Expanded Categorial Syntax

Gary Hardegree
Department of Philosophy
University of Massachusetts
Amherst, MA 01003

---

1. Introduction 2
2. Case Marking 2
3. Case-Inflected Types 4
4. Re-Categorizing Verbs 4
5. Simple Examples 5
6. Type Mismatch! 7
7. Standard Categorial-Composition and Logic 7
8. Expanded Categorial-Composition 8
9. The Logic of Grammatical-Composition 9
10. Classical Logic and Intuitionistic Logic do not Correctly Model Grammatical-Composition 9
11. Monotonic Logics do not Correctly Model Grammatical-Composition 10
12. Relevance Logic (Non-Monotonic Variant) 11
13. Examples of Valid Arguments in Relevance Logic 11
14. Examples of Invalid Arguments in Relevance Logic 12
15. Relevance Logic Does not Properly Model Grammatical-Composition 12
16. Linear Logic 13
17. Hyper-Relevance Logic 13
18. The Problem 14
19. Summary of Logical Systems Considered 14

---

1. Introduction

In order to overcome the numerous difficulties mentioned in the previous chapter, we propose to revise and expand categorial grammar in the following ways.

(1) we add case-markers, which include the usual thematic cases as well as anaphoric cases;
(2) we add category-multiplication (×) as a full-fledged category operation;
(3) we significantly expand the rules for categorial-composition.

Although each of these proposals does significant theoretical work for us, the most profound adjustment is item (3). According to standard categorial grammar, grammatical-composition consists exclusively in *function-application*. In particular, one takes an item of type $A \to B$ (function from A into B) and an item of type A (element of A) and constructs an item of type B (element of B).[1] Thinking of this as a logical operation, in which $\to$ is understood as if-then, grammatical-composition corresponds to the argument form known as *modus ponens*.

We propose to expand grammatical-composition from a *single* inference form (*modus ponens*) to *infinitely-many* inference forms – indeed all valid inference forms! The obvious question then is what logical system do we use to judge validity. We propose that the appropriate logical system is a weakened form of system R of relevance logic, which we refer to as Categorial Logic. Relevance logic is characterized by its insistence that there be a *connection* between antecedents and consequents of true conditionals, and between premises and conclusions of valid arguments. Most well-known systems of logic do not satisfy this requirement. On the other hand, relevance criteria seem especially appropriate to the logic of grammatical-composition, since a *sine qua non* of valid grammatical-composition is that *all* inputs are used in arriving at the output.

2. Case Marking

Cases mark *roles* of noun-phrases in respect to verbs, where the marking is usually accomplished by one of three syntactic devices

word-order
inflection[2]
adposition

the latter of which includes pre-position and post-position. For example, English *mostly* uses word-order to distinguish subjects of verbs from objects of verbs, but it also uses inflection and prepositions to mark cases.

On the other hand, Korean and Japanese use postpositions. For example, in the Japanese sentence[3]

John ga Mary ni hon o yatta

[1] There is also polyadic function application, which we ignore for the moment, but consider in Section 0.
[2] Linguists use the word 'inflection' both in phonology and morphology. In morphology, 'inflection' refers to numerous methods of word "bending", including verb and noun adjustments pertaining to person, number, gender, tense, aspect, mood, voice, and case.
[3] This example is borrowed from www.sil.org.

the postposition 'ga' indicates that 'John' is the subject of the verb 'yatta' [= 'gave'], whereas the postposition 'ni' indicates that 'Mary' is the indirect object, and the postposition 'o' indicates that 'hon' [= 'book'] is the direct object.

By contrast, the following sentence employs the other three case-marking methods.

Jay sold his car to Kay

First, the preposition 'to' indicates that 'Kay' is the indirect object of the verb 'sold'. The subject of the verb is 'Jay', which is so marked by preceding the verb and the direct object is 'his car', which is so marked by following the verb. Finally, the word 'his' is an inflected form of 'he' used to indicate possession.

Note that, since it makes no difference to semantics, we do not distinguish inflection, which is part of word structure (morphology), from other case-marking methods, including word order and adposition. In particular, for the sake of simplifying our terminology, we will loosely call all these devices 'case-inflections'. Also, we will concentrate on case-inflection, so when we use the word 'inflection' without modification, we *usually* mean case-inflection.

In English, the prominent cases include the following.

nominative	usually the *subject* of a verb phrase, usually marked by prefixing { I am | you are | he is | she is | we are | they } are happy
accusative	usually the *direct object* of a transitive verb phrase, usually marked by postfixing everyone respects { me | you | him | her | us | them }
dative	usually the *indirect object* of a di-transitive verb, usually marked by 'to' Jay wrote a letter to { me | you | him | her | us | them }
ablative	usually the *indirect object* of a di-transitive verb, usually marked by 'from' Kay received a letter from { me | you | him | her | us | them }
bylative[4]	usually the agent in a passive construction, usually marked by 'by' Jay is respected by Kay { me | you | him | her | us | them }
genitive[5]	associated with certain *inherently-relational* common nouns, including: mother, brother, friend, capital marked by 'of' and also by the suffix apostrophe-'s' (plus irregular forms)

[4] There is no generally accepted name for this case; we propose 'bylative', which is 'by' plus 'lative', is meant to be parallel to 'ablative', which is 'ab' ['from'] plus 'lative'.

[5] Genitive case and possessive "case" are morphologically very similar, but semantically quite different. We do not include possessive case, since we prefer to treat possessive forms as adjectival prepositions, not as case-markers. See later chapter.

3. Case-Inflected Types

For the purpose of categorially rendering cases, we propose case-inflected types, including the following.[6]

(1)	D_1	nominative-inflected	definite-noun phrases
(2)	D_2	accusative-inflected	definite-noun phrases
(3)	D_3	dative-inflected	definite-noun phrases
(4)	D_4	ablative-inflected	definite-noun phrases
(5)	D_5	bylative-inflected	definite-noun phrases
(6)	D_6	genitive-inflected	definite-noun phrases

The inflectional morphemes themselves are notated, and categorized, as follows.[7]

(1)	[+1]	nominative inflection	$D \rightarrow D_1$	
(2)	[+2]	accusative inflection	$D \rightarrow D_2$	
(3)	[+3]	dative inflection	$D \rightarrow D_3$	$D_2 \rightarrow D_3$
(4)	[+4]	ablative inflection	$D \rightarrow D_4$	$D_2 \rightarrow D_4$
(5)	[+5]	bylative inflection	$D \rightarrow D_5$	$D_2 \rightarrow D_5$
(5)	[+6]	genitive inflection	$D \rightarrow D_6$	$D_2 \rightarrow D_6$

More generally, we propose to use integers to encode cases. In addition to the "positive" cases, which correspond to thematic roles, we also propose the "nullative" case, encoded by 0, and infinitely-many "negative" cases, encoded by negative integers. The latter are used for anaphoric cross-referencing and binding.

4. Re-Categorizing Verbs

Next, we propose to categorize all verbs according to what inflectional-types they take as arguments. For example, an ordinary transitive verb like 'respect' takes both a nominative argument and an accusative argument, and is categorized as follows.

type(respect) = $D_2 \rightarrow (D_1 \rightarrow S)$

On the other hand, di-transitive verbs come in at least two varieties, illustrated as follows.

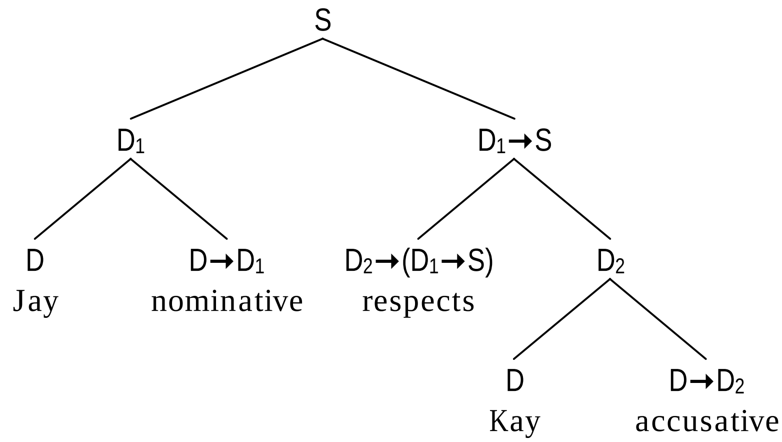
type(sell) = $D_2 \rightarrow [D_3 \rightarrow (D_1 \rightarrow S)]$
type(buy) = $D_2 \rightarrow [D_4 \rightarrow (D_1 \rightarrow S)]$

For example 'sell' takes an accusative argument, and delivers a functor that takes a dative argument, and delivers a functor that takes a nominative argument and delivers a sentence.

[6] More generally, we propose to case-inflect all types, and we propose an infinite sequence of case-markers.
[7] Note carefully that the last four case-markers are presented with two types. The secondary type is proposed in order to render case-marking prepositions syntactically similar to adjunctive prepositions.

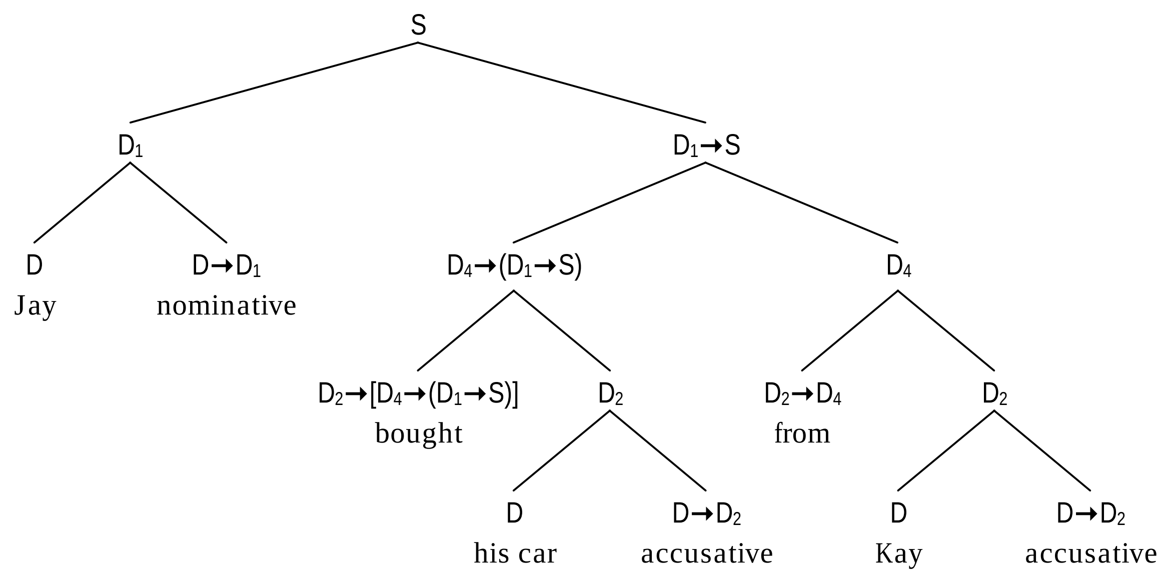
5. Simple Examples

1. Jay respects Kay



In this example, the nominative and accusative morphemes are not pronounced in English,[8] although they are pronounced in other languages – e.g., Latin and German [using inflection], and Korean and Japanese [using post-positions].

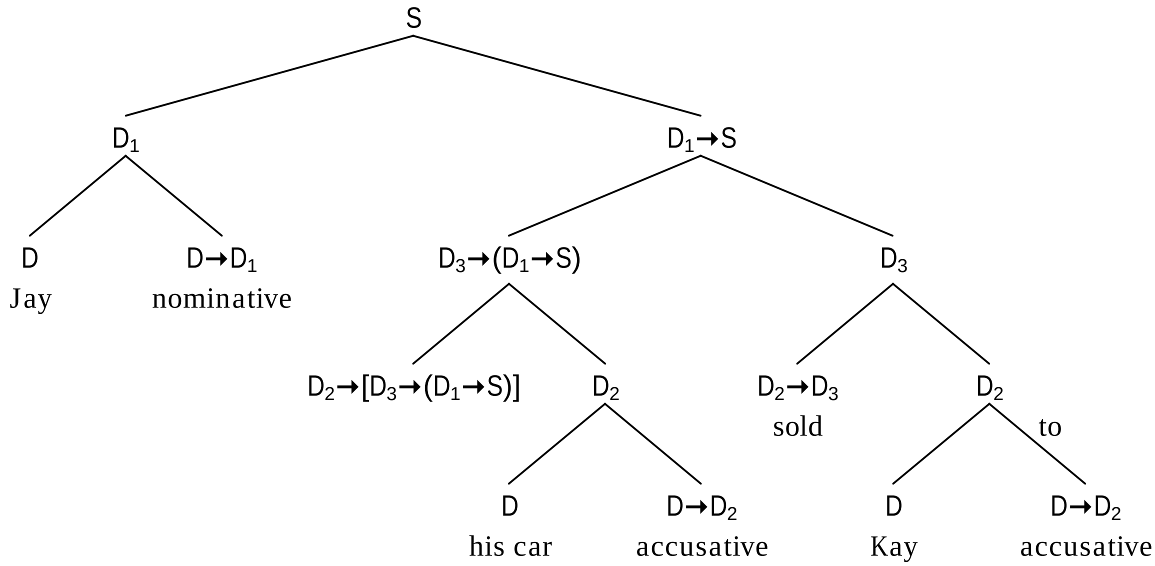
2. Jay bought his car from Kay



We propose to treat 'from' as a case-marking pre-position, which in particular takes an accusative input and delivers an ablative output. The following is a similar example.

[8] Except in the sense that the subject must be pronounced before the verb, and the object must be pronounced after the verb.

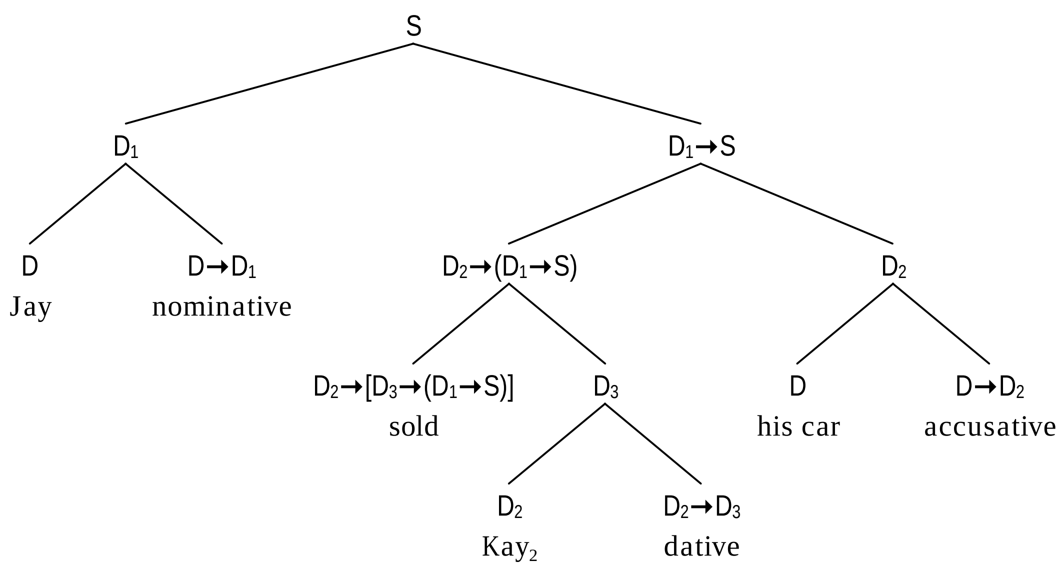
3. Jay sold his car to Kay



We similarly propose to treat ‘to’ as a case-marking preposition, which in particular takes an accusative input and delivers a dative output.

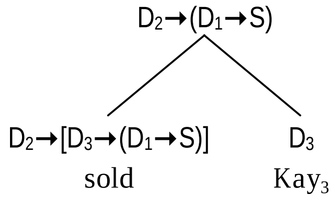
We next observe that English offers an alternative dative construction, illustrated in the following example.

4. Jay sold Kay his car



6. Type Mismatch!

We now face an immediate and serious categorial problem. In particular, in the previous example, notice the following sub-tree.



Notice in particular that 'sell' officially takes an accusative argument, but 'Kay_3' is dative. Thus, we have a type mismatch.

We propose to solve this problem by expanding grammatical-composition.

7. Standard Categorial-Composition and Logic

According to standard categorial grammar, grammatical-composition consists *exclusively* in *function-application*, of which there are two closely-related forms. In its simplest form, one takes an item of type $A \rightarrow B$ and an item of type A, and one constructs an item of type B.

This procedure can be depicted as follows.

$$\begin{array}{l} A \rightarrow B \\ A \\ \hline B \end{array}$$

Now, if we think of arrow as the logical if-then connective, then this composition procedure corresponds to the following argument form.

$$\begin{array}{l} \text{if } A\text{, then } B \\ A \\ \hline B \end{array}$$

This is the most basic inference form of all – *modus ponens*.

There is also the general form of function-application. In this form, one takes an item of type $(A_1 \times \ldots \times A_k) \rightarrow B$ and items of respective types $A_1, \ldots, A_k$, and one constructs an item of type B. This may be depicted as follows.

$$\begin{array}{l} (A_1 \times \ldots \times A_k) \rightarrow B \\ A_1 \\ \ldots \\ A_k \\ \hline B \end{array}$$

Once again, we can translate this into logic, by translating '$\times$' as 'and', and '$\rightarrow$' as 'if…then', in which case we obtain the following argument form.

if A_1 and … and A_k , then B
A_1
…
A_k
___
B

This too is a valid argument form, known as *multiple modus ponens*.

8. Expanded Categorial-Composition

Now let's go back and look at the suspect composition from Section 5.

$D_2 \rightarrow [D_3 \rightarrow (D_1 \rightarrow S)]$
D_3
___
$D_2 \rightarrow (D_1 \rightarrow S)$

Although it is a bit more complicated, this too corresponds to a valid argument form.

if A_2, then if A_3, then if A_1, then B
A_3
___
if A_2, then if A_1, then B

This, in turn, is a special case of the following argument form.

if A, then if B, then C
B
___
if A, then C

We propose to call this argument form *secondary modus ponens*.

In order to authorize the above composition, and many others, we propose to expand grammatical-composition as follows.

> Let $\mathcal{A}_1 , \ldots, \mathcal{A}_k , C$ be syntactic-types. Then C is an **admissible grammatical product** of $\mathcal{A}_1 , \ldots, \mathcal{A}_k$ precisely if:
>
> $$\mathcal{A}_1 , \ldots, \mathcal{A}_k \vdash C$$
>
> where all the symbols are understood as logical.

The latter clause subdivides as follows.

(1)	$\rightarrow$	is interpreted as	if-then
(2)	$\times$	is interpreted as	and[9]
(3)	$\vdash$	is interpreted as	logical entailment[10]

9. The Logic of Grammatical-Composition

The obvious next question is what logical system do we use to judge logical entailment? In this connection, there are several prominent choices for the logic of 'if-then', including the following.

Some well-known initial candidates include:

(1)	classical logic	CL	Frege, Russell, …
(2)	intuitionistic logic	IL	Brouwer
(3)	relevance logic	RL	Anderson & Belnap
(4)	linear logic	LL	Jean-Yves Girard
(5)	strict-entailment logic	SEL	C.I. Lewis
(6)	subjunctive-conditional logic	SCL	D. Lewis, Stalnaker

10. Classical Logic and Intuitionistic Logic do not Correctly Model Grammatical-Composition

Since it is the strongest logic,[11] Classical Logic provides the maximum number of valid-compositions, but unfortunately it also provides compositions we do not want! For example, the following is a principle of Classical Logic.[12]

(c1) $P ; Q \vdash P \rightarrow Q$

Accordingly, if we use Classical Logic to model grammatical-composition, then the following composition is officially licensed.

$S ; S \vdash S \rightarrow S$

In other words, it is grammatically admissible to compose a sentential-operator out of two sentences. Now, it seems highly implausible that two sentences can be composed into a sentential-operator, so we must reject (c1) in the logic of grammatical-composition.

In an important sense, to be explained shortly, (c1) follows from the following inference principle, which is also valid in Classical Logic.

(c2) $Q \vdash P \rightarrow Q$

---

[9] More properly, the multiplicative-counterpart of 'if-then', which in classical logic and intuitionistic logic is logical-and, but in other logics is more complicated. Nevertheless, our intuitions will generally be well-served by thinking of $\times$ as logical-and.

[10] Logical entailment is a relation expressed in the meta-language, where the formula-homonyms are names in the meta-language of the formulas in the object-language. For example, the following expression in the meta-language

$P \rightarrow Q ; P \vdash Q$

is the claim that the formula 'Q' follows logically from the two formulas 'P→Q' and 'P'.

[11] If a weaker logic validates an argument form $\mathbb{A}$, then a stronger logic also validates $\mathbb{A}$, granting the two logics both pass judgment on $\mathbb{A}$.

[12] Although nothing hinges on this, we use one arrow symbol '→' for the logical-arrow operator, and another arrow symbol '➞' for the categorial-arrow operator.

This is an inference principle found especially obnoxious by many pioneers of alternative logical systems, including the strict-entailment logics of C.I. Lewis[13] and the relevant-entailment logics of Anderson and Belnap.[14]

Grammatically-understood, (c2) is equally obnoxious, since it has the following instance

$$S \vdash D \rightarrow S$$

which authorizes transforming a sentence into a verb phrase, a procedure that seems grammatically inadmissible.

In addition to (c1) and (c2), the following Classical Logic principles also yield implausible composition principles.[15]

(c3) $(P \rightarrow Q) \rightarrow Q \vdash (Q \rightarrow P) \rightarrow P$
(c4) $(P \rightarrow Q) \rightarrow P \vdash P$

In light of the numerous examples of invalid grammatical-compositions proffered by Classical Logic, we must accordingly conclude that Classical Logic does not properly model grammatical composition.

Before continuing, we note that both (c1) and (c2) [but not (c3) and (c4)] are also principles of Intuitionistic Logic, so we must similarly conclude that Intuitionistic Logic does not properly model grammatical composition either.

11. Monotonic Logics do not Correctly Model Grammatical-Composition

A principle that is fundamental to nearly every logical system that has been considered over the past few millennia is the principle of *monotonicity*. The basic idea is that a valid argument cannot be made into an invalid argument simply by adding premises. The following meta-principle is a special case of monotonicity.[16]

$$\mathcal{B} \vdash \mathcal{C} \;\Rightarrow\; \mathcal{A} \,;\, \mathcal{B} \vdash \mathcal{C}$$

Next, suppose we also grant the following *identity* principle

$$\mathcal{B} \vdash \mathcal{B}$$

which is generally regarded as a minimum requirement of any formal system that presumes to model reasoning. Putting monotonicity and identity together, we obtain the following *simplification principle*.

$$\mathcal{A} \,;\, \mathcal{B} \vdash \mathcal{B}$$

Suppose we include this principle in the logic of grammatical-composition. Then the following composition is authorized.

$$D \,;\, S \vdash S$$

[13] C.I. Lewis, ‘Implication and the Algebra of Logic’, *Mind* N.S., 21 (1912), 522-31.
[14] Alan Anderson and Nuel Belnap, *Entailment*, Princeton University Press, 1975.
[15] A single-premise argument form corresponds to a type-shifting move.
[16] Here, we use the arrow-symbol ‘⇒’ as the meta-language's if-then connective.

In the proposed composition, we compose a sentence out of a proper-noun phrase and a sentence – presumably by simply discarding the proper-noun phrase. However, it seems manifestly plausible that a valid grammatical-composition *must* meaningfully utilize *all* its inputs in constructing its output.

Accordingly, in order for a logic to model grammatical-composition, it cannot contain the simplification principle, and so it cannot be monotonic. Alas, nearly every logical system mentioned above is monotonic – the exceptions being Relevance Logic and Linear Logic.

12. Relevance Logic (Non-Monotonic Variant)

As its name suggests, Relevance Logic is characterized by a sensitivity to matters of *relevance* – in particular, between premises and conclusions of arguments, and between antecedents and consequents of conditional (if-then) statements. In particular, for relevance logic, a conditional statement cannot be true unless there is a connection between the antecedent and the consequent, and an argument cannot be valid unless there is a connection between *some* of the premises and the conclusion. Furthermore, in the *non-monotonic* variant of relevance logic, an argument cannot be valid unless there is a connection between *all* the premises and the conclusion. Most well-known systems of logic do not satisfy any of these requirements.

On the other hand, it seems that relevance *desiderata* are tailor-made for modeling grammatical-composition. For example, in grammar, it is presumed that a grammatical functor actually uses its input by way of generating its output, and it is also presumed that grammatical-composition uses all its input in producing its output.

13. Examples of Valid Arguments in Relevance Logic

(1)	$A \to (A \to B) \vdash A \to B$	[Contraction]
(2)	$B \to C \,;\, A \to B \vdash A \to C$	[Transitivity]
(3)	$A \to (B \to C) \vdash B \to (A \to C)$	[Permutation]
(4)	$A \to (B \to C) \,;\, A \to B \vdash A \to C$	[Conditional Modus Ponens]
(5)	$A \to (B \to C) \,;\, B \vdash A \to C$	[Secondary *Modus Ponens*]
(6)	$A \vdash (A \to B) \to B$	[Duality]
(7)	$(A \to A) \to B \vdash B$	[Law of Assertion]
(8)	$(A \to C) \to D \,;\, A \to B \vdash (B \to C) \to D$	[Inflection]
(9)	$(A \to C) \to D \,;\, A \to (B \to C) \vdash B \to D$	[Permutation + Transitivity]
(10)	$A \vdash A \times A$	[Duplication]
(11)	$B \to C \,;\, A \times B \vdash A \times C$	[Addition]
(12)	$(A \times B) \to C \dashv\vdash A \to (B \to C)$	[Schönfinkel's Law][17]
(13)	$A \to B \,;\, A \to C \vdash A \to (B \times C)$	[Conditional Multiplication]
(14)	$A \to B \,;\, A \to C \,;\, (B \times C) \to D \vdash A \to D$	[Generalized Conjunction]

[17] I hereby propose to name this principle after Moses Schönfinkel. See later for an explanation.

14. Examples of Invalid Arguments in Relevance Logic

※ $A \vdash B \to A$ [Positive Paradox]

(valid in Intuitionistic Logic, and hence Classical Logic)

※ $A \to B \vdash A \to (A \to B)$ [Expansion]

(valid in Intuitionistic Logic, and hence Classical Logic)

※ $A \vdash B \to B$ [Tautology]

(valid in Intuitionistic Logic, and hence Classical Logic)

※ $(A \to B) \to B \vdash (B \to A) \to A$ [Lukasiewicz's Law[18]]

(valid in Classical Logic, but not Intuitionistic Logic)

※ $(A \to B) \to A \vdash A$ [Peirce's Law]

(valid in Classical Logic, but not Intuitionistic Logic)

※ $A \times B \vdash A$
※ $A \times B \vdash B$ [Simplification]

(valid in Intuitionistic Logic and Classical Logic)

※ $A \to (B \times C) \vdash A \to B$
※ $A \to (B \times C) \vdash A \to C$ [Distribution]

(valid in Intuitionistic Logic and Classical Logic)

15. Relevance Logic Does not Properly Model Grammatical-Composition

Relevance Logic does a very decent job of modeling grammatical composition, but it is too strong. In particular, we note the following three problematic inferences.

1. Contraction

$$A \to (A \to B) \vdash A \to B$$

If we use this to model grammatical-composition, then we obtain the following type-shifting principle

$$S \to (S \to S) \vdash S \to S$$

which says that a two-place connective automatically converts into a one-place connective, which seems implausible.

2. Duplication

$$A \vdash A \times A$$

If we use this to model grammatical-composition, then we obtain the following grammatical principle.

---

[18] I hereby propose to name this principle after Jan Lukasiewicz (1878-1956), who invented multi-valued logic. In his system, one can define disjunction in terms of conditional via: $P \vee Q =_{df} (P \to Q) \to Q$.

$$D \vdash D \times D$$

This amounts to saying that a proper-noun phrase can duplicate itself (repeatedly) and accordingly serve as the input for an unlimited number of functors (e.g., VPs), which seems implausible.

3. Tautology

A tautology may be defined as a formula that can be shown using no premises whatsoever. The following is the simplest example.

$$\varnothing \vdash A \rightarrow A$$

From a grammatical point of view, this seems undesirable, since it allows phrases (types) to enter a tree "out of thin air".

4. Law of Assertion

$$(A \rightarrow A) \rightarrow B \vdash B$$

If we use this to model grammatical-composition, then we obtain the following grammatical principle, where C is the type of common-noun phrases.

$$(C \rightarrow C) \rightarrow (C \rightarrow C) \vdash C \rightarrow C$$

From this, we obtain the following composition principle.

$$(C \rightarrow C) \rightarrow (C \rightarrow C) \,;\, C \vdash C$$

A *modifier* is, by definition, a phrase of type $\mathfrak{I} \rightarrow \mathfrak{I}$. For example, a common-noun modifier (adjective) is a phrase of type $(C \rightarrow C)$, and an adjective-modifier is a phrase of type $(C \rightarrow C) \rightarrow (C \rightarrow C)$. According to the grammatical principle proposed above, an adjective-modifier like ‘very’ can be combined with a common noun like ‘dog’ to produce a common noun ‘very dog’. This is not plausible.

16. Linear Logic

By way of correcting these problems, we consider a still weaker logical system, Linear Logic (LL). The basic underlying idea is that, whereas Relevance Logic requires that every assumption is used *at least once*, Linear Logic requires that every assumption is used *exactly once*.

17. Hyper-Relevance Logic

Linear Logic gets rid of Contraction and Duplication, but it does not get rid of Assertion. To accomplish this, we additionally ban the empty index $\varnothing$, which amounts to banning arguments with no premises. This makes sense from a grammatical point of view, since we do not want phrases (types) entering a tree "out of thin air". We dub the resulting system ‘Hyper-Relevance Logic’ (HRL);[19] the further underlying idea is that, in reaching a conclusion, must use at least one premise.

---

[19] This is also known in the literature as "Linear Logic without Identity".

18. The Problem

By moving to HRL, we get rid of many inference principles that do not properly model grammatical-composition, but we also get rid of inference principles that seem desirable, including the following, which are all valid in RL.

(1) $A\to B \,;\, A\to C \vdash A\to(B\times C)$ [Conditional Multiplication]
(2) $A\to(B\to C) \,;\, A\to B \vdash A\to C$ [Conditional Modus Ponens]
(3) $A\to B \,;\, A\to C \,;\, (B\times C)\to D \vdash A\to D$ [Generalized Conjunction]

It seems that we are in a "Goldilocks situation" – we have one system, RL, which is too strong, and we have another system, HRL, which is too weak, but we seek a system that is "just right". In the next chapter, "Categorial Logic", we present the details of our proposed system, which we call CGL [short for 'categorial grammar logic'], alternatively G [short for 'grammar' and 'Goldilocks'].

19. Summary of Logical Systems Considered So Far

1. Examples of Arguments Valid in all Systems

(1) $B\to C \,;\, A\to B \vdash A\to C$ [Transitivity]
(2) $A\to(B\to C) \vdash B\to(A\to C)$ [Permutation]
(3) $A\to(B\to C) \,;\, B \vdash A\to C$ [Secondary *Modus Ponens*]
(4) $A \vdash (A\to B)\to B$ [Lifting]
(5) $(A\to C)\to D \,;\, A\to B \vdash (B\to C)\to D$ [Inflection]
(6) $(A\to C)\to D \,;\, A\to(B\to C) \vdash B\to D$ [Permutation + Transitivity]
(7) $B\to C \,;\, A\times B \vdash A\times C$ [Addition]
(8) $(A\times B)\to C \dashv\vdash A\to(B\to C)$ [Schönfinkel's Law]

2. CGL-Valid Arguments Rejected by LL

(1) $A\to(B\to C) \,;\, A\to B \vdash A\to C$ [Conditional Modus Ponens]
(2) $A\to B \,;\, A\to C \vdash A\to(B\times C)$ [Conditional Multiplication]
(3) $A\to B \,;\, A\to C \,;\, (B\times C)\to D \vdash A\to D$ [Generalized Conjunction]

3. LL-Valid Arguments Rejected by CGL

(1) $(A\to A)\to B \vdash B$ [Law of Assertion]

4. RL-Valid Arguments Rejected by LL

(1) $A\to(A\to B) \vdash A\to B$ [Contraction]
(2) $A \vdash A\times A$ [Duplication]

5. IL-Valid Arguments Rejected by RL

(1) $A \vdash B\to A$ [Positive Paradox]
(2) $A\to B \vdash A\to(A\to B)$ [Expansion]
(3) $A \vdash B\to B$ [Tautology]
(4) $A\times B \vdash A \,;\; A\times B \vdash B$ [Simplification]

6. CL-Valid Arguments Rejected by IL

(1) $(A\to B)\to B \vdash (B\to A)\to A$ [Lukasiewicz's Law]
(1) $(A\to B)\to A \vdash A$ [Peirce's Law]