Chapter 4

Proof Search

Linear logic as introduced by Girard and presented in the previous chapter is a rich system for the formalization of reasoning involving state. It conservatively extends intuitionistic logic and can therefore also serve as the logical basis for general constructive mathematics. Searching for proofs in such an expressive logic is difficult, and one should not expect silver bullets.

Depending on the problem, proof search in linear logic can have a variety of applications. In the domain of planning problems (see Section 2.4) searching for a proof means searching for a plan. In the domain of concurrent computation (see Petri nets in Section 3.2 or the π-calculus in Section 3.6) searching for a proof means searching for possible computations. In the domain of logic programming (which we investigate in detail in Chapter **??**), searching for a proof according to a fixed strategy is the basic paradigm of computation. In the domain of functional programming and type theory (which we investigate in Chapter **??**), searching for a proof means searching for a program satisfying a given specification.

Each application imposes different requirements on proof search, but there are underlying basic techniques which recur frequently. In this chapter we take a look at some basic techniques, to be exploited in subsequent chapters.

4.1 Bottom-Up Proof Search and Inversion

The literature is not in agreement on the terminology, but we refer to the process of creating a derivation from the desired judgment on upward as *bottom-up* proof search. A snap-shot of a bottom-up search is a partial derivation, with undecided judgments at the top. Our goal is to derive all remaining judgments, thereby completing a proof.

We proceed by selecting a judgment which remains to be derived and an inference rule with which it might be inferred. We also may need to determine exactly how the conclusion of the rule matches the judgment. For example, in the $\otimes$R rule we need to decide how to split the linear hypotheses between the two premises. After these choices have been made, we reduce the goal of deriving the judgment to a number of subgoals, one for each premise of the selected rule. If there are no premises, the subgoal is solved. If there are no subgoals left, we have derived the original judgment.

One important observation about bottom-up proof search is that some rules are *invertible*, that is, the premises are derivable whenever the conclusion is derivable. The usual direction states that the conclusion is evident whenver the premises are. Invertible rules can safely be applied whenever possible without losing completeness, although some care must be taken to retain a terminating procedure in the presence of unrestricted hypotheses. We also separate *weakly invertible* rules, which only apply when there are no linear hypotheses (besides possibly the principal proposition of the inference rule). For example, we cannot apply the rule $\mathbf{1}$R whenever the judgment is $\Gamma;\Delta \vdash \mathbf{1}$, although it is safe to do so when there are no linear hypotheses. Similarly, we cannot use the initial sequent rule to infer $\Gamma;\Delta, A \Longrightarrow A$ unless $\Delta = \cdot$. Strongly invertible rules apply regardless of any other hypotheses.

Theorem 4.1 (Inversion Lemmas) *The following table lists invertible, weakly invertible, and non-invertible rule in intuitionistic linear logic.*

Strongly Invertible	*Weakly Invertible*	*Not Invertible*
$\multimap$R		$\multimap$L
$\otimes$L, $\mathbf{1}$L	$\mathbf{1}$R	$\otimes$R
&R, $\top$R		&L$_1$, &L$_2$
$\oplus$L, $\mathbf{0}$L		$\oplus$R$_1$, $\oplus$R$_2$
$\forall$R, $\exists$L		$\forall$L, $\exists$R
$\supset$R, !L	!R	$\supset$L

We exclude the init *and* copy *rules, since they are neither proper left nor proper right rules.*

Proof: For invertible rules we prove that each premise follows from the conclusion. For non-invertible rules we give a counterexample. The two sample case below are representative: for invertible rules we apply admissibility of cut, for non-invertible rules we consider a sequent with the same proposition on the left and right.

Case: $\multimap$R is invertible. We have to show that $\Gamma;(\Delta, A) \Longrightarrow B$ is derivable whenver $\Gamma;\Delta \Longrightarrow A \multimap B$ is derivable, so we assume $\Gamma;\Delta \Longrightarrow A \multimap B$. We also have $\Gamma; A, A \multimap B \Longrightarrow B$, which follows by one $\multimap$L rule from two initial sequents. From the admissibility of cut (Theorem 3.8) we then obtain directly $\Gamma;(\Delta, A) \Longrightarrow B$.

Case: $\multimap$L is not invertible. Consider $\cdot; A \multimap B \Longrightarrow A \multimap B$ for parameters A and B. There is only one way to use $\multimap$L to infer this, which leads to $\cdot;\cdot \Longrightarrow A$ and $\cdot; B \Longrightarrow A \multimap B$, neither of which is derivable. Therefore $\multimap$L is not invertible in general.

□

As a final, general property for bottom-up proof search we show that we can restrict ourselves to initial sequents of the form $\Gamma; P \Longrightarrow P$, where P is an atomic proposition. We write $\Gamma; \Delta \stackrel{-}{\Longrightarrow} A$ for the restricted judgment whose rules are as for $\Gamma; \Delta \Longrightarrow A$, except that initial sequents are restricted to atomic propositions. Obviously, if $\Gamma; \Delta \stackrel{-}{\Longrightarrow} A$ then $\Gamma; \Delta \Longrightarrow A$.

Theorem 4.2 (Completeness of Atomic Initial Sequents) *If* $\Gamma; \Delta \Longrightarrow A$ *then* $\Gamma; \Delta \stackrel{-}{\Longrightarrow} A$.

Proof: By induction on the the structure of $\mathcal{D} :: (\Gamma; \Delta \Longrightarrow A)$. In each case except initial sequents, we appeal directly to the induction hypothesis and infer $\Gamma; \Delta \stackrel{-}{\Longrightarrow} A$ from the results. For initial sequents, we use an auxiliary induction on the structure of the proposition A. We show only one case—the others are similar in that they follow the local expansions, translated from natural deduction to the setting of the sequent calculus. If local completeness did not hold for a connective, then atomic initial sequents would be incomplete as well.

Case: $A = A_1 \otimes A_2$. Then we construct

$$\cfrac{\cfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma; A_1 \stackrel{-}{\Longrightarrow} A_1\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \Gamma; A_2 \stackrel{-}{\Longrightarrow} A_2\end{array}}{\Gamma; A_1, A_2 \stackrel{-}{\Longrightarrow} A_1 \otimes A_2}\,{\otimes}\mathrm{R}}{\Gamma; A_1 \otimes A_2 \stackrel{-}{\Longrightarrow} A_1 \otimes A_2}\,{\otimes}\mathrm{L}$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ exist by induction hypothesis on A_1 and A_2.

□

The theorems in this section lead to a search procedure with the following general outline:

1. Pick a goal sequent to solve.
2. Decide to apply a right rule to the consequent or a left rule to a hypothesis.
3. Determine the remaining parameters (either how to split the hypotheses, or on the terms which may be required).
4. Apply the rule in the backward direction, reducing the goal to possibly several subgoals.

A lot of choices remain in this procedure. They can be classified according to the type of choice which must be made. This classification will guide us in the remainder of this chapter, as we discuss how to reduce the inherent non-determinism in the procedure above.

- Conjunctive choices. We know all subgoals have to be solved, but the order in which we attempt to solve them is not determined. In the simplest case, this is a form of *don't-care non-determinism*, since all subgoals have to be solved. In practice, it is not that simple since subgoals may interact once other choices have been made more deterministic. Success is a special case of conjunctive choice with no conjuncts.

- Disjunctive choices. We don't know which left or right rule to apply. Invertible rules are always safe, but once they all have been applied, many possibilities may remain. This is a form of *don't-know non-determinism*, since a sequence of correct guesses will lead to a derivation if there is one. In practice, this may be solved via backtracking, for example. Failure is a special case of a disjunctive choice with zero alternatives.

- Resource choices. We do not know how to divide our resources in the multiplicative rules. This is a special case of don't-know non-determinism which can be solved by different techniques collectively referred to as "resource managment". Resource management interacts tightly with other disjunctive and conjunctive choices.

- Universal choices. In the $\forall$R and $\exists$L rules we have to choose a new parameter. Fortunately, this is a trivial choice, since *any* new parameter will work, and its name is not important. Hence this is a form of don't-care non-determinism.

- Existential choices. In the $\exists$R and $\forall$L rules we have to choose a term t to substitute for the bound variable. Since there are potentially infinitely many terms (depending on the domain of quantification), this is a form of don't-know non-determinism. In practice, this is solved by *unification*, discussed in Section **??**.

4.2 Focusing

Focusing combines two basic phases in order to reduce non-determinism in proof search while remaining sound and complete.

1. (Inversion) Strongly invertible rules are applied eagerly. The order of these rule applications does not matter, so this is an instance of don't-care non-determinism.

2. (Focusing) After some steps we arrive at a sequent where all applicable rules with the exception of copy or init are non-invertible. Now we *focus*, either on a particular hypothesis or on the conclusion and apply a sequence of non-invertible rules until we have exposed an invertible principle connective. At this point in the proof search we return to the inversion phase.

We refer to this strategy as *focused proof search.* The idea and method are due to Andreoli [And92]; it is closely related to logic programming and the notion of *uniform proof* [MNPS91] as we will see in Chapter **??**.

Just like the sequent calculus followed inevitably from natural deduction, focused proof search seems to follow inevitably from the sequent calculus. It is remarkably robust in that in our experience, any logic that admits a clean sequent calculus also admits a similarly clean focusing calculus. This is true even for logics such as classical logic for which good natural deduction systems that arise from judgmental considerations are elusive.

While the basic intuition is simple, giving an unambiguous specification of focusing is a non-trivial task. Both the proper representation of the don't-care non-determinism and the notion of focus proposition for phase (2) require some experience and (eventually) lengthy correctness proofs.

In order to aid the description of the rules, we define some classes of propositions. We say A is *right asynchronous* if the top-level connective of A has a strongly invertible right rule. Similarly, A is *left asynchronous* if the top-level connective of A has a strongly invertible left rule. The intuition is that of asynchronous communication, where a sending process can proceed immediately without waiting for receipt of its message. Dually, a proposition is *right* or *left synchronous* is its top-level connective has a non-invertible or only weakly invertible right or left rule, respectively.

Atomic	P
Right Asynchronous	$A_1 \multimap A_2, A_1 \& A_2, \top, A_1 \supset A_2, \forall x.\ A$
Left Asynchronous	$A_1 \otimes A_2, \mathbf{1}, A_1 \oplus A_2, \mathbf{0}, !A, \exists x.\ A$
Right Synchronous	$A_1 \otimes A_2, \mathbf{1}, A_1 \oplus A_2, \mathbf{0}, !A, \exists x.\ A$
Left Synchronous	$A_1 \multimap A_2, A_1 \& A_2, \top, A_1 \supset A_2, \forall x.\ A$

Note that the left asynchronous and right synchronous propositions are identical, as are the right asynchronous and left synchronous. We therefore really need only two classes of propositions, but this tends to be very confusing.

Inversion. The first phase of proof search decomposes all asynchronous connectives. This means there is a lot of don't-care non-determinism, since the order in which the rules are applied is irrelevant. We build this into our system by *fixing* a particular order in which the asynchronous connectives are decomposed: first the succedent, then the antecedents from right to left. This means we need a new form of hypothetical judgment, an *ordered hypothetical judgment,* since otherwise we cannot prescribe a fixed order. We thoroughly treat this judgment form in Chapter **??**. We write

$$\Gamma; \Delta; \Omega \Longrightarrow A \Uparrow$$

where

Γ are unrestricted hypotheses (which may be arbitrary),
Δ are linear hypotheses that *may not be left asynchronous*,
Ω are ordered hypotheses (which may be arbitrary),
A is the goal (which may be arbitrary).

The first set of rules treats all right asynchronous connectives.

$$\frac{\Gamma; \Delta; \Omega, A \Longrightarrow B \Uparrow}{\Gamma; \Delta; \Omega \Longrightarrow A \multimap B \Uparrow} \multimap R \qquad \frac{\Gamma; \Delta; \Omega \Longrightarrow A \Uparrow \qquad \Gamma; \Delta; \Omega \Longrightarrow B \Uparrow}{\Gamma; \Delta; \Omega \Longrightarrow A \& B \Uparrow} \& R$$

$$\frac{}{\Gamma; \Delta; \Omega \Longrightarrow \top \Uparrow} \top R \qquad \frac{\Gamma, A; \Delta; \Omega \Longrightarrow B \Uparrow}{\Gamma; \Delta; \Omega \Longrightarrow A \supset B} \supset R$$

$$\frac{\Gamma; \Delta; \Omega \Longrightarrow [a/x]A}{\Gamma; \Delta; \Omega \Longrightarrow \forall x.\, A} \forall R^a$$

Once the goal proposition is no longer asynchronous, we proceed with the hypotheses in Ω, decomposing all left asynchronous propositions in them. We write

$$\Gamma; \Delta; \Omega \Uparrow \Longrightarrow C$$

where

Γ are unrestricted hypotheses (arbitrary)
Δ are linear hypotheses (not left asynchronous)
Ω are ordered hypotheses (arbitrary)
C is the goal (not right asynchronous)

First, we have the rule to transiton to this judgment.

$$\frac{\Gamma; \Delta; \Omega \Uparrow \Longrightarrow C, \quad C \text{ not right asynchronous}}{\Gamma; \Delta; \Omega \Longrightarrow C \Uparrow} \Uparrow R$$

Next we have rules to decompose the left asynchronous proposition in Ω *in order*, that is, at the right end of the ordered hypotheses.

$$\frac{\Gamma; \Delta; \Omega, A, B \Uparrow \Longrightarrow C}{\Gamma; \Delta; \Omega, A \otimes B \Uparrow \Longrightarrow C} \otimes L \qquad \frac{\Gamma; \Delta; \Omega \Uparrow \Longrightarrow C}{\Gamma; \Delta; \Omega, \mathbf{1} \Uparrow \Longrightarrow C} \mathbf{1}L$$

$$\frac{\Gamma; \Delta; \Omega, A \Uparrow \Longrightarrow C \qquad \Gamma; \Delta; \Omega, B \Uparrow \Longrightarrow C}{\Gamma; \Delta; \Omega, A \oplus B \Uparrow \Longrightarrow C} \oplus L \qquad \frac{}{\Gamma; \Delta; \Omega, \mathbf{0} \Uparrow \Longrightarrow C} \mathbf{0}L$$

$$\frac{\Gamma, A; \Delta; \Omega \Uparrow \Longrightarrow C}{\Gamma; \Delta; \Omega, !A \Uparrow \Longrightarrow C} !L \qquad \frac{\Gamma; \Delta; \Omega, [a/x]A \Uparrow \Longrightarrow C}{\Gamma; \Delta; \Omega, \exists x.\, A \Uparrow \Longrightarrow C} \exists L^a$$

When we encounter a proposition that is not left asynchronous, we move it into Δ so that we can eventually eliminate all propositions from Ω.

$$\frac{\Gamma; \Delta, A; \Omega \Uparrow \Longrightarrow C, \quad A \text{ not left asynchrounous}}{\Gamma; \Delta; \Omega, A \Uparrow \Longrightarrow C} \Uparrow L$$

Decision. When we start from $\Gamma; \cdot; \Omega \Longrightarrow A \Uparrow$, searching backwards, we can always reach a situation of the form $\Gamma'; \Delta'; \cdot \Uparrow \Longrightarrow C$ for each leaf where C is not right asynchronous and Δ' contains no propositions that are left asynchronous. At this point we need to decide which proposition to focus on. Note that we always focus on a single proposition. First the case we focus on the right. For this we need a new judgment

$$\Gamma; \Delta \Longrightarrow A \Downarrow$$

where

Γ are unrestricted hypotheses (arbitrary),
Δ are linear hypotheses (not left asynchronous),
A is the focus proposition (arbitrary).

If we focus on the left, we need an analogous judgment.

$$\Gamma; \Delta; A \Downarrow \Longrightarrow C$$

where

Γ are unrestricted hypotheses (arbitary),
Δ are linear hypotheses (not left asynchronous),
A is the focus proposition (arbitrary),
C is the succedent (not right asynchronous)

The decision is between the following rules that transition into these two jugments.

$$\frac{\Gamma; \Delta \Longrightarrow C \Downarrow, \quad C \text{ not atomic}}{\Gamma; \Delta; \cdot \Uparrow \Longrightarrow C}\,\mathsf{decideR}$$

$$\frac{\Gamma; \Delta; A \Downarrow \Longrightarrow C}{\Gamma; \Delta, A; \cdot \Uparrow \Longrightarrow C}\,\mathsf{decideL} \qquad \frac{\Gamma, A; \Delta; A \Downarrow \Longrightarrow C}{\Gamma, A; \Delta; \cdot \Uparrow \Longrightarrow C}\,\mathsf{decideL!}$$

Note that $\mathsf{decideL!}$ is justified by the copy rule.

Focusing. Once we have decided which proposition to focus on, we apply a succession of non-invertible rules.

$$\frac{\Gamma; \Delta_1 \Longrightarrow A_1 \Downarrow \qquad \Gamma; \Delta_2 \Longrightarrow A_2 \Downarrow}{\Gamma; \Delta_1, \Delta_2 \Longrightarrow A_1 \otimes A_2 \Downarrow}\,\otimes\mathrm{R} \qquad \frac{}{\Gamma; \cdot \Longrightarrow \mathbf{1} \Downarrow}\,\mathbf{1}\mathrm{R}$$

$$\frac{\Gamma; \Delta \Longrightarrow A \Downarrow}{\Gamma; \Delta \Longrightarrow A \oplus B \Downarrow}\,\oplus\mathrm{R}_1 \qquad \frac{\Gamma; \Delta \Longrightarrow B \Downarrow}{\Gamma; \Delta \Longrightarrow A \oplus B \Downarrow}\,\oplus\mathrm{R}_2$$

$$\text{no right rule for } \mathbf{0} \qquad \frac{\Gamma; \Delta \Longrightarrow [t/x]A \Downarrow}{\Gamma; \Delta \Longrightarrow \exists x.\, A \Downarrow}\,\exists\mathrm{R}^a$$

$$\frac{\Gamma; \cdot; \cdot \Longrightarrow A \Uparrow}{\Gamma; \cdot \Longrightarrow\, !A \Downarrow}\,!\mathrm{R}$$

The last rule is a somewhat special case: because !R is weakly right invertible, we immediately transition back to break down the right asynronous connectives in A. In the other weakly right invertible rule, $\mathbf{1}$R, we conclude the proof so no special provision is necessary.

The corresponding left rules are as follows:

$$\frac{\Gamma; \Delta_2; B \Downarrow \Longrightarrow C \qquad \Gamma; \Delta_1; \cdot \Longrightarrow A \Uparrow}{\Gamma; \Delta_1, \Delta_2; A \multimap B \Downarrow \Longrightarrow C} \multimap L$$

$$\frac{\Gamma; \Delta; A \Downarrow \Longrightarrow C}{\Gamma; \Delta; A \& B \Downarrow \Longrightarrow C} \& L_1 \qquad \frac{\Gamma; \Delta; B \Downarrow \Longrightarrow C}{\Gamma; \Delta; A \& B \Downarrow \Longrightarrow C} \& L_2$$

$$\text{no left rule for } \top \qquad \frac{\Gamma; \Delta; B \Downarrow \Longrightarrow C \qquad \Gamma; \cdot; \cdot \Longrightarrow A \Uparrow}{\Gamma; \Delta; A \supset B \Downarrow \Longrightarrow C} \supset L$$

$$\frac{\Gamma; \Delta; [t/x]A \Downarrow \Longrightarrow C}{\Gamma; \Delta; \forall x.\, A \Downarrow \Longrightarrow C} \forall L$$

Eventually we must break down the focus proposition to the point where it is no longer synchronous. If it is atomic, we either succeed or fail in our overall proof attempt. In other cases we switch back to the inversion judgment.

$$\frac{}{\Gamma; \Delta; P \Downarrow \Longrightarrow P} \mathsf{init}$$

$$\frac{\Gamma; \Delta; A \Uparrow \Longrightarrow C \quad A \text{ not atomic and not left synchronous}}{\Gamma; \Delta; A \Downarrow \Longrightarrow C} \Downarrow L$$

$$\frac{\Gamma; \Delta; \cdot \Longrightarrow A \Uparrow}{\Gamma; \Delta \Longrightarrow A \Downarrow} \Downarrow R$$

The soundness of these rules is relatively easy to establish, since, in the end, the system just represents a restriction on the application of the usual left, right, initial and copy rules. Completeness of the corresponding system for classical linear logic has been proven by Andreoli [And92] and is lengthy. The completeness of the rules above has not yet been considered, but Andreoli's techniques would seem to apply fairly directly.[1]

In order to state soundness formally, we use convention that Δ, Ω joins the contexts Δ and Ω, ignoring the order of the hypotheses in Ω.

Theorem 4.3 (Soundness of Focusing)

1. *If* $\Gamma; \Delta; \Omega \Longrightarrow A \Uparrow$ *then* $\Gamma; (\Delta, \Omega) \Longrightarrow A$.
2. *If* $\Gamma; \Delta; \Omega \Uparrow \Longrightarrow C$ *then* $\Gamma; (\Delta, \Omega) \Longrightarrow C$.

[1] [*This might make an interesting class project.*]

3. *If* $\Gamma; \Delta \Longrightarrow A \Downarrow$ *then* $\Gamma; \Delta \Longrightarrow A$.

4. *If* $\Gamma; \Delta; A \Downarrow \Longrightarrow C$ *then* $\Gamma; (\Delta, A) \Longrightarrow C$

Proof: By straightforward simultaneous induction on the structure of the given deductions. □

Bibliography

[ABCJ94] D. Albrecht, F. Bäuerle, J. N. Crossley, and J. S. Jeavons. Curry-Howard terms for linear logic. In ??, editor, *Logic Colloquium '94*, pages ??–?? ??, 1994.

[Abr93] Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.

[And92] Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):197–347, 1992.

[Bar96] Andrew Barber. Dual intuitionistic linear logic. Technical Report ECS-LFCS-96-347, Department of Computer Science, University of Edinburgh, September 1996.

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

[Bie94] G. Bierman. On intuitionistic linear logic. Technical Report 346, University of Cambridge, Computer Laboratory, August 1994. Revised version of PhD thesis.

[BS92] G. Bellin and P. J. Scott. On the π-calculus and linear logic. Manuscript, 1992.

[Cer95] Iliano Cervesato. Petri nets and linear logic: a case study for logic programming. In M. Alpuente and M.I. Sessa, editors, *Proceedings of the Joint Conference on Declarative Programming (GULP-PRODE'95)*, pages 313–318, Marina di Vietri, Italy, September 1995. Palladio Press.

[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.

[Gir93] J.-Y. Girard. On the unity of logic. *Annals of Pure and Applied Logic*, 59:201–217, 1993.

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

[Mil92] D. Miller. The π-calculus as a theory in linear logic: Preliminary results. In E. Lamma and P. Mello, editors, *Proceedings of the Workshop on Extensions of Logic Programming*, pages 242–265. Springer-Verlag LNCS 660, 1992.

[Mil99] Robin Milner. *Communicating and Mobile Systems: the π-Calculus*. Cambridge University Press, 1999.

[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.

[MNPS91] Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.

[MOM91] N. Martí-Oliet and J. Meseguer. From Petri nets to linear logic through categories: A survey. *Journal on Foundations of Computer Science*, 2(4):297–399, December 1991.

[PD01] Frank Pfenning and Rowan Davies. A judgmental reconstruction of modal logic. *Mathematical Structures in Computer Science*, 11:511–540, 2001. Notes to an invited talk at the *Workshop on Intuitionistic Modal Logics and Applications* (IMLA'99), Trento, Italy, July 1999.

[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.

[Tro93] A. S. Troelstra. Natural deduction for intuitionistic linear logic. Prepublication Series for Mathematical Logic and Foundations ML-93-09, Institute for Language, Logic and Computation, University of Amsterdam, 1993.