Chapter 1

Introduction

In mathematics, one sometimes lives under the illusion that there is just one logic that formalizes the correct principles of mathematical reasoning, the so-called *predicate calculus* or *classical first-order logic*. By contrast, in philosophy and computer science, one finds the opposite: there is a vast array of logics for reasoning in a variety of domains. We mention intuitionistic logic, sorted logic, modal logic, description logic, temporal logic, belief logic, dynamic logic, Hoare logic, specification logic, evaluation logic, relevance logic, higher-order logic, non-monotonic logic, bunched logic, non-commutative logic, affine logic, and, yes, linear logic. Many of these come in a variety of flavors.

There are several reasons for these differing views on logic. An important reason is that in mathematics we use logic only in principle, while in computer science we are interested in using logic in practice. For example, we can eliminate sorts from predicate logic by translating them to predicates and relativizing quantifiers. For example, $\forall x{:}s.\ A(x)$ can be reformulated as $\forall x.\ S(x) \supset A(x)$. This means, in principle, we do not have to bother with sorts when studying logic. On the other hand, practical reasoning with formulas after sorts have been eliminated is much more complex than before. An intrinisic property of an object (t has sort s) has now become a proof obligation ($S(t)$ is true). As a result, some theorem provers such as SPASS instead apply essentially the opposite transformation, translating monadic predicates into sorts before or during the theorem proving process.

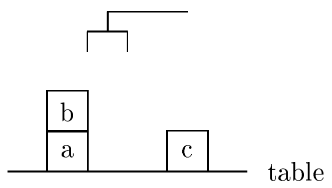
Another important difference between the mathematical and computational point of view lies in the conceptual dependency between the notions of proof and truth. In traditional mathematics we are used to thinking of "truth" as existing abstractly, independently of anyone "knowing" the truth or falsehood of a proposition. Proofs are there to demonstrate truth, but truth is really independent of proof. In computer science, however, we have to be concerned with computation. Proofs in this context show how to construct (= compute) objects whose existence is asserted in a proposition. This means that the notions of construction and proof come before the notion of truth. For example, $\exists x.\ A(x)$ is true if we can construct a t such that $A(t)$ is true. Implication is another example, where $A \supset B$ is true if we can construct a proof of B from a proof of A.

Our approach to linear logic is strongly influenced by both of these points. First, we identify an important problem domain, namely *reasoning with state*, that can be translated into the predicate calculus only with a great deal of coding which makes simple situations appear complex. Second, we develop an appropriate logic *constructively*. This means we explain the meaning of the connectives via their proof rules, and not by an external mathematical semantics. This is both philosophically sound and pragmatically sufficient to understand a logic and how to use it.

Before we launch into examples and informal description of linear logic, we should point out that our perspective is neither historical (linear logic instead arose from domain theory) nor the most popular (much of the current work on linear logic accepts the non-constructive law of excluded middle). On the other hand, we believe our intuitionistic view of linear logic has its own compelling beauty, simplicity, and inevitability, following the tradition of Gentzen [Gen35], Prawitz [Pra65], and Martin-Löf [ML96]. Furthermore, intuitionistic linear logic can directly accomodate most applications that classical linear logic can, but not vice versa.

The interested reader is referred to the original paper by Girard [Gir87], and several surveys [Lin92, Sce93, Tro92] for other views on linear logic. A historical introduction [Doš93] and context for linear and other so-called substructural logics outside computer science can be found in [SHD93].

As a motivating example for linear logic we consider the so-called *blocks world*, which is often used to illustrate planning problems in artificial intelligence. It consists of various blocks stacked on a table and a robot arm that is capable of picking up and putting down one block at a time. We are usually given an initial configuration and some goal to achieve. The diagram below shows typical situation.



We would like to describe this situation, the legal moves, and the problem of achieving a particular goal in logical form. This example led to an independent discovery of a fragment of linear logic by Bibel [Bib86] around the same time that Girard developed linear logic based on a very different foundations.

on(x, y)	block x is on block y
tb(x)	block x is on the table
holds(x)	robot arm holds block x
empty	robot arm is empty
clear(x)	the top of block x is clear

A state is described by a collection of propositions that are true. For example, the state above would be described as

$$\Delta_0 = (\mathrm{empty}, \mathrm{tb}(a), \mathrm{on}(b, a), \mathrm{clear}(b), \mathrm{tb}(c), \mathrm{clear}(c))$$

A goal to be achieved can also be described as a logical proposition such as on(a, b). We would like to develop a logical system so that we can prove a goal G from some assumptions Δ if and only if the goal G can be achieved from the initial state Δ. In this kind of representation, plans correspond to proofs. The immediate problem is how to describe legal moves. Consider the following description:

> *If the robot hand is empty, a block x is clear, and x is on y, then we can pick up the block, that is, achieve a state where the robot hand holds x and y is clear.*

One may be tempted to formulate this as a logical implication.

$$\forall x.\, \forall y.\, (\mathrm{empty} \wedge \mathrm{clear}(x) \wedge \mathrm{on}(x, y)) \supset (\mathrm{holds}(x) \wedge \mathrm{clear}(y))$$

However, this encoding is incorrect. With this axiom we can derive contradictory propositions such as empty$\wedge$holds(b). The problem is clear: logical assumptions persist. In other words, ordinary predicate calculus has no notion of state.

One can try to solve this problem in a number of ways. One way is to introduce a notion of time. If we $\bigcirc A$ to denote the truth of A at the next time, then we might say

$$\forall x.\, \forall y.\, (\mathrm{empty} \wedge \mathrm{clear}(x) \wedge \mathrm{on}(x, y)) \supset \bigcirc(\mathrm{holds}(x) \wedge \mathrm{clear}(y))$$

Now the problem above has been solved, since propositions such as empty $\wedge$ $\bigcirc$holds(b) are not contradictory. However, we now have the opposite problem: we have not expressed that "everything else" stays the same when we pick up a block. Expressing this in temporal logic is possible, but cumbersome. At heart, the problem is that we don't really need a logic of time, but a logic of state.

Miraculously, this is quite easy to achieve by changing our rules on how assumptions may be used. We write

$$A_1\ \mathit{true}, \ldots, A_n\ \mathit{true} \Vdash C\ \mathit{true}$$

to denote that we can prove C from assumptions $A_1, \ldots, A_n$, using every assumption *exactly once*. Another reading of this judgment is:

If we had resources $A_1, \ldots, A_n$ we could achieve goal C.

We refer to the judgment above as a *linear hypothetical judgment*. The order in which assumptions are presented is irrelevant, so we freely allow them to be exchanged. We use the letter Δ to range over a collection of linear assumptions.

From our point of view, the reinterpretation of logical assumptions as consumable resources is the central insight in linear logic from which all else follows in a systematic fashion. Such a seemingly small change has major consequences in properties of the logic and its logical connectives. First, we consider the laws that are derived from the nature of the linear hypothetical judgment itself, without regard to any logical connectives. The first expresses that if we have a resource A we can achieve goal A.

$$\frac{}{A\ \mathit{true} \Vdash A\ \mathit{true}}\,\mathsf{hyp}$$

Note that there may not be any leftover resources, since all resources must be used exactly once. The second law in some sense defines the meaning of linear hypothetical judgments.

If $\Delta \Vdash A$ true and Δ', A true $\Vdash C$ true then $\Delta, \Delta' \Vdash C$ true.

Informally: if we know how to achieve goal A from Δ, and if we know how to achieve C from A and Δ', then we can achieve C if we have both collections of resources, Δ and Δ'. We write Δ, Δ' as concatentation of the resources. This law is called a *substitution principle*, since it allows us to substitute a proof of A *true* for uses of the assumption A *true* in another deduction. The substitution principle does not need to be assumed as a primitive rule of inference. Instead, we want to assure that whenever we can derive the first two judgments, we can already derive the third directly. This expresses that our logical laws have not violated the basic interpretation of the linear hypothetical judgment: we can never obtain more from a resource A than is allowable by our understanding of the linear hypothetical judgment.

Next we introduce a few connectives, considering each in turn.

Simultaneous Conjunction. We write $A \otimes B$ if A and B are true in the same state. For example, we should be able to prove $A\ \mathit{true}, B\ \mathit{true} \Vdash A \otimes B\ \mathit{true}$. The rule for infering a simultaneous conjunction reads

$$\frac{\Delta \Vdash A\ \mathit{true} \qquad \Delta' \Vdash B\ \mathit{true}}{\Delta, \Delta' \Vdash A \otimes B\ \mathit{true}}\,\otimes I$$

Read from the conclusion to the premises:

In order to achieve goal $A \otimes B$ we divide our resources into Δ and Δ' and show how to achieve A using Δ and B using Δ'.

This is called an *introduction rule*, since it introduce a logical connective in the conclusion. An introduction rule explains the meaning of a connective by explaining how to achieve it as a goal. Conversely, we should also specify how to use our knowledge that we can achieve $A \otimes B$. This is specified in the *elimination rule.*

$$\frac{\Delta \Vdash A \otimes B\ \mathit{true} \qquad \Delta', A\ \mathit{true}, B\ \mathit{true} \Vdash C\ \mathit{true}}{\Delta, \Delta' \Vdash C\ \mathit{true}} \otimes E$$

We read an elimination rule downward, from the premise to the conclusion:

> *If we know that we can achieve $A \otimes B$ from Δ, we can proceed as if we had both A and B together with some other resources Δ'. Whatever goal C we can achieve form these resources, we can achieve with the joint resources Δ and Δ'.*

Intuitively, it should be clear that this is sound from the meaning of linear hypothetical judgments explained above and summarized in the substitution principle. We will see later more formally how to check that introduction and elimination rules for a connective fit together correctly.

Alternative Conjunction. We write $A \& B$ if we can goals A and B with the current resources, but only *alternatively*. For example, if we have one dollar, we can buy a cup of tea or we can buy a cup of coffee, but we cannot buy them both at the same time. For this reason this is also called *internal choice*. Do not confuse this with disjunction or "exclusive or", the way we often do in natural language! A logical disjunction (also called *external choice*) would correspond to a vending machine that promises to give you tea or coffee, but you cannot choose between them.

The introduction rule for alternative conjunction appears to duplicate the resources.

$$\frac{\Delta \Vdash A\ \mathit{true} \qquad \Delta \Vdash B\ \mathit{true}}{\Delta \Vdash A \& B\ \mathit{true}} \& I$$

However, this is an illusion: since we will actually have to make a choice between A and B, we will only need one copy of the resources. That we are making an internal choice is also apparent in the elimination rules. If we know how to achieve $A \& B$ we but we have to choose between two rules to obtain either A or B.

$$\frac{\Delta \Vdash A \& B\ \mathit{true}}{\Delta \Vdash A\ \mathit{true}} \& E_L \qquad \frac{\Delta \Vdash A \& B\ \mathit{true}}{\Delta \Vdash B\ \mathit{true}} \& E_R$$

Note that we do not use alternative conjunction directly in the blocks world example.

Linear Implication. For our blocks world example, we also need a form of implication: if we had resource A we could achieve B. This is written as $A \multimap B$. It expresses the meaning of the linear hypothetical judgment as a proposition.

$$\frac{\Delta, A\ \mathit{true} \Vdash B\ \mathit{true}}{\Delta \Vdash A \multimap B\ \mathit{true}} \multimap I$$

The elimination rule for $A \multimap B$ allows us to conclude that B can be achieved, if we can achieve A.

$$\frac{\Delta \Vdash A \multimap B\ \mathit{true} \qquad \Delta' \Vdash A\ \mathit{true}}{\Delta, \Delta' \Vdash B\ \mathit{true}} \multimap E$$

Note that we need to join the resources, which should be clear from our intuitive understanding of assumptions as resources.

Without formalizing it, we also assume that we have a universal quantifier with its usual logical meaning. Then we can express the legal moves in the blocks world with the following axioms:

$$\begin{array}{lcl}
\mathrm{geton} & : & \forall x.\, \forall y.\, \mathrm{empty} \otimes \mathrm{clear}(x) \otimes \mathrm{on}(x,y) \multimap \mathrm{holds}(x) \otimes \mathrm{clear}(y), \\
\mathrm{gettb} & : & \forall x.\, \mathrm{empty} \otimes \mathrm{clear}(x) \otimes \mathrm{tb}(x) \multimap \mathrm{holds}(x), \\
\mathrm{puton} & : & \forall x.\, \forall y.\, \mathrm{holds}(x) \otimes \mathrm{clear}(y) \multimap \mathrm{empty} \otimes \mathrm{on}(x,y) \otimes \mathrm{clear}(x), \\
\mathrm{puttb} & : & \forall x.\, \mathrm{holds}(x) \multimap \mathrm{empty} \otimes \mathrm{tb}(x) \otimes \mathrm{clear}(x).
\end{array}$$

Each of these represents a particular possible action, assuming that it can be carried out successfully. Matching the left-hand side of one these rules will consume the corresponding resources so that, for example, the proposition *empty* with no longer be available after the *geton* action has been applied.

For a given state $\Delta = A_1, \ldots, A_n$ we write $\bigotimes \Delta = A_1 \otimes \cdots \otimes A_n$. The we can reach state Δ' from state Δ if and only if we can prove

$$\Vdash (\bigotimes \Delta) \multimap (\bigotimes \Delta')\ \mathit{true}$$

where the axioms for the legal moves may be used arbitrarily many times. The reader is invited to prove various instances of the planning problem using the rules above.

This is still somewhat unsatisfactory. First of all, we may want to solve a planning problem where not the complete final state, but only some desired aspect of the final state (such as $\mathrm{on}(a, b)$) is specified. Second, the axioms fall outside of the framework on linear hypothetical judgments: they may be used in an unrestricted manner, while state is used linearly.

The first problem is easily remedied by adding another logical constant. The second is more complicated and postponed until the full discussion of natural deduction.

Additive Truth. The goal $\top$ can *always* be achieved, regardless of which resources we currently have. We can also think of it as consuming all available resources.

$$\frac{}{\Delta \Vdash \top\ \mathit{true}} \top I$$

Consequently, we have no information when we know $\top$ and there is no elimination rule. It should be noted that $\top$ is the unit of alternative conjunction in the sense that $A \& \top$ is equivalent to A.

We can use $\top$ in order to specify incomplete goals. For example, if we want to show that we can achieve a state where block a is no b, but we do not care about any other aspect of the state, we can ask if we can prove

$$\Delta_0 \Vdash \mathrm{on}(a,b) \otimes \top$$

where Δ_0 is the representation of the initial state. There is another form of trivial goal we discuss next.

Multiplicative Truth. The goal $\mathbf{1}$ can be achieved if we have no resources.

$$\frac{}{\cdot \Vdash \mathbf{1}\ \mathit{true}}\ \mathbf{1}I$$

Here we denote the empty collection of resources with "$\cdot$". In this case, knowing $\mathbf{1}$ *true* actually does give us some information, namely that the resources we have can be consumed. This is reflected in the elimination rule.

$$\frac{\Delta \Vdash \mathbf{1}\ \mathit{true} \qquad \Delta' \Vdash C\ \mathit{true}}{\Delta, \Delta' \Vdash C\ \mathit{true}}\ \mathbf{1}E$$

Multiplicative truth is the unit of $\otimes$ in the sense that $A \otimes \mathbf{1}$ is equivalent to A.

Using our intuitive understanding of the connectives, we can decide various judgments. And, of course, we can back this up with proofs given the rules above. We only give two examples here.

$$A \multimap (B \multimap C)\ \mathit{true} \Vdash (A \otimes B) \multimap C\ \mathit{true}$$

Informally we reason as follows:

In order to show $(A \otimes B) \multimap C$ we assume $A \otimes B$ and show C.
If we know $A \otimes B$ *true* we have both A and B simultaneously.
Using A and $A \multimap (B \multimap C)$ we can then obtain $B \multimap C$.
Using B we can now obtain C.

Note that we use every assumption exactly once in this argument—linearity is preserved.

$$A \otimes B\ \mathit{true} \Vdash A \& B\ \mathit{true}$$

This linear hypothetical judgment cannot be true for arbitrary A and B (although it could be true for some specific A and B). We reason as follows:

Assume $A \otimes B\ \mathit{true} \Vdash A \& B\ \mathit{true}$ holds for arbitrary A and B.
We know $A\ \mathit{true}, B\ \mathit{true} \Vdash A \otimes B\ \mathit{true}$.
Therefore, by the substitution principle, $A\ \mathit{true}, B\ \mathit{true} \Vdash A \& B\ \mathit{true}$.
We also know $A \& B\ \mathit{true} \Vdash A\ \mathit{true}$ by the hypothesis rule and $\&E_L$.
Therefore $A\ \mathit{true}, B\ \mathit{true} \Vdash A\ \mathit{true}$, again by substitution.
But this is a contradiction to the meaning of the linear hypothetical judgment (B is not used).

Bibliography

[Bib86] Wolfgang Bibel. A deductive solution for plan generation. *New Generation Computing*, 4:115–132, 1986.

[Doš93] Kosta Došen. A historical introduction to substructural logics. In Peter Schroeder-Heister and Kosta Došen, editors, *Substructural Logics*, pages 1–30. Clarendon Press, Oxford, England, 1993.

[Gen35] Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935. Translated under the title *Investigations into Logical Deductions* in [Sza69].

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Lin92] P. Lincoln. Linear logic. *ACM SIGACT Notices*, 23(2):29–37, Spring 1992.

[ML96] Per Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws. *Nordic Journal of Philosophical Logic*, 1(1):11–60, 1996.

[Pra65] Dag Prawitz. *Natural Deduction.* Almquist & Wiksell, Stockholm, 1965.

[Sce93] A. Scedrov. A brief guide to linear logic. In G. Rozenberg and A. Salomaa, editors, *Current Trends in Theoretical Computer Science*, pages 377–394. World Scientific Publishing Company, 1993. Also in Bulletin of the European Association for Theoretical Computer Science, volume 41, pages 154–165.

[SHD93] Peter Schroeder-Heister and Kosta Došen, editors. *Substructural Logics.* Number 2 in Studies in Logic and Computation. Clarendon Press, Oxford, England, 1993.

[Sza69] M. E. Szabo, editor. *The Collected Papers of Gerhard Gentzen.* North-Holland Publishing Co., Amsterdam, 1969.

[Tro92] A. S. Troelstra. *Lectures on Linear Logic.* CSLI Lecture Notes 29, Center for the Study of Language and Information, Stanford, California, 1992.