TABLEAU METHODS
FOR SUBSTRUCTURAL LOGICS

1 INTRODUCTION

Over the last few decades a good deal of research in logic has been prompted by the realization that logical systems can be successfully employed to formalize and solve a variety of computational problems. Traditionally, the theoretical framework for most applications was assumed to be classical logic. However, this assumption often turned out to clash with researchers’ intuitions even in well-established areas of application. Let us consider, for example, what is probably the most representative of these application areas: logic programming. The idea that the execution of a Prolog program is to be understood as a derivation in classical logic has played a key role in the development of the area. This interpretation is the leitmotiv of Kowalski’s well-known [1979], whose aim is described as an attempt ‘to apply the traditional methods of [classical] logic to contemporary theories of problem solving and computer programming’. However, here are some quotations which are clearly in conflict with the received view (and with each other) as to the correct interpretation of logic programs:

Relevance Logic not only shows promise as a standard for modular reasoning systems, but it has, in a sense, been already adopted by artificial intelligence researchers. The resolution method for Horn clauses appears to be based on classical logic, but procedural derivation (see [Kowalski, 1979]), the method actually used for logic programming, is not complete for classical logic, and is in fact equivalent to Relevance Logic[…] the systems of modules which are actually used in computer science have the feature that relevance, not classical logic, provides a theory of their behaviour. [Garson, 1989, p.214]

According to the standard view, a logic program is a definite set of Horn clauses. Thus logic programs are regarded as syntactically restricted first-order theories within the framework of classical logic. Correspondingly, the proof-theory of logic programs is considered as a specialized version of classical resolution, known as SLD-resolution. This view, however, neglects the fact that a program clause $a_0 \leftarrow a_1, a_2, \ldots, a_n$ is an expression of a fragment of positive logic [a subsystem of Intuitionistic Logic] rather than an implication formula of classical logic. The logical behaviour of such clauses is in no way related to any negation or complement operation. So (positive) logic programs are ‘sub-classical’. The classical interpretation seems to be a semantical overkill’ [Wagner, 1991, p.835].

This is just an example of a general phenomenon which arises in the application of logic: we start from a logic $L$ (e.g. classical logic) and develop a deduction system
for $L$. Then, in order to adapt this system to a given application problem, we often introduce ad hoc procedural ‘perturbations’ which restrict the original logic $L$ and turn it into another logic $L^*$. Now we have two alternatives. We may regard $L^*$ as the result of application-dependent constraints which belong to the ‘pragmatics’ of $L$. If we adopt this conservative option, the behaviour of $L^*$ belongs to the domain of applied logic, the effects of the perturbations and the overall behaviour of the perturbed system can be regarded as typical engineering problems. Alternatively, instead of imposing some theoretical system borrowed from the literature on pure logic, we may decide to recognize $L^*$ as a first-class citizen, i.e. a logical system in its own sake, which can become an independent object of investigation. In this way the behaviour of $L^*$ becomes a new theoretical problem and belongs to the domain of pure logic. We can then try to provide a theoretical characterization of the new system by exploiting its analogy with more familiar ones (for instance, we can axiomatize $L^*$, produce Gentzen-style proof systems, develop algebraic and relational semantics for it).

In this chapter we shall focus on an important family of logical systems, which arise from ‘perturbations’ of classical and intuitionistic logic and are known as substructural or resource logics. Historically, the subject of substructural logics arises from the combination of four main components:

- the tradition of Relevance Logic (Anderson, Belnap, Meyer);
- the work on BCK Logic (Fitch, Nelson, Meredith, Prior);
- the work on Categorial Logic, (Curry, Howard, Lambek, Van Benthem);
- the recent developments in Linear Logic, (Girard, Lafont).

We shall not attempt here to provide an introduction to the subject and shall assume the reader to be familiar at least with the basic ideas involved. The reader is referred to [Dosen, 1993] for the necessary historical and conceptual background. We shall, however, try and give a flavour of the subject by briefly discussing its impact on the traditional notion of logical deduction.

### 1.1 Background on substructural logics

Substructural logics ultimately stem from Gentzen’s *Investigations into Logical Deduction* [Gentzen, 1935], and in particular from his characterization of deductive inference in terms of calculi of sequents. Such calculi can be seen as axiomatizations of the notion of logical consequence, expressed by the turnstile $\vdash$, which characterize classical and intuitionistic inferences. Gentzen’s analysis brought to light that the properties of $\vdash$ fall in two categories:

- Properties characterizing the behaviour of the logical operators $\rightarrow, \land, \lor, \neg$, fixed by the operational rules.
Properties pertaining to the interpretation of $\vdash$, fixed by the structural rules (including the cut rule).

Gentzen’s approach already contained the possibility of heuristic variations. The operational rules of his sequent calculus can be seen as expressing the meaning of the logical operators, while the structural properties of the turnstile as expressing the properties of the underlying information system, i.e. the rules that govern the information-processing mechanism. From the viewpoint of the present chapter, this heuristic is central. Many perturbations of classical and intuitionistic logic, prompted by their interaction with computational, philosophical or linguistic problems, are captured by re-interpreting the relation $\vdash$, via variations of its structural properties.

Such ‘perturbed’ systems may have a decent proof-theory and even an intuitive semantics, but can we call them ‘logics’? An answer to this question depends, of course, on our definition of ‘logic’. Very seldom definitions are completely stable, and this is no exception. A ‘logic’ is usually identified with a consequence relation, i.e. a binary relation formalizing the intuitive notion of logical consequence. According to the traditional definition, which was first formulated explicitly by Tarski [1930a; 1930b], a consequence relation is a relation $\vdash$ between sets of formulae and formulae satisfying the following conditions:

(Identity) \[ A \vdash A \]

(Monotonicity) \[ \Gamma \vdash B \Rightarrow \Gamma, A \vdash B \]

(Transitivity) \[ \Gamma, A \vdash B \text{ and } \Delta \vdash A \Rightarrow \Gamma, \Delta \vdash B \]

where $\Gamma$ and $\Delta$ are sets of formulae (we write, as usual, ‘$A$’ instead of ‘$\{A\}$’ and ‘$\Gamma, A$’ instead of ‘$\Gamma \cup \{A\}$’).

For instance, the system of intuitionistic implication can be shown to correspond to the smallest consequence relation closed under the following additional condition concerning the $\rightarrow$ operator:

(Cond) \[ \Gamma, A \vdash B \iff \Gamma \vdash A \rightarrow B \]

The closure conditions in the definition of consequence relation are all structural conditions, i.e. they do not involve any specific logical operator, but express basic properties of the notion of logical consequence. The emergence of Relevance Logic (see the monumental [Anderson and Belnap, 1975] and [Anderson et al., 1992]; for an overview see [Dunn, 1986]) showed the inadequacy of this traditional definition. If the system $R$ proposed by Anderson and Belnap had to be called ‘logic’, the definition had to be amended. Let’s see why.

\[1\]This terminology goes back to Gentzen [1935] and his distinction between structural and operational rules in the sequent calculi.
The whole idea of Relevance Logic is that in a ‘proper’ deduction all the premises have to be used at least once to establish the conclusion, so as to stop the validity of the notorious ‘fallacies of relevance’ such as the so-called ‘positive paradox’ \( A \rightarrow (B \rightarrow A) \). This criterion of use is ultimately sufficient to prevent the fallacies from being provable. For instance, in the typical natural deduction proof of the positive paradox the assumption \( B \) is discharged ‘vacuously’ by the application of the \( \rightarrow \)-introduction rule, i.e. it is not used in obtaining the conclusion of the subproof constituting the premiss of the rule application. If we restrict our notion of proof in such a way that all the assumptions have to be used in order to obtain the conclusion of the proof, such ‘vacuous’ applications of the \( \rightarrow \)-introduction rule are not allowed, since the subproof which occurs as premiss would not be a ‘proper proof’. So, the standard proof is no longer an acceptable proof of the positive paradox, and it can be shown that no alternative proofs can be found.

The criterion of use is clearly of a ‘metalevel’ nature. It takes the form of a side-condition on the application of the natural deduction rules. Let us consider the restricted deducibility relation \( \vdash^{ND}_{\mathbf{R}} \) which incorporates this side-condition. Is it a consequence relation in the traditional sense? The answer is obviously ‘no’, because it does not satisfy (Monotonicity), in that this condition would allow the addition of ‘irrelevant’ assumptions which are not used in deducing the conclusion. So, if we want to consider the system of relevant implication as a logic, we have to drop (Monotonicity) from our definition of a consequence relation.

But this is not the whole story. In the system \( \mathbf{R} \), the definition of a ‘relevant’ deduction requires that every single occurrence of an assumption is used to obtain the conclusion. Now, let us write \( \Gamma \vdash_{\mathbf{R}} A \), where \( \Gamma \) is a finite sequence of formulae, to mean that there is an \( \mathbf{R} \)-deduction of \( A \) using all the elements of \( \Gamma \) (which are occurrences of formulae). Consider the statement \( A, A \vdash_{\mathbf{R}} A \). This is not provable because there is no way of using both occurrences of \( A \) in the antecedent in order to obtain the conclusion, i.e. one of these two occurrences is ‘irrelevant’. Therefore, while \( A \vdash_{\mathbf{R}} A \) is trivially provable, \( A, A \vdash_{\mathbf{R}} A \) is not. It becomes provable if we ‘dilute’ the criterion of use to the effect that at least one occurrence of each assumption needs to be used, as in the ‘mingle’ system (see [Dunn, 1986]). The trouble is that the distinction between these two different approaches to the notion of relevance cannot be expressed by the traditional notion of consequence relation. Indeed, according to this notion, a consequence relation is taken to be a relation between sets of formulae and formulae. But there is no way to distinguish the set \( \{ A, A \} \) from the set \( \{ A \} \) and, therefore, between \( A, A \vdash_{\mathbf{R}} A \) and \( A \vdash_{\mathbf{R}} A \). To make this sort of distinction we need a finer-grained notion of consequence relation.

In fact, such a finer-grained notion was already contained in Gentzen’s calculus of sequents [Gentzen, 1935]. In this approach, a (single-conclusion\(^2\)) consequence

---

\(^2\)Gentzen also considered multiple-conclusion consequence relations and showed that this more general notion was more convenient for the formalization of classical logic. See [Gentzen, 1935].
relation is axiomatized as a relation between a finite sequence of formulae and a formula. That such a relation holds between a sequence \( \Gamma \) and a formula \( A \) is expressed by the sequent \( \Gamma \vdash A \) (Gentzen used the notation \( \Gamma \implies A \), but we prefer the ‘turnstile’ notation for reasons of uniformity). The axioms of Gentzen’s system are given by all the sequents of the form \( A \vdash A \). Gentzen specified also two sets of rules to derive new sequents from given ones, that he called operational rules and structural rules. While the first type of rules embodied, in his view, the meaning of the logical operators, the second type embodied the meaning of \( \vdash \). These structural rules are the following:

- **Weakening**
  \[
  \Gamma, \Delta \vdash B \\
  \Gamma, A, \Delta \vdash B
  \]

- **Exchange**
  \[
  \Gamma, A, B, \Delta \vdash C \\
  \Gamma, B, A, \Delta \vdash C
  \]

- **Contraction**
  \[
  \Gamma, A, A, \Delta \vdash B \\
  \Gamma, A, \Delta \vdash B
  \]

- **Cut**
  \[
  \Gamma \vdash A \\
  \Delta, \Lambda, \Lambda \vdash B \\
  \Delta, \Gamma, \Lambda \vdash B
  \]

Here the Weakening rule corresponds to the Monotonicity condition of a consequence relation, and Cut corresponds to Transitivity, except that, in the sequent formulation, the structure on the left of \( \vdash \) is a sequence, rather than a set, of formulae. Moreover, in Gentzen’s system the role of the Identity condition is played by the assumption that every axiom \( A \vdash A \) can be used at any step of a sequent proof. The remaining rules of Contraction and Exchange have the effect of making Gentzen’s relations \( \vdash \) deductively equivalent to the corresponding consequence relations with sets as first argument.

Gentzen’s richer axiomatization provides the means of characterizing systems like \( R \) as ‘substructural’ consequence relations, i.e. logics for which the standard structural rules of Gentzen’s axiomatization may not hold. In the case of \( R \), the rule which is dropped is the Weakening rule, responsible for the arbitrary introduction of ‘irrelevant’ items in the antecedent of a sequent. After removing Weakening, one can consider a rule symmetric to the Contraction rule:

- **Expansion**
  \[
  \Gamma, A, \Delta \vdash B \\
  \Gamma, A, A, \Delta \vdash B
  \]

The more ‘liberal’ mingle system is then distinguished from \( R \) by the fact that it allows for this weaker version of the Weakening rule.
The discussion of Relevance Logic clearly brings out the general idea that variations in the notion of logical consequence correspond to variations in the allowed structural rules of a suitable sequent-based system, leaving the basic operational rules unchanged. With Girard’s Linear Logic [Girard, 1987; Avron, 1988a] the ‘substructural movement’ reached its climax. Linear Logic completely rejects the ‘vagueness’ of traditional proof-theory concerning the use and manipulation of assumptions in a deductive process. A ‘proper’ proof is one in which every assumption is used exactly once. If a particular assumption \( \Delta \) can be used \textit{ad libitum}, this has to be made explicit by prefixing it with the ‘storage’ operator \( ! \). This means that the Contraction rule is not sound in Linear Logic, since it says, informally, that a proof of \( B \) from two occurrences of \( \Delta \) can be turned into a proof of \( B \) from one occurrence only of \( \Delta \). But this is impossible, unless \( \Delta \) is one of those assumptions which can be used \textit{ad libitum}, in which case we should prefix it with the storage operator. In the ‘non-commutative’ variant of Linear Logic — which was anticipated in 1958 by Lambek [1958] as a system intended for applications to mathematical linguistics (see Abrusci [1990; 1991] and [van Benthem, 1991] for further developments — also the order in which assumptions are used becomes crucial, and therefore the Exchange rule is also disallowed.

In this chapter we shall focus on tableau methods for substructural logics and shall discuss two main lines of research: the approach based on ‘proof-theoretic’ tableaux developed by McRobbie, Belnap and Meyer, which is motivated by the work done in the tradition of Relevance Logic; and the approach based on ‘labelled tableaux’, an outgrowth of Gabbay’s research program on Labelled Deductive Systems.

### 1.2 Substructural Consequence Relations

In this chapter we take a consequence relation as a relation \( \vdash \) between sequences of formulae and formulae, satisfying:

- **Identity**
  \[ A \vdash A \]

- **Surgical Cut**
  \[
  \Gamma \vdash A, \Delta, A \Delta \vdash B \\
  \Delta, \Gamma, A \Delta \vdash B
  \]

Since the \( \Gamma, \Delta \) range over sequences, an application of the cut rule replaces an occurrence of the formula \( A \) with the sequence \( \Gamma \) exactly in the position of \( A \). This is why we call the cut ‘surgical’.

**Structural rules.** Apart from the cut rule, which is part of their definition, consequence relations may or may not obey any of the following conditions describing their structural properties:
We can think of a sequent $\Gamma \vdash A$ as stating that the piece of information expressed by $A$ is ‘contained’ in the data structure expressed by $\Gamma$. Then different combinations of the above structural rules can be seen as different ways of defining the properties of the ‘information flow’ expressed by the turnstile, depending on the structure of the data and on the allowed ways of manipulating it. For instance, if we disallow the Weakening rule, the consequence relation becomes sensitive to the relevance of the data to the conclusion: all the data has to be used in the derivation process. If also Expansion is disallowed, this notion of relevance extends to the single occurrences of the formulae in the data (as in Anderson and Belnap’s system of Relevance Logic, each occurrence of a formula in the data has to be used). If Contraction is disallowed, each item of data can be used only once and has to be replicated if it is used more than once (as in Girard’s Linear Logic and its satellites). In this way the process of deriving a formula becomes more similar to a physical process and the consequence relation becomes sensitive to the resources employed. Finally, if Exchange is disallowed, the ‘chronology’ of this process — i.e. the order in which formulae occur in the data — becomes significant (as in the Lambek calculus).

Notice that if Exchange is allowed, the antecedent of a sequent can be regarded as a multiset.\(^3\) If Contraction and Expansion are also allowed (notice that the Weakening rule implies the Expansion rule) the antecedent of a sequent can be regarded as a set of formulae, as with the standard Gentzen systems, i.e. the number of occurrences of formulae does not count, nor does the order in which they occur.

The operator $\rightarrow$. The conditional operator, or ‘implication’ as is often improperly called, is usually characterized by the following equivalence:

\[ C \rightarrow \quad \Gamma \vdash A \rightarrow B \iff \Gamma, A \vdash B \]

which, under the assumption of Identity and Surgical cut is equivalent to the pair of sequent rules:

\[ (1) \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \quad \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Delta, A \rightarrow B, \Gamma \vdash C} \]

\(^3\)For multiset-based consequence relations, the reader is referred to [Avron, 1991].
For instance, we can derive the only-if part of $C \rightarrow$ from the Gentzen rules (the if-part is a primitive rule) as follows:

\[
\frac{A \vdash A \quad B \vdash B}{\Gamma \vdash A \rightarrow B \quad A \rightarrow B, A \vdash B} \quad \text{[CUT]}
\]

Notice that if no structural rule, except the cut rule, is to be used in the derivation, the shape of the latter depends crucially on the format of the operational rules and of the cut rule. Notice also that in systems which do not allow Exchange the conditional operator splits into two operators defined as follows:

\[
\begin{align*}
C_{\rightarrow_1} & \quad \Gamma, A \vdash B \iff \Gamma \vdash A \rightarrow_1 B \\
C_{\rightarrow_2} & \quad A, \Gamma \vdash B \iff \Gamma \vdash A \rightarrow_2 B.
\end{align*}
\]

Of course, if Exchange is allowed $\rightarrow_1$ and $\rightarrow_2$ collapse into each other. In this context we shall use the symbol $\rightarrow$ without subscripts.

The operators $\otimes$ and $\wedge$. In the classical and intuitionistic sequent calculus the comma occurring in the left-hand side of a sequent is associated with classical conjunction. This operator represents a particular way of combining pieces of information. Its inferential role depends on two components: the operational rules defining its meaning, and the structural rules defining the meaning of the turnstile. In the new setting, in which the only conditions which are required to hold are Identity and Surgical cut, the comma is no longer equivalent to classical conjunction. However, the new consequence relations may still be closed under the standard condition defining classical conjunction. Such condition characterizes a new type of ‘substructural’ conjunction, denoted by $\otimes$ and sometimes called ‘tensor’ which is no longer classical (because of the failure of some or all of the structural rules), and yet is defined in the same way as its classical version. Therefore, in a sense, it has the same ‘meaning’, namely:

\[
C_{\otimes} \quad \Gamma, A \otimes B, \Delta \vdash C \iff \Gamma, A, B, \Delta \vdash C
\]

Clearly, a sequent $\Gamma \vdash A$ becomes equivalent to $\otimes \Gamma \vdash A$ where $\otimes \Gamma$ denotes the $\otimes$-concatenation of the formulae in $\Gamma$.

Given Identity and Surgical cut, $C_{\otimes}$ can be easily shown to be equivalent to the pair of sequent rules:

\[
\begin{align*}
\text{(2)} & \quad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} & \quad \frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, A \otimes B, \Delta \vdash C}
\end{align*}
\]

\footnote{For the time being we are considering only single-conclusion sequents. Later on we shall consider also multi-conclusion sequents in the context of logical systems with classical (involutive) negation.}
It is not difficult to see that $C_\otimes$ together with Identity and Surgical Cut, implies:

$$
\frac{A \vdash B \quad C \vdash D}{A \otimes C \vdash B \otimes D}
$$

Moreover, let us consider the following restricted form of cut:

Transitivity

$$
\frac{A \vdash B \quad B \vdash C}{A \vdash C}
$$

Then, it is easy to see that Transitivity, together with Identity, $C_\otimes$ and (3), imply Surgical Cut.

The rules for $\otimes$, as well as its definition $C_\otimes$, describe a type of conjunction which is sometimes called ‘context-free’ since its characterization does not impose any special condition on the ‘context’, i.e. on the structures of side formulae (the $\Gamma$ and $\Delta$). A ‘context-dependent’ type of conjunction is expressed by the following equivalence:

$$
C_\wedge \quad \Gamma \vdash A \wedge B \iff \Gamma \vdash A \text{ and } \Gamma \vdash B
$$

which corresponds to the following sequent rules:

$$
\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \quad \frac{\Gamma, A, \Delta \vdash C}{\Gamma, A \wedge B, \Delta \vdash C} \quad \frac{\Gamma, B, \Delta \vdash C}{\Gamma, A \wedge B, \Delta \vdash C}
$$

Here, in the two-premiss rule, the premises must share the same structure of side formulae, that is the two proofs of $A$ and $B$ must depend exactly on the same structure of assumptions, namely $\Gamma$. The sequent $\Gamma \vdash A \wedge B$ expresses the fact that from this $\Gamma$ we can derive either $A$ or $B$ at our choice. Of course, if we allowed Contraction, we could derive both $A$ and $B$ and then ‘shrink’ the two copies of $\Gamma$ used in this derivation into one. If we allowed Weakening, on the other hand, we could derive $A$ and $B$ from different data-structures, say $\Gamma$ and $\Delta$ and then expand both to $\Gamma, \Delta$ so as to satisfy the condition of the rule. Thus, under the assumption of Weakening and Contraction, $C_\wedge$ and the corresponding sequent rules would not define any new operator different from $\otimes$. If either of these rules is disallowed, they do define a new operator which is called, in Girard’s terminology, additive conjunction.

The operator $\neg$. Negation can be defined à la Johansson, in terms of implication and the ‘falsum’ constant $\bot$. In this approach $\neg A$ is defined as $A \rightarrow \bot$. Then the condition $C_{\neg}$ above immediately yields:

$$
C_{\neg} \quad \Gamma, A \vdash \bot \iff \Gamma \vdash \neg A.
$$

Again, it is easy to see that, given Identity and Surgical cut, the condition $C_{\neg}$ is equivalent to the following pair of sequent rules:
The disjunction operator can be defined by the following equivalence:

\[ C \iff \Gamma, A \vdash B \iff \Gamma, A \vdash C \text{ and } \Gamma, B \vdash C \]

which corresponds to the following sequent rules:

\[
\begin{align*}
\Gamma, A \vdash C & \quad \Gamma, B \vdash C \\
\Gamma, A \lor B \vdash C & \\
\Gamma \vdash A & \\
\Gamma \vdash B &
\end{align*}
\]

Notice that the disjunction operator defined by these rules is ‘context-dependent’ in a sense similar to the sense in which \( \land \) is: a crucial condition in the two-premiss rule is that the context of the inference step, namely the structure of side formulae \( \Gamma \), is the same in both premisses. This type of disjunction is called ‘additive’ to distinguish it from a context-free type of disjunction that arises when we consider logics with an involutive negation operator (see below).

Involutive negation is characterized by the additional condition expressed by the ‘double negation law’:

\[ \neg\neg A \vdash A \]

which, given \( C_\lor \) or equivalently the sequent rules in (5) – Identity and Surgical Cut, is equivalent to the rule:

\[
\begin{align*}
\Gamma, \neg A \vdash \bot & \\
\Gamma \vdash A &
\end{align*}
\]

Let us now define a binary operator \( \boxtimes \) as follows:

\[ A \boxtimes B \equiv \neg A \rightarrow B \]

If (7) holds and Exchange is allowed, \( \boxtimes \) is commutative and associative. Moreover, the following equivalences also hold:

\[
\begin{align*}
\Gamma, A \vdash B & \iff \Gamma \vdash \neg A \boxtimes B \\
\Gamma, \neg A \vdash B & \iff \Gamma \vdash A \boxtimes B
\end{align*}
\]

Thus, in the logics satisfying the Exchange property and the double negation law (7) we can naturally introduce multi-conclusion sequents of the form

\[ \Gamma \vdash \Delta \]

where \( \Delta \) is (like \( \Gamma \)) a list of formulae, with the intended meaning \( \otimes \Gamma \vdash \boxtimes \Delta \) (by \( \otimes \Gamma \) and \( \boxtimes \Delta \) we denote, respectively, the \( \otimes \)-concatenation of the formulae in \( \Gamma \) and
the $\exists$-concatenation of the formulae in $\Delta$). The properties of the (classical) negation operator allow us to translate back and forth from the single-conclusion formulation to the multi-conclusion one. The operator $\exists$ corresponds to the comma in the right-hand side of a multi-conclusion sequent, just as the operator $\otimes$ corresponds to the comma in the left-hand side. Under these circumstances the following equivalences can also be easily shown:

(11) $A \otimes B \Vdash \neg(A \to \neg B)$

(12) $A \otimes B \Vdash \neg(\neg A \exists B)$

When all the usual structural rules are allowed, $\exists$ is clearly equivalent to classical disjunction, just as $\otimes$ is equivalent to classical conjunction.

Notice that in the logics satisfying (7) and Exchange both $\otimes$ and $\exists$ are commutative, so the antecedent and the succedent of a sequent can be regarded as multisets rather than just sequences, and Surgical Cut can be replaced by the more familiar (though not necessarily more natural):

\[
\Gamma \vdash A, \Delta \quad \Lambda, A \vdash \Pi \\
\Gamma, A \vdash \Delta, \Pi
\]

The reader can easily derive multi-conclusion versions of $C_{\otimes}, C_{\rightarrow}$ and $C_{\rightarrow}$, and the corresponding sequent rules. Clearly $\exists$ can be defined directly by the following condition:

\[
C_{\exists} \quad \Gamma \vdash \Delta, A, B, \Lambda \iff \Gamma \vdash \Delta, A \exists B, \Lambda
\]

This condition is equivalent to the following pair of rules:

\[
\begin{align*}
\Gamma, A &\vdash \Delta \quad B, \Lambda \vdash \Pi \\
\Gamma, A \exists B, \Lambda &\vdash \Delta, \Pi \\
\Gamma &\vdash \Delta, A, B, \Lambda \\
\Gamma &\vdash \Delta, A \exists B, \Lambda
\end{align*}
\]

which bring up the ‘context-free’ character of this type of disjunction as opposed to the context-dependent character of $\lor$.

We conclude this section with a terminological remark. Following Girard [1987] the logical operators we have been considering so far can be classified into two categories, the multiplicatives and the additives. The multiplicative operators are the ‘context-free’ ones, i.e. those which can be defined via rules of inference with no conditions on the context of the inference step (see [Avron, 1988b] for this characterization), and include $\rightarrow, \neg, \otimes$ and $\exists$. The ‘additive’ operators, are the context-dependent conjunction and disjunction $\land$ and $\lor$.

2 PROOF-THEORETIC TABLEAUX

An early example of tableau methods for substructural logics is Dunn’s method of ‘coupled trees’ for first degree entailment, described in Fiutting’s introduction.