Chapter 3

Sequent Calculus

In the previous chapter we developed linear logic in the form of natural deduction, which is appropriate for many applications of linear logic. It is also highly economical, in that we only needed one basic judgment (*A true*) and two judgment forms (linear and unrestricted hypothetical judgments) to explain the meaning of all connectives we have encountered so far. However, it is not well-suited directly proof search, because this involves mixing forward and backward reasoning even if we restrict ourselves to searching for normal deductions.

In this chapter we develop a sequent calculus as a calculus of proof search for normal natural deductions. We then extend it with a rule of cut that allows us to model arbitrary natural deductions. The central theorem of this chapter is *cut elimination* which shows that the cut rule is admissible. We obtain the normalization theorem for natural deduction as a direct consequence of this theorem. It was this latter application which lead to the original discovery of the sequent calculus by Gentzen [Gen35]. There are many useful immediate corollaries of the cut elimination theorem, such as consistency of the logic, or the disjunction property.

3.1 Cut-Free Sequent Calculus

In this section we transcribe the process of searching for *normal* natural deductions into an inference system. In the context of sequent calculus, proof search is seen entirely as the bottom-up construction of a derivation. This means that elimination rules must be turned "upside-down" so they can also be applied bottom-up rather than top-down.

In terms of judgments we develop the sequent calculus via a splitting of the judgment "*A is true*" into two judgments: "*A is a resource*" (*A res*) and "*A is a goal*" (*A goal*). Ignoring unrestricted hypothesis for the moment, the main judgment

$$w_1{:}A_1\ \mathit{res}, \ldots, w_n{:}A_n\ \mathit{res} \Longrightarrow C\ \mathit{goal}$$

expresses

> Under the linear hypothesis that we have resources $A_1, \ldots, A_n$ we can achieve goal C.

In order to model validity, we add inexhaustible resources or *resource factories*, written *A fact*. We obtain

$$(v_1{:}B_1\,\mathit{fact}, \ldots, v_m{:}B_m\,\mathit{fact}); (w_1{:}A_1\;\mathit{res}, \ldots, w_n{:}A_n\;\mathit{res}) \Longrightarrow C\;\mathit{goal},$$

which expresses

> Under the unrestricted hypotheses that we have resource factories $B_1, \ldots, B_m$ and linear hyptheses that we have resources $A_1, \ldots, A_n$, we can achieve goal C.

As before, the order of the hypothesis (linear or unrestricted) is irrelevant, and we assume that all hypothesis labels v_j and w_i are distinct.

Resources and goals are related in that with the resource A we can achieve goal A. Recall that the linear hypothetical judgment requires us to use all linear hypotheses exactly once. We therefore have the following rule.

$$\frac{}{\Gamma; u{:}A\;\mathit{res} \Longrightarrow A\;\mathit{goal}}\;\mathsf{init}_u$$

We call such as sequent *initial* and write init. Note that, for the moment, we do not have the opposite: if we can achieve goal A we cannot assume A as a resource. The corresponding rule will be called cut and is shown later to be admissible, that is, every instance of this rule can be eliminated from a proof. It is the desire to rule out cut that necessitated splitting truth into two judgments.

Note that the initial rule does *not* follow directly from the nature of linear hypothetical judgments, since $A\;\mathit{res}$ and $A\;\mathit{goal}$ are different judgments. Instead, it explicitly states a connection between resources and goals. A rule that concludes $\Gamma, A\;\mathit{res} \Longrightarrow A\;\mathit{res}$ is also evident, but is not of interest here since we never consider the judgment $A\;\mathit{res}$ in the succedent of a sequent.

We also need a rule that allows a factory to produce a resource. This rule is called copy and sometimes refered to as *dereliction*.

$$\frac{(\Gamma, v{:}A\,\mathit{fact}); (\Delta, w{:}A\;\mathit{res}) \Longrightarrow C\;\mathit{goal}}{(\Gamma, v{:}A\,\mathit{fact}); \Delta \Longrightarrow C\;\mathit{goal}}\;\mathsf{copy}_v$$

Note how this is different from the unrestricted hypothesis rule in natural deduction. Factories are directly related to resources and only indirectly to goals.

The remaining rules are divided into *right* and *left* rules, which correspond to the *introduction* and *elimination* rules of natural deduction, respectively. The right rules apply to the goal, while the left rules apply to resources. In the following, we adhere to common practice and omit labels on hypotheses and consequently also on the justifications of the inference rules. The reader should keep in mind, however, that this is just a short-hand, and that there are, for example, two *different* derivations of $(A, A); \cdot \Longrightarrow A$, one using the first copy of A and one using the second.

Hypotheses.

$$\frac{}{\Gamma; A \Longrightarrow A}\,\mathsf{init} \qquad \frac{(\Gamma, A); (\Delta, A) \Longrightarrow C}{(\Gamma, A); \Delta \Longrightarrow C}\,\mathsf{copy}$$

Multiplicative Connectives.

$$\frac{\Gamma; \Delta, A \Longrightarrow B}{\Gamma; \Delta \Longrightarrow A \multimap B}\multimap\mathrm{R} \qquad \frac{\Gamma; \Delta_1 \Longrightarrow A \qquad \Gamma; \Delta_2, B \Longrightarrow C}{\Gamma; \Delta_1, \Delta_2, A \multimap B \Longrightarrow C}\multimap\mathrm{L}$$

$$\frac{\Gamma; \Delta_1 \Longrightarrow A \qquad \Gamma; \Delta_2 \Longrightarrow B}{\Gamma; \Delta_1, \Delta_2 \Longrightarrow A \otimes B}\otimes\mathrm{R} \qquad \frac{\Gamma; \Delta, A, B \Longrightarrow C}{\Gamma; \Delta, A \otimes B \Longrightarrow C}\otimes\mathrm{L}$$

$$\frac{}{\Gamma; \cdot \Longrightarrow \mathbf{1}}\,\mathbf{1}\mathrm{R} \qquad \frac{\Gamma; \Delta \Longrightarrow C}{\Gamma; \Delta, \mathbf{1} \Longrightarrow C}\,\mathbf{1}\mathrm{L}$$

Additive Connectives.

$$\frac{\Gamma; \Delta \Longrightarrow A \qquad \Gamma; \Delta \Longrightarrow B}{\Gamma; \Delta \Longrightarrow A \& B}\,\&\mathrm{R} \qquad \frac{\Gamma; \Delta, A \Longrightarrow C}{\Gamma; \Delta, A \& B \Longrightarrow C}\,\&\mathrm{L}_1 \qquad \frac{\Gamma; \Delta, B \Longrightarrow C}{\Gamma; \Delta, A \& B \Longrightarrow C}\,\&\mathrm{L}_2$$

$$\frac{}{\Gamma; \Delta \Longrightarrow \top}\,\top\mathrm{R} \qquad \textit{No } \top \textit{ left rule}$$

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

$$\textit{No } \mathbf{0} \textit{ right rule} \qquad \frac{}{\Gamma; \Delta, \mathbf{0} \Longrightarrow C}\,\mathbf{0}\mathrm{L}$$

Quantifiers.

$$\frac{\Gamma; \Delta \Longrightarrow [a/x]A}{\Gamma; \Delta \Longrightarrow \forall x.\, A}\ \forall\mathrm{R}^a \qquad \frac{\Gamma; \Delta, [t/x]A \Longrightarrow C}{\Gamma; \Delta, \forall x.\, A \Longrightarrow C}\ \forall\mathrm{L}$$

$$\frac{\Gamma; \Delta \Longrightarrow [t/x]A}{\Gamma; \Delta \Longrightarrow \exists x.\, A}\ \exists\mathrm{R} \qquad \frac{\Gamma; \Delta, [a/x]A \Longrightarrow C}{\Gamma; \Delta, \exists x.\, A \Longrightarrow C}\ \exists\mathrm{L}^a$$

Exponentials.

$$\frac{(\Gamma, A); \Delta \Longrightarrow B}{\Gamma; \Delta \Longrightarrow A \supset B}\ \supset\mathrm{R} \qquad \frac{\Gamma; \cdot \Longrightarrow A \qquad \Gamma; \Delta, B \Longrightarrow C}{\Gamma; \Delta, A \supset B \Longrightarrow C}\ \supset\mathrm{L}$$

$$\frac{\Gamma; \cdot \Longrightarrow A}{\Gamma; \cdot \Longrightarrow\ !A}\ !\mathrm{R} \qquad \frac{(\Gamma, A); \Delta \Longrightarrow C}{\Gamma; (\Delta, !A) \Longrightarrow C}\ !\mathrm{L}$$

To obtain a normal deduction from a sequent derivation we map instances of right rules to corresponding introduction rules. Left rules have to be turned "upside-down", since the elimination rule corresponding to a left rule works in the opposite direction. This reverse of direction is captured in the proof of the following theorem by appeals to the substitution property: we extend a natural deduction at a leaf by substituting a one-step deduction for the use of a hypothesis. Note that the terse statement of this theorem (and also of the completeness theorem below) hide the fact that the judgments forming the assumptions Γ and Δ are different in the sequent calculus and natural deduction.

Theorem 3.1 (Soundness of Sequent Derivations)
If $\Gamma; \Delta \Longrightarrow A$ *then* $\Gamma; \Delta \vdash A \uparrow$.

Proof: By induction on the structure of the derivation of $\Gamma; \Delta \Longrightarrow A$. Initial sequents are translated to the $\downarrow\uparrow$ coercion, and use of an unrestricted hypothesis follows by a substitution principle (Lemma 2.2). For right rules we apply the corresponding introduction rules. For left rules we either directly construct a derivation of the conclusion after an appeal to the induction hypothesis ($\otimes$L, $\mathbf{1}$L, $\oplus$L, $\mathbf{0}$L, $\exists$L, !L) or we appeal to a substitution principle of atomic natural deductions for hypotheses ($\multimap$ L, $\&\mathrm{L}_1$, $\&\mathrm{L}_2$, $\forall$L, $\supset$L). □

The completeness theorem reverses the translation from above. In this case we have to generalize the induction hypothesis so we can proceed when we encounter a coercion from atomic to normal derivations. It takes some experience to find the generalization we give below. Fortunately, the rest of the proof is then straightforward.

Theorem 3.2 (Completeness of Sequent Derivations)

1. *If* $\Gamma; \Delta \vdash A \uparrow$ *then there is a sequent derivation of* $\Gamma; \Delta \Longrightarrow A$*, and*

2. *if* $\Gamma; \Delta \vdash A \downarrow$ *then for any formula* C *and derivation of* $\Gamma; \Delta', A \Longrightarrow C$ *there is a derivation of* $\Gamma; (\Delta', \Delta) \Longrightarrow C$*.*

Proof: By simultaneous induction on the structure of the derivations of $\Gamma; \Delta \vdash A \uparrow$ and $\Gamma; \Delta \vdash A \downarrow$. □

3.2 Another Example: Petri Nets

In this section we show how to represent Petri nets in linear logic. This example is due to Martì-Oliet and Meseguer [MOM91], but has been treated several times in the literature.

A *Petri net* is defined by a collection of *places*, *transitions*, *arcs*, and *tokens*. Every transition has input arcs and output arcs that connect it to places. The system evolves by changing the tokens in various places according to the following rules.

1. A transition is enabled if every place connected to it by an input arc contains at least one token.

2. We non-deterministically select one of the enabled transitions in a net to fire.

3. A transition fires by removing one token from each input place and adding one token to each output place of the transition.

Slightly more generally, an arc may have a *weight* n. For an input arc this means there must be at least n tokens on the place to enable a transition. When the transition fires, n tokens are removed from the token at the beginning of an arc with weight n. For an output arc this means that n new tokens will be added to the place at its end. By default, an arc with no listed weight has weight one. There are other variations and generalizations of Petri nets, but we will not consider them here. Figure 3.1 displays some typical Petri net structures.

It is quite easy to represent a Petri net in linear logic. The idea is that the fixed topology of the net is represented as a collection of unrestricted propositions Γ. The current state of the net as given by the tokens in the net is represent as collection of resources Δ. We can reach state Δ_1 from state Δ_0 iff $\Gamma; \cdot \vdash (\bigotimes \Delta_0) \multimap (\bigotimes \Delta_1)$. That is, provability will correspond precisely to reachability in the Petri net. We formulate this below in a slightly differently, using the sequent calculus as a tool.

To accomplish this, we represent every place by an atomic predicate. If there are k tokens on place p, we add k copies of p into the representation of the state Δ. For every transition we add a rule $p_1 \otimes \cdots \otimes p_m \multimap q_1 \otimes \cdots \otimes q_n$ to Γ, where $p_1, \ldots, p_m$ are the places at the beginning of the input arcs and $q_1 \otimes \cdots \otimes q_n$ are the places at the end of the output arcs. If an arc has multiplicity k, we simply add k copies of p to either the antecedent or the succedent of the corresponding

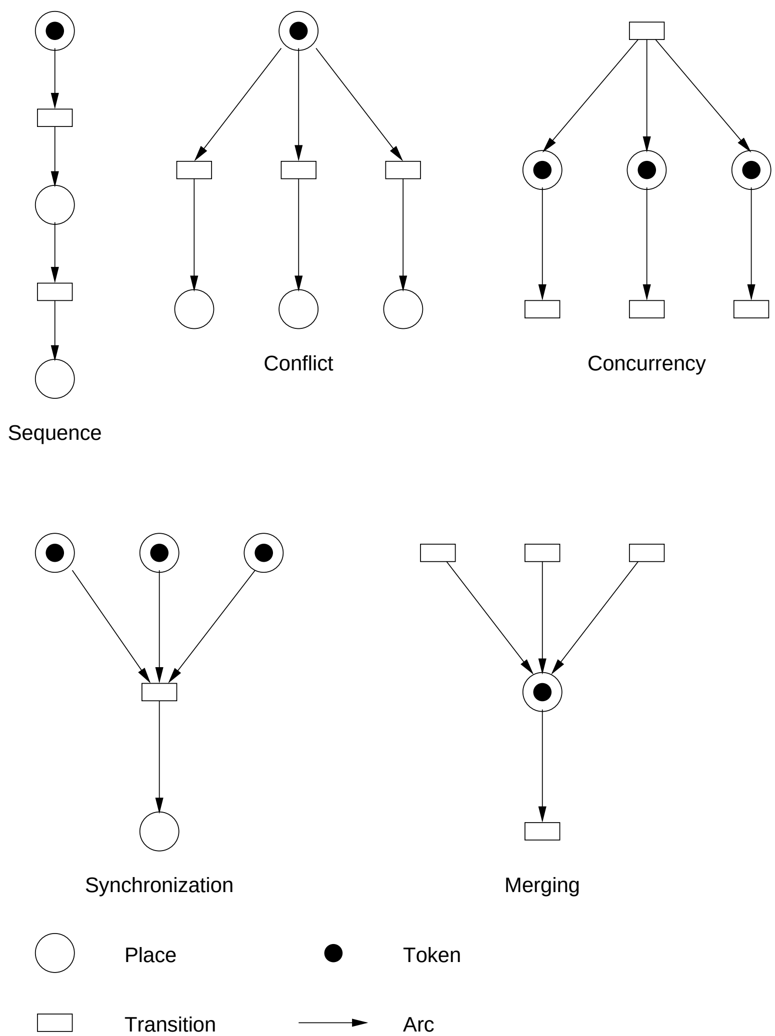
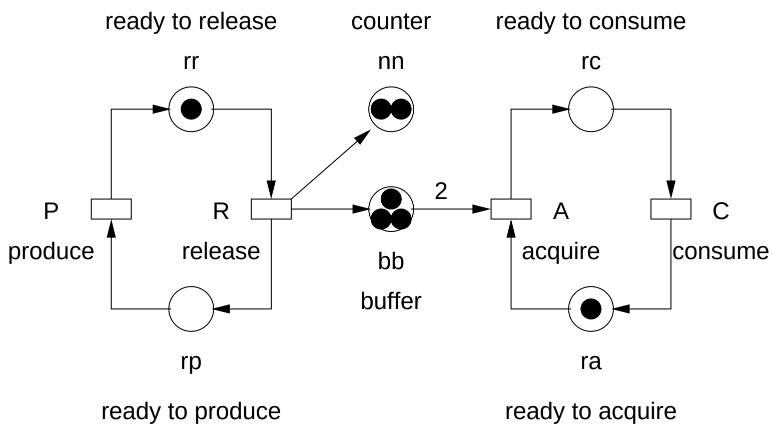


Figure 3.1: Some Petri Net Structures

linear implication representing the transition. As an example, consider the following Petri net (used in [?]).



Note that arc form the buffer to the acquire transition has weight two, so two tokens in the buffer are converted to one token to be in the place ready to consume.

The representation of this Petri net consists of the following unrestricted rule in Γ and the initial state in Δ_0.

$$\begin{aligned} \Gamma \quad &= \quad P : \mathsf{rp} \multimap \mathsf{rr} \\ &\qquad R : \mathsf{rr} \multimap \mathsf{rp} \otimes \mathsf{nn} \otimes \mathsf{bb} \\ &\qquad A : \mathsf{bb} \otimes \mathsf{bb} \otimes \mathsf{ra} \multimap \mathsf{rc} \\ &\qquad C : \mathsf{rc} \multimap \mathsf{ra} \\ \Delta_0 \quad &= \quad \mathsf{rr}, \mathsf{nn}, \mathsf{nn}, \mathsf{bb}, \mathsf{bb}, \mathsf{bb}, \mathsf{ra} \end{aligned}$$

Informally, it is quite easy to understand that the propositions above represent the given Petri nets. We now consider a slightly different from of the adequacy theorem in order exploit the sequent calculus

Adequacy for Encoding of Petri Nets.

Assume we are given a Petri net with places $P = \{p_1, \ldots, p_n\}$, transitions $T = \{t_1, \ldots, t_m\}$. We represent the transitions as unrestricted assumptions Γ as sketched above, and a token assignment as a collection of linear hypotheses $\Delta = (q_1, \ldots, q_k)$ where q_j are places, possibly containing repetitions. Then the marking Δ_1 is reachable from marking Δ_0 if and only if

$$\begin{array}{c} \Gamma; \Delta_1 \Longrightarrow C \\ \vdots \\ \Gamma; \Delta_0 \Longrightarrow C \end{array}$$

for an arbitrary proposition C.

Considered bottom up, this claims that, for any C, we can reduce the problem of proving $\Gamma; \Delta_0 \vdash C$ to the problem of proving $\Gamma; \Delta_1 \vdash C$.

3.3 Deductions with Lemmas

One common way to find or formulate a proof is to introduce a lemma. In the sequent calculus, the introduction and use of a lemma during proof search is modelled by the rules of cut, cut for lemmas used as linear hypotheses, and cut! for lemmas used as factories or resources. The corresponding rule for intuitionistic logic is due to Gentzen [Gen35]. We write $\Gamma; \Delta \overset{+}{\Longrightarrow} A$ for the judgment that A can be derived with the rules from before, plus one of the two cut rules below.

$$\frac{\Gamma; \Delta \overset{+}{\Longrightarrow} A \quad \Gamma; (\Delta', A) \overset{+}{\Longrightarrow} C}{\Gamma; \Delta, \Delta' \overset{+}{\Longrightarrow} C}\,\mathsf{cut} \qquad \frac{\Gamma; \cdot \overset{+}{\Longrightarrow} A \quad (\Gamma, A); \Delta' \overset{+}{\Longrightarrow} C}{\Gamma; \Delta' \overset{+}{\Longrightarrow} C}\,\mathsf{cut!}$$

Note that the linear context in the left premise of the cut! rule must be empty, because the new hypothesis A in the right premise is unrestricted in its use.

From the judgmental point of view, the first cut rule corresponds to the inverse of the init rule. Ignoring extraneous hypotheses, the init rule states $A\ \mathit{res} \Longrightarrow A\ \mathit{goal}$. To go the opposity way means that we are allowed to assume $A\ \mathit{res}$ if we have shown $A\ \mathit{goal}$. This is exactly what the cut rule expresses. The cut! expresses that if we can achieve a goal A without using any linear resources, we can manufacture as many copies of the resource A as we like.

On the side of natural deduction, these rules correspond to substitution principles. They can be related to normal and atomic derivations only if we allow an additional coercion from normal to atomic derivations. This is because the left premise corresponds to a derivation of $\Gamma; \Delta \vdash A \uparrow$ which can be substituted into a derivation of $\Gamma; \Delta', A \downarrow \vdash C \uparrow$ only we have this additional coercion. Of course, the resulting deductions are no longer normal in the sense we defined before, so we write $\Gamma; \Delta \vdash^{+} A \downarrow$ and $\Gamma; \Delta \vdash^{+} A \uparrow$. These judgments are defined with the same rules as $\Gamma; \Delta \vdash A \uparrow$ and $\Gamma; \Delta \vdash A \downarrow$, plus the following coercion.

$$\frac{\Gamma; \Delta \vdash^{+} A \uparrow}{\Gamma; \Delta \vdash^{+} A \downarrow}\,\uparrow\downarrow$$

It is now easy to prove that arbitrary natural deductions can be annotated with $\uparrow$ and $\downarrow$, since we can arbitrarily coerce back and forth between the two judgments.

Theorem 3.3 *If* $\Gamma; \Delta \vdash A$ *then* $\Gamma; \Delta \vdash^{+} A \uparrow$ *and* $\Gamma; \Delta \vdash^{+} A \downarrow$

Proof: By induction on the structure of $\mathcal{D} :: (\Gamma; \Delta \vdash A)$. □

Theorem 3.4

1. *If* $\Gamma; \Delta \vdash^{+} A\!\uparrow$ *then* $\Gamma; \Delta \vdash A$.

2. *If* $\Gamma; \Delta \vdash^{+} A\!\downarrow$ *then* $\Gamma; \Delta \vdash A$.

Proof: My mutual induction on $\mathcal{N} :: (\Gamma; \Delta \vdash^{+} A\!\uparrow)$ and $\mathcal{A} :: (\Gamma; \Delta \vdash^{+} A\!\downarrow)$. □

It is also easy to relate the cut rules to the new coercions (and thereby to natural deductions), plus four substitution principles.

Property 3.5 (Substitution)

1. *If* $\Gamma; \Delta \vdash^{+} A \downarrow$ *and* $\Gamma; (\Delta', u{:}A\!\downarrow) \vdash^{+} C \uparrow$ *then* $\Gamma; (\Delta, \Delta') \vdash^{+} C \uparrow$.

2. *If* $\Gamma; \Delta \vdash^{+} A \downarrow$ *and* $\Gamma; (\Delta', u{:}A\!\downarrow) \vdash^{+} C \downarrow$ *then* $\Gamma; (\Delta, \Delta') \vdash^{+} C \downarrow$.

3. *If* $\Gamma; \cdot \vdash^{+} A \downarrow$ *and* $(\Gamma, v{:}A\!\downdownarrows); \Delta' \vdash^{+} C \uparrow$ *then* $\Gamma; \Delta' \vdash^{+} C \uparrow$.

4. *If* $\Gamma; \cdot \vdash^{+} A \downarrow$ *and* $(\Gamma, v{:}A\!\downdownarrows); \Delta' \vdash^{+} C \downarrow$ *then* $\Gamma; \Delta' \vdash^{+} C \downarrow$.

Proof: By mutual induction on the structure of the given derivations. □

We can now extend Theorems 3.1 and 3.2 to relate sequent derivations with cut to natural deductions with explicit lemmas.

Theorem 3.6 (Soundness of Sequent Derivations with Cut)
If $\Gamma; \Delta \stackrel{+}{\Longrightarrow} A$ *then* $\Gamma; \Delta \vdash^{+} A\!\uparrow$.

Proof: As in Theorem 3.1 by induction on the structure of the derivation of $\Gamma; \Delta \stackrel{+}{\Longrightarrow} A$. An inference with one of the new rules cut or $\mathsf{cut!}$ is translated into an application of the $\uparrow\downarrow$ coercion followed by an appeal to one of the substitution principles in Property 3.5. □

Theorem 3.7 (Completeness of Sequent Derivations with Cut)

1. *If* $\Gamma; \Delta \vdash^{+} A\!\uparrow$ *then there is a sequent derivation of* $\Gamma; \Delta \stackrel{+}{\Longrightarrow} A$, *and*

2. *if* $\Gamma; \Delta \vdash^{+} A\!\downarrow$ *then for any formula* C *and derivation of* $\Gamma; (\Delta', A) \stackrel{+}{\Longrightarrow} C$ *there is a derivation of* $\Gamma; (\Delta', \Delta) \stackrel{+}{\Longrightarrow} C$.

Proof: As in the proof of Theorem 3.2 by induction on the structure of the given derivations. In the new case of the $\uparrow\downarrow$ coercion, we use the rule of cut. The other new rule, $\mathsf{cut!}$, is not needed for this proof, but is necessary for the proof of admissibility of cut in the next section. □

3.4 Cut Elimination

We viewed the sequent calculus as a calculus of proof search for natural deduction. The proofs of the soundness theorems 3.2 and 3.7 provide ways to translate cut-free sequent derivations into normal natural deductions, and sequent derivations with cut into arbitrary natural deductions.

This section is devoted to showing that the two rules of cut are redundant in the sense that any derivation in the sequent calculus which makes use of the rules of cut can be translated to one that does not. Taken together with the soundness and completeness theorems for the sequent calculi with and without cut, this has many important consequences.

First of all, a proof search procedure which looks only for cut-free sequent derivations will be complete: any derivable proposition can be proven this way. When the cut rule

$$\frac{\Gamma; \Delta \stackrel{+}{\Longrightarrow} A \qquad \Gamma; (\Delta', A) \stackrel{+}{\Longrightarrow} C}{\Gamma; \Delta', \Delta \stackrel{+}{\Longrightarrow} C}\,\mathsf{cut}$$

is viewed in the bottom-up direction the way it would be used during proof search, it introduces a new and arbitrary proposition A. Clearly, this introduces a great amount of non-determinism into the search. The cut elimination theorem now tells us that we never need to use this rule. All the remaining rules have the property that the premises contain only instances of propositions in the conclusion, or parts thereof. This latter property is often called the *subformula property*.

Secondly, it is easy to see that the logic is *consistent*, that is, not every proposition is provable. In particular, the sequent $\cdot; \cdot \Longrightarrow \mathbf{0}$ does not have a cut-free derivation, because there is simply no rule which could be applied to infer it! This property clearly fails in the presence of cut: it is *prima facie* quite possible that the sequent $\cdot; \cdot \stackrel{+}{\Longrightarrow} \mathbf{0}$ is the conclusion of the cut rule.

Along the same lines, we can show that a number of propositions are *not derivable* in the sequent calculus and therefore not true as defined by the natural deduction rules. Examples of this kind are given at the end of this section.

We prove cut elimination by showing that the two cut rules are *admissible rules of inference* in the sequent calculus without cut. An inference rule is admissible if whenever we can find derivations for its premises we can find a derivation of its conclusion. This should be distinguished from a *derived rule of inference* which requires a direct derivation of the conclusion from the premises. We can also think of a derived rule as an evident hypothetical judgment where the premises are (unrestricted) hypotheses.

Derived rules of inference have the important property that they remain evident under any extension of the logic. An admissible rule, on the other hand, represents a global property of the deductive system under consideration and may well fail when the system is extended. Of course, every derived rule is also admissible.

Theorem 3.8 (Admissibility of Cut)

1. *If* $\Gamma; \Delta \Longrightarrow A$ *and* $\Gamma; (\Delta', A) \Longrightarrow C$ *then* $\Gamma; (\Delta, \Delta') \Longrightarrow C$.
2. *If* $\Gamma; \cdot \Longrightarrow A$ *and* $(\Gamma, A); \Delta' \Longrightarrow C$ *then* $\Gamma; \Delta' \Longrightarrow C$.

Proof: By nested inductions on the structure of the cut formula A and the given derivations, where induction hypothesis (1) has priority over (2). To state this more precisely, we refer to the given derivations as $\mathcal{D} :: (\Gamma; \Delta \Longrightarrow A)$, $\mathcal{D}' :: (\Gamma; \cdot \Longrightarrow A)$, $\mathcal{E} :: (\Gamma; (\Delta, A) \Longrightarrow C)$, and $\mathcal{E}' :: ((\Gamma, A); \Delta' \vdash C)$. Then we may appeal to the induction hypothesis whenever

a. the cut formula A is strictly smaller, or

b. the cut formula A remains the same, but we appeal to induction hypothesis (1) in the proof of (2) (but when we appeal to (2) in the proof of (1) the cut formula must be strictly smaller), or

c. the cut formula A and the derivation $\mathcal{E}$ remain the same, but the derivation $\mathcal{D}$ becomes smaller, or

d. the cut formula A and the derivation $\mathcal{D}$ remain the same, but the derivation $\mathcal{E}$ or $\mathcal{E}'$ becomes smaller.

Here, we consider a formula smaller it is an immediate subformula, where $[t/x]A$ is considered a subformula of $\forall x.\ A$, since it contains fewer quantifiers and logical connectives. A derivation is smaller if it is an immediate subderivation, where we allow weakening by additional unrestricted hypothesis in one case (which does not affect the structure of the derivation).

The cases we have to consider fall into 5 classes:

Initial Cuts: One of the two premises is an initial sequent. In these cases the cut can be eliminated directly.

Principal Cuts: The cut formula A was just inferred by a right rule in $\mathcal{D}$ and by a left rule in $\mathcal{E}$. In these cases we appeal to the induction hypothesis (possibly several times) on smaller cut formulas (item (a) above).

Copy Cut: The cases for the $\mathsf{cut!}$ rule are treated as right commutative cuts (see below), except for the rule of dereliction which requires an appeal to induction hypothesis (1) with the same cut formula (item (b) above).

Left Commutative Cuts: The cut formula A is a side formula of the last inference in $\mathcal{D}$. In these cases we may appeal to the induction hypotheses with the same cut formula, but smaller derivation $\mathcal{D}$ (item (c) above).

Right Commutative Cuts: The cut formula A is a side formula of the last inference in $\mathcal{E}$. In these cases we may appeal to the induction hypotheses with the same cut formula, but smaller derivation $\mathcal{E}$ or $\mathcal{E}'$ (item (d) above).[1]

[1] [*some cases to be filled in*]

□

Using the admissibility of cut, the cut elimination theorem follows by a simple structural induction.

Theorem 3.9 (Cut Elimination)
If $\Gamma; \Delta \stackrel{+}{\Longrightarrow} C$ *then* $\Gamma; \Delta \Longrightarrow C$.

Proof: By induction on the structure of $\mathcal{D} :: (\Gamma; \Delta \stackrel{+}{\Longrightarrow} C)$. In each case except cut or cut! we simply appeal to the induction hypothesis on the derivations of the premises and use the corresponding rule in the cut-free sequent calculus. For the cut and cut! rules we appeal to the induction hypothesis and then admissibility of cut (Theorem 3.8) on the resulting derivations. □

3.5 Consequences of Cut Elimination

The first and most important consequence of cut elimination is that every natural deduction can be translated to a normal natural deduction. The necessary construction is implicit in the proofs of the soundness and completeness theorems for sequent calculi and the proofs of admissibility of cut and cut elimination. In Chapter **??** we will see a much more direct, but in other respects more complicated proof.

Theorem 3.10 (Normalization for Natural Deductions)
If $\Gamma; \Delta \vdash A$ *then* $\Gamma; \Delta \vdash A\uparrow$.

Proof: Directly, using theorems from this chapter.

$\Gamma; \Delta \vdash A$	Assumption
$\Gamma; \Delta \vdash^{+} A$	By Theorem 3.3
$\Gamma; \Delta \stackrel{+}{\Longrightarrow} A$	By completeness of sequent derivations with cut (Theorem 3.7)
$\Gamma; \Delta \Longrightarrow A$	By cut elimination (Theorem 3.9)
$\Gamma; \Delta \vdash A\uparrow$	By soundness of cut-free sequent derivations (Theorem 3.1)

□

As a second consequence, we see that linear logic is *consistent*: not every proposition can be proved. A proof of consistency for both intuitionistic and classical logic was Gentzen's original motivation for the development of the sequent calculus and his proof of cut elimination.

Theorem 3.11 (Consistency of Intuitionistic Linear Logic)
$\cdot; \cdot \vdash \mathbf{0}$ *true is not derivable.*

Proof: If the judgment were derivable, by Theorems 3.3, 3.7, and 3.9, there must be a cut-free sequent derivation of $\cdot; \cdot \Longrightarrow \mathbf{0}$. But there is no rule with which we could infer this sequent (there is no right rule for $\mathbf{0}$), and so it cannot be derivable. □

A third consequence is called the *disjunction property*. Note that in ordinary classical logic this property fails.

Theorem 3.12 (Disjunction Property for Intuitionistic Linear Logic) *If* $\cdot;\cdot \vdash A \oplus B$ *true then either* $\cdot;\cdot \vdash A$ *true or* $\cdot;\cdot \vdash B$ *true.*

Proof: Assume $\cdot;\cdot \vdash A \oplus B$ *true*. Then, by completeness of the cut-free sequent calculus, $\cdot;\cdot \Longrightarrow A \oplus B$. But there are only two rules that end with this judgment: $\oplus R_1$ and $\oplus R_2$. Hence either $\cdot;\cdot \Longrightarrow A$ or $\cdot;\cdot \Longrightarrow B$. Therefore, by soundness of the sequent calculus, $\cdot;\cdot \vdash A$ *true* or $\cdot;\cdot \vdash B$ *true* □

Note that these theorems are just special cases, and many other properties of the connectives follow from normalization and cut elimination.

As another kind of example, we can show that various propositions are *not* theorems of linear logic. Consider

$$\cdot; A \multimap (B \otimes C) \vdash (A \multimap B) \otimes (A \multimap C)$$

Intuitively, this should clearly not hold for arbitrary A, B, and C (although it could be true for some specific ones). But if we know the completeness of the cut-free sequent calculus this is easy to show. Consider

$$\cdot; A \multimap (B \otimes C) \Longrightarrow (A \multimap B) \otimes (A \multimap C).$$

There are only two possible rules that could have been used to deduce this conclusion, $\multimap R$ and $\otimes R$.

In case the last rule is $\multimap R$, one of the premises will be

$$\cdot;\cdot \Longrightarrow A$$

which is not provable for arbitrary A. In case the last rule is $\otimes R$, the linear hypothesis must be propagated to the left or right premise. Assume it goes to the left (the other case is symmetric). Then the right premise must be

$$\cdot;\cdot \Longrightarrow A \multimap C$$

which could only be infered by $\multimap R$, which leaves

$$\cdot; A \Longrightarrow C.$$

Again, unless we know more about A and C no rule applies. Hence the judgment above has no proof.

3.6 Another Example: The π-Calculus

The π-calculus was designed by Milner as a foundational calculus to investigate properties of communicating and mobile systems [**?**]. The first formulation below differs slightly from Milner's in the details of specification, but one can also give a completely faithful representation as shown in our second version.[2]

The basic syntactic categories in the π-calculus are *names*, *actions*, and *processes*. Names are simply written as variables x, y or a. They serve simultaneously as communication channels and the data that is transmitted along the channels. They constitute the only primitive data objects in the π-calculus, which makes it somewhat tedious to write non-trivial examples. In this sense it is similar to Church's pure λ-calculus, which was designed as a pure calculus of functions in which other data types such as natural numbers can be encoded.

Action prefixes π define the communication behavior of processes. We have

$$\begin{array}{rcll} \pi & ::= & x(y) & \text{receive } y \text{ along } x \\ & | & \overline{x}\langle y\rangle & \text{send } y \text{ along } x \\ & | & \tau & \text{unobservable (internal) action} \end{array}$$

Process expressions P define the syntax of processes in the π-calculus. They rely on *sums* M, which represent a non-deterministic choice between processes waiting to perform an action (either input, output, or an internal action).

$$\begin{array}{rcll} P & ::= & M & \text{sum} \\ & | & \mathsf{0} & \text{termination} \\ & | & P_1 \mid P_2 & \text{composition} \\ & | & \mathsf{new}\ a\ P & \text{restriction} \\ & | & !P & \text{replication} \\ M & | & M_1 + M_2 & \text{choice} \\ & | & \pi.\ P & \text{guarded process} \end{array}$$

Milner now defines a *structural congruence* that identifies process expressions that are only distinguished by the limitations of syntax. For example, the process composition operator $P \mid Q$ should be commutative and associative so that a collection of concurrent processes can be written as $P_1 \mid \cdots \mid P_n$. Similarly, sums $M + N$ should be commutative and associative. $\mathsf{0}$ is the unit of composition so that a terminated process simply disappears.

Names require that we add the renaming of bound variables to our structural congruence. In particular, $\mathsf{new}\ a\ P$ binds a in P and $x(y).\ P$ binds y in P. Note that, conversely, $\overline{x}\langle y\rangle.\ P$ does *not* bind any variables: the name y is just sent along x. The order of consecutive bindings by $\mathsf{new}\ a$ may be changed, and we can extend or contract the scope of a $\mathsf{new}\ a$ binder across process composition as follows:

$$\mathsf{new}\ x\ (P \mid Q) \equiv P \mid (\mathsf{new}\ x\ Q)$$

provided x is not among the free names of P. This law of *scope extrusion* (read right to left) is important this it means a process can propagate a local names to its environment.

[2] [*None of the material in this example has been proven correct at this time. Nor have we carefully surveyed the literature such as [**?**, **?**].*]

Finally, we have a rule of replication $!P \equiv P \mid !P$. Read from left to right it means a process $!P$ can replicate itself arbitrarily many times. From right to left the rule is of somewhat questionable value, since it would require recognizing structural equivalence of two active process expressions and then contracting them.

We will not formally model structural equivalence, because its necessary aspects will be captured by properties of the linear context Δ that contains active process expressions. Instead of repeating Milner's formal definition of the reaction rules, we explain them through their encoding in linear logic. The idea is the state of a process is represented by two proposition $\mathsf{proc}(P)$ for a process P and $\mathsf{choice}(M)$ for a sum M. A linear context

$$\Delta = \mathsf{proc}(P_1), \ldots, \mathsf{proc}(P_n), \mathsf{choice}(M_1), \ldots, \mathsf{choice}(M_m)$$

represents a state where processes P_i are executing concurrently and choices M_j are waiting to be made. Furthermore an unrestricted context

$$\Gamma = \mathsf{proc}(Q_1), \ldots, \mathsf{proc}(Q_p)$$

represents processes Q_k that may replicate themselves an arbitrary number of times. Informally, computation is modelled *bottom-up* in the sequent calculus, so that

$$\begin{array}{c} \Gamma_\pi, \Gamma_1; \Delta_1 \Longrightarrow C \\ \vdots \\ \Gamma_\pi, \Gamma_0; \Delta_0 \Longrightarrow C \end{array}$$

if we can transition from state Δ_0 with replicating processes Γ_0 to a state Δ_1 with replicating processes Γ_1. Here, C is arbitrary (in some sense, computation never stops) and Γ_π are the rules describing the legal reactions of the π-calculus as given below.

Process Composition $(P \mid Q)$**.** This just corresponds to a *fork* operation that generates two concurrently operating processes P and Q.

$$\mathsf{fork} : \mathsf{proc}(P \mid Q) \multimap \mathsf{proc}(P) \otimes \mathsf{proc}(Q)$$

Termination $\mathsf{0}$**.** This just corresponds to an *exit* operation, elimination the process.

$$\mathsf{exit} : \mathsf{proc}(\mathsf{0}) \multimap \mathbf{1}$$

Restriction $\mathsf{new}\ a\ P(a)$**.** The notation $P(a)$ represents a process P with some arbitrary number of occurrences of the bound variable a. We then write $P(x)$ for the result of substituting x for *all* occurrences of a in P. The new operation simply creates a new name, x, substitutes this for a in $P(a)$, and continues with $P(x)$. The freshness condition on x can be enforced easily by the corresponding condition on the left rule for the existential quantifier $\exists$L in the sequent calculus.

$$\mathsf{gen} : \mathsf{proc}(\mathsf{new}\ a\ P(a)) \multimap \exists x.\ \mathsf{proc}(P(x))$$

While this is not completely formal at present, once we introduce the concept of *higher-order abstract syntax* in Section **??** we see that it can easily be modeled in linear logic.

Replication $!P$. This just moves the process into the unrestricted context so that as many copies of P can be generated by the use of the copy rule as needed.

$$\mathsf{promote} : \mathsf{proc}(!P) \multimap !\mathsf{proc}(P)$$

Coincidentally, this is achieved in linear logic with the "*of course*" modality that is also written as "!".

Sum M. A process expression that is a sum goes into a state where it can perform an action, either silent (τ) or by a reaction between input and output processes.

$$\mathsf{suspend} : \mathsf{proc}(M) \multimap \mathsf{choice}(M)$$

It is now very tempting to define choice simply as internal choice. That is,

$$?\mathsf{choose} : \mathsf{choice}(M_1 + M_2) \multimap \mathsf{choice}(M_1) \& \mathsf{choice}(M_2)$$

However, this does not correspond to semantics of the π-calculus. Instead, $M_1 + \cdots + M_n$ can perform an action if

1. either one of the M_i can perform a silent action τ in which case all alternatives M_j for $j \neq i$ are discarded,
2. or two guarded actions $x(y).\ P(y)$ and $\overline{x}\langle z\rangle.\ Q$ react, leaving processes $P(z)$ and Q while discarding all other alternatives.

We model this behavior with two auxiliary predicates $\mathsf{react}(M, N, P, Q)$ which is true if sums M and N can react, leaving processes P and Q, and $\mathsf{silent}(M, P)$ which is true if M can make a silent transition to P. These are invoked non-deterministically as follows:

$$\begin{array}{lcl} \mathsf{external} & : & \mathsf{choice}(M) \otimes \mathsf{choice}(N) \otimes !\mathsf{react}(M, N, P, Q) \multimap \mathsf{proc}(P) \otimes \mathsf{proc}(Q) \\ \mathsf{internal} & : & \mathsf{choice}(M) \otimes !\mathsf{silent}(M, P) \multimap \mathsf{proc}(P) \end{array}$$

Note the use of "!" before the react and silent propositions which indicates that proofs of these propositions do not refer to the current process state.

Reaction. Basic reaction is synchronous communication along a channel x. This is augmented by a rule to choose between alternatives.

$$\begin{array}{lcl} \mathsf{synch} & : & \mathsf{react}(x(y).\ P(y), \overline{x}\langle z\rangle.\ Q, P(z), Q) \\ \mathsf{choose}_2 & : & \mathsf{react}(M, N, P, Q) \multimap (\mathsf{react}(M + M_0, N, P, Q) \\ & & \qquad \& \mathsf{react}(M_0 + M, N, P, Q) \\ & & \qquad \& \mathsf{react}(M, N + N_0, P, Q) \\ & & \qquad \& \mathsf{react}(M, N_0 + M, P, Q)) \end{array}$$

Note that the synchronization rule synch again employs our notation for substitution.

Silent Action. A basic silent action simply discards the guard τ. This is augmented by a rule to choose between alternatives.

$$\begin{array}{rcl} \mathsf{tau} & : & \mathsf{silent}(\tau.\ P, P) \\ \mathsf{choose}_1 & : & \mathsf{silent}(M, P) \multimap (\mathsf{silent}(M + M_0, P) \& \mathsf{silent}(M_0 + M, P)) \end{array}$$

That's it! To model Milner's notion of structural equivalence faithfully we would need at least one other rule

$$?\mathsf{collect} : !\mathsf{proc}(P) \otimes \mathsf{proc}(P) \multimap !\mathsf{proc}(P)$$

but this is of questionable merit and rather an artefact of overloading the notion of structural congruence with too many tasks.

3.7 Exercises

Exercise 3.1 Consider if $\otimes$ and $\&$ can be distributed over $\oplus$ or *vice versa*. There are four different possible equivalences based on eight possible entailments. Give sequent derivations for the entailments that hold.

Exercise 3.2 Prove that the rule

$$\frac{(\Gamma, A\&B, A, B); \Delta \Longrightarrow C}{(\Gamma, A\&B); \Delta \Longrightarrow C}\,\&L!$$

is admissible in the linear sequent calculus. Further prove that the rule

$$\frac{(\Gamma, A \otimes B, A, B); \Delta \Longrightarrow C}{(\Gamma, A \otimes B); \Delta \Longrightarrow C}\,\otimes L!$$

is *not* admissible.

Determine which other connectives and constants have similar or analogous admissible rules directly on resource factories and which ones do not. You do not need to formally prove admissibility or unsoundness of your proposed rules.

Exercise 3.3 In the proof of admissibility of cut (Theorem 3.8) show the cases where

1. $\mathcal{D}$ ends in $\multimap$R and $\mathcal{E}$ ends in $\multimap$L and we have a principal cut.

2. $\mathcal{D}$ is arbitrary and $\mathcal{E}$ ends in $\multimap$L and we have a a right commutative cut.

3. $\mathcal{D}$ ends in !R and $\mathcal{E}$ and in !L and we have a principal cut.

Exercise 3.4 Reconsider the connective $A \multimapboth B$ from Exercise 2.9 which is true if A linearly implies B and vice versa.

- Give sequent calculus rules corresponding to your introduction and elimination rules.

- Show the new cases in the proof of soundness of the sequent calculus (Theorem 3.1).

- Show the new cases in the proof of completeness of the sequent calculus (Theorem 3.2).

- Show the new cases for principal cuts in the proof of admissibility of cut (Theorem 3.8).

Exercise 3.5 An extension of the notion of Petri net includes *inhibitor arcs* as inputs to a transition. An inhibitor arc lets a transition fire only if the place it is connected to does not contain any tokens. Show how to extend or modify the encoding of Petri nets from Section 3.2 so that it also models inhibitor arcs.

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.

[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.

[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.

[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.