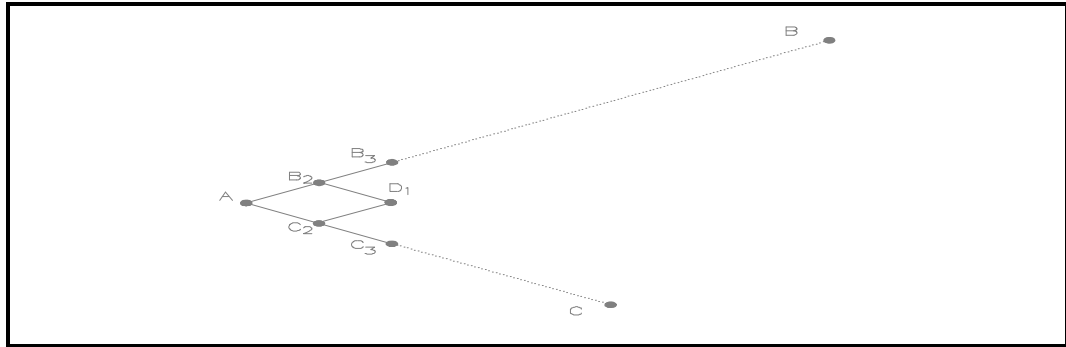
To prove (*) it is convenient to establish something a little stronger. Let $\mathbf{A} \Rightarrow \mathbf{B}$ iff \mathbf{B} is obtained from \mathbf{A} by replacing one or more occurrences of \mathbf{C} in \mathbf{A} with \mathbf{C}' where $\mathbf{C} \triangleright \mathbf{C}'$. (We resist the temptation to say that in this case \mathbf{A} is multi-subdirectly-reducible to \mathbf{B} .) Let $\mathbf{A} \Rightarrow_n \mathbf{C}$ iff there are formulas $\mathbf{B}_1, \dots, \mathbf{B}_n$ ($n > 0$) such that $\mathbf{A}=\mathbf{B}_1$, $\mathbf{C}=\mathbf{B}_n$ and $\mathbf{B}_1 \Rightarrow \dots \Rightarrow \mathbf{B}_n$. Thus $\mathbf{A} \Rightarrow_0 \mathbf{B}$ iff $\mathbf{A}=\mathbf{B}$, $\mathbf{A} \Rightarrow_1 \mathbf{B}$ iff $\mathbf{A} \Rightarrow \mathbf{B}$, and $\mathbf{A} \Rightarrow_n \mathbf{B}$ for some n iff $\mathbf{A} \gg \mathbf{B}$. Then a more informative version of (*) is:

(**) If $\mathbf{A} \Rightarrow_m \mathbf{B}$ and $\mathbf{A} \Rightarrow_n \mathbf{C}$ then, for some \mathbf{D} , $\mathbf{B} \Rightarrow_n \mathbf{D}$ and $\mathbf{C} \Rightarrow_m \mathbf{D}$.

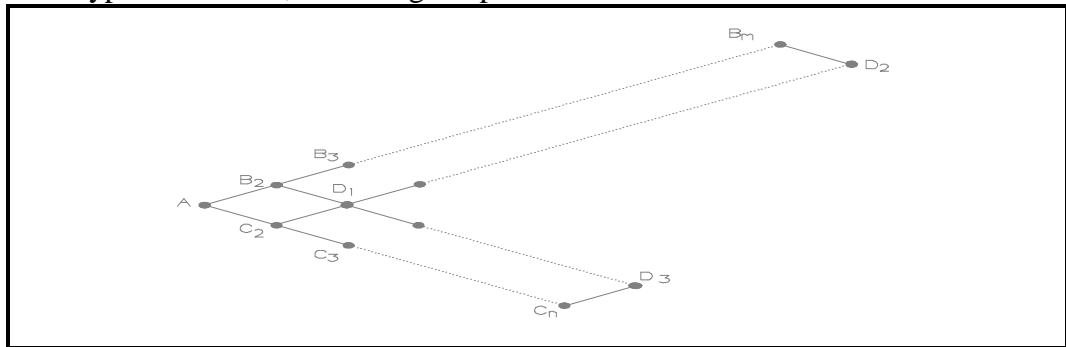
The whole of (**) can be proved just from the case in which m and n are one. To see this suppose $\mathbf{A} \Rightarrow_m \mathbf{B}$ and $\mathbf{A} \Rightarrow_n \mathbf{C}$. The situation is depicted below.



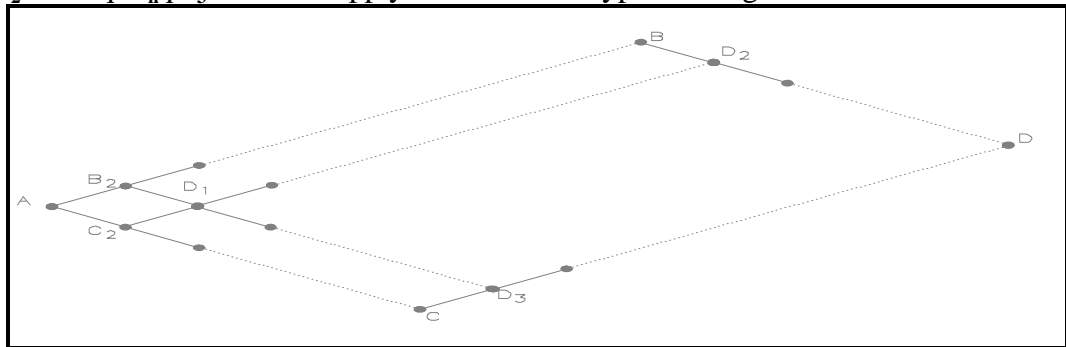
We must show that for some \mathbf{D} , $\mathbf{B} \Rightarrow_n \mathbf{D}$ and $\mathbf{C} \Rightarrow_m \mathbf{D}$. We proceed by induction on the larger of m and n —henceforth, $\max(m,n)$. First suppose m and n are both zero. Then $\mathbf{A}=\mathbf{B}$ and $\mathbf{A}=\mathbf{C}$, which implies $\mathbf{B}=\mathbf{C}$. So $\mathbf{D}=\mathbf{A}$ is the required \mathbf{D} . Now suppose the theorem holds when $\max(m,n) < k$ and also (if necessary) for the case in which m and n are both one. Then we can extend the diagram as shown below.



Here we have $B_2 \Rightarrow \dots \Rightarrow B_n$ and $B_2 \Rightarrow D_1$, as well as $C_2 \Rightarrow \dots \Rightarrow C_m$ and $C_2 \Rightarrow D_1$. We can now apply the induction hypothesis twice, extending the picture as shown.



Here $D_1 \Rightarrow_{m-1} D_2$ and $D_1 \Rightarrow_{n-1} D_3$. We can apply the induction hypothesis again as shown below.



A glance at the picture confirms that D has the required property. Thus to prove (**), you need only prove that if $A \Rightarrow B$ and $A \Rightarrow C$ then, for some D , $B \Rightarrow D$ and $C \Rightarrow D$. This can be done by a formula induction on A .

8[p]. (Termination of disabbreviation) The proof sketched in problem 7 above shows that for every formula A of \mathcal{L}^+ there is a formula A' of \mathcal{L} such that every sequence of disabbreviation steps from A that does not terminate at A' can be continued so that it does terminate at A' . It does not, however, rule out the possibility of an infinite sequence of disabbreviation steps, i.e., an infinite chain of the form $A_1 \triangleright A_2 \triangleright \dots$.

a. Give an alternative proof of theorem 3.5 which shows that every sequence of disabbreviation steps must terminate. Hint: The definition of direct abbreviation establishes a total order on the special symbols of \mathcal{L}^+ . Represent a formula of \mathcal{L}^+ as a labeled tree. Each branch of such a tree contains a finite set of occurrences of symbols. Associate the branches with their sets of symbol occurrences and associate the formula with the set of all the sets associated with the branches. The ordering on connectives can then be extended to an ordering on branches and further extended to an ordering on formula trees. (See section 3 of the appendix.) Show that every disabbreviation step reduces the order of a formula and that the formulas of \mathcal{L} have minimal order.

b. Give a general characterization of the kinds of abbreviation to which the arguments of problem 7 and part a of this problem could apply. (A natural way to do this would be to give general conditions that the clauses in the definition of direct abbreviation should satisfy.)

9[p]. (Proof of theorem 3.7). The formulas of \mathcal{L}^+ are generated from those of \mathcal{L} by the relation \Rightarrow of problem 7 above. Prove theorem 3.7 by induction on \mathbf{A} as so generated. [Hint: Suppose \mathbf{A} abbreviates \mathbf{B} , $\alpha \models \mathbf{A}$ but $\alpha \not\models \mathbf{B}$. According to definition 3.6, $\alpha \models \mathbf{A}$ means that \mathbf{A} abbreviates some formula \mathbf{C} that is true relative to α . By (***) of problem 7, for some formula \mathbf{D} , $\mathbf{B} \Rightarrow \mathbf{D}$ and $\mathbf{C} \Rightarrow \mathbf{D}$. The induction hypothesis then shows that $\alpha \models \mathbf{C}$ implies $\alpha \models \mathbf{D}$, even though $\alpha \not\models \mathbf{B}$ implies $\alpha \not\models \mathbf{D}$.]