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

Focusing eliminates a lot of non-determinism regarding the choices among possible inference rules. We summarize again the essential improvements over the plain sequent calculus.

Inversion. Invertible rules are applied eagerly without considering alternatives. This does not jeopardize completeness precisely because these rules are (strongly) invertible. The introduction of ordered hypotheses Ω allows eliminates the don't-care non-determinism arises from the choice of invertible rule by fixing an order. It is an important property of the focusing system that in fact the order is irrelevant in the end, i.e., exchange between ordered hypotheses is admissible for the focusing calculus.

Focusing. Once a non-invertible rule has been selected, a sequence of non-invertible rules is applied to the focus proposition. This may involve don't-know non-deterministic choices, such as the question which side of a disjunction may be proven. However, other choices (such as interleaving left rules between independent linear hypotheses) are eliminated from consideration.

Initial Sequents. When we focus on a hypothesis, we break down the principal connective of the focus proposition until is has been reduced to either invertible or atomic. In case it is atomic, it must match an atomic goal—otherwise we fail. This improvement will be particularly important under the logic programming interpretation of linear logic.

As listed at the end of the last section, this still leaves a significant amount of non-determinism which needs to be controlled during proof search. Instead of piling these improvements on top of each other, we consider each one separately. In a realistic application, such as general theorem proving, logic programming, model checking, many of these high-level optimizations may need to be applied simultaneously and their interaction considered.

4.3 Unification

We begin with a discussion of *unification*, a technique for eliminating existential non-determinism. When proving a proposition of the form $\exists x.\ A$ by its right rule in the sequent calculus, we must supply a term t and then prove $[t/x]A$. The domain of quantification may include infinitely many terms (such as the natural numbers), so this choice cannot be resolved simply by trying all possible terms t. Similarly, when we use a hypothesis of the form $\forall x.\ A$ we must supply a term t to substitute for x.

Fortunately, there is a better technique which is sound and complete for syntactic equality between terms. The basic idea is quite simple: we postpone the choice of t and instead substitute a new *existential variable* (often called *meta-variable* or *logic variable*) X for x and continue with the bottom-up construction of a derivation. When we reach initial sequents we check if there is a substitution for the existential variables such that the hypothesis matches the conclusion. If so, we apply this instantiation globally to the partial derivation and continue to search for proofs of other subgoals. Finding an instantiation for existential variables under which two propositions or terms match is called *unification*. It is decidable if a unifying substitution or *unifier* exists, and if so, we can effectively compute it in linear time. Moreover, we can do so with a minimal commitment and we do not need to choose between various possible unifiers.

Because of its central importance, unification has been thoroughly investigated. Herbrand [Her30] is given credit for the first description of a unification algorithm in a footnote of his thesis, but it was not until 1965 that it was introduced into automated deduction through the seminal work by Alan Robinson [Rob65, Rob71]. The first algorithms were exponential, and later almost linear [Hue76, MM82] and linear algorithms [MM76, PW78] were discovered. In the practice of theorem proving, generally variants of Robinson's algorithm are still used, due to its low constant overhead on the kind of problems encountered in practice. For further discussion and a survey of unification, see [Kni89]. We describe a variant of Robinson's algorithm.

Before we describe the unification algorithm itself, we relate it to the problem of proof search. For this we use a general method of *residuation*. It should be kept in mind that this is mostly a foundational tool and not necessarily a direct path to an implementation. We enrich the judgment $\Gamma; \Delta \overset{-}{\Longrightarrow} A$ by a *residual proposition* F such that

1. if $\Gamma; \Delta \overset{-}{\Longrightarrow} A$ then $\Gamma; \Delta \overset{-}{\Longrightarrow} A \setminus F$ and F is true, and

2. if $\Gamma; \Delta \overset{-}{\Longrightarrow} A \setminus F$ and F is true then $\Gamma; \Delta \overset{-}{\Longrightarrow} A$.

Generally, we cannot prove such properties directly by induction, but we need to generalize them, exhibiting the close relationship between the derivations of the sequents and residual formulas F.

Residual formulas F are amenable to specialized procedures such as unification, since they are drawn from a simpler logic or deductive system than the general propositions A. In practice they are often solved *incrementally* rather than collected throughout a derivation and only solved at the end. This is important for the early detection of failures during proof search. Incremental solution of residual formulas is the topic of Exercise **??**.

What do we need in the residual propositions so that existential choices and equalities between atomic propositions can be expressed? The basic proposition is one of equality between atomic propositions, $P_1 \doteq P_2$. We also have conjunction $F_1 \wedge F_2$, since equalities may be collected from several subgoals, and $\top$ if there are no residual propositions to be proven. Finally, we need the existential quantifier $\exists x.\, F$ to express the scope of existential variables, and $\forall x.\, F$ to express the scope of parameters introduced in a derivation. We add equality between terms, since it is required to describe the unification algorithm itself. We refer to the logic with these connectives as *unification logic*, defined via a deductive system.

$$\textit{Formulas} \quad F \quad ::= \quad P_1 \doteq P_2 \mid t_1 \doteq t_2 \mid F_1 \wedge F_2 \mid \top \mid \exists x.\, F \mid \forall x.\, F$$

The main judgment "F *is valid*", written $\models F$, is defined by the following rules, which are consistent with, but more specialized than the rules for these connectives in intuitionistic natural deduction (see Exercise **??**).

$$\frac{}{\models P \doteq P}\doteq\mathrm{I} \qquad\qquad \frac{}{\models t \doteq t}\doteq\mathrm{I}'$$

$$\frac{\models F_1 \qquad \models F_2}{\models F_1 \wedge F_2}\wedge\mathrm{I} \qquad\qquad \frac{}{\models \top}\top\mathrm{I}$$

$$\frac{\models [t/x]F}{\models \exists x.\, F}\exists\mathrm{I} \qquad\qquad \frac{\models [a/x]F}{\models \forall x.\, F}\forall\mathrm{I}^a$$

The $\forall\mathrm{I}^a$ rule is subject to the usual proviso that a is a new parameter not occurring in $\forall x.\, F$. There are no elimination rules, since we do not need to consider hypotheses of the form $\models F$, which is the primary reason for the simplicity of theorem proving in the unification logic.

We enrich the sequent calculus with residual formulas from the unification logic, postponing all existential choices. Recall that in practice we merge residuation and solution in order to discover unprovable residual formulas as soon as possible. This merging of the phases is not represented in our system.

Hypotheses. Initial sequents residuate an equality between its principal propositions. Any solution to the equation will unify P' and P, which means that this will translate to a correct application of the initial sequent rule in the original system.

$$\frac{}{\Gamma; P' \stackrel{-}{\Longrightarrow} P \setminus P' \doteq P}\mathsf{init} \qquad\qquad \frac{(\Gamma, A); (\Delta, A) \stackrel{-}{\Longrightarrow} C \setminus F}{(\Gamma, A); \Delta \stackrel{-}{\Longrightarrow} C \setminus F}\mathsf{copy}$$

Propositional Connectives. We just give a few sample rules for the connectives which do not involve quantifiers, since all of them simply propagate or combine unification formulas, regardless whether they are additive, multiplicative, or exponential.

$$\frac{\Gamma; \Delta, A \xrightarrow{=} B \setminus F}{\Gamma; \Delta \xrightarrow{=} A \multimap B \setminus F} \multimap\mathrm{R} \qquad \frac{\Gamma; \Delta_1 \xrightarrow{=} A \setminus F_1 \qquad \Gamma; \Delta_2, B \xrightarrow{=} C \setminus F_2}{\Gamma; \Delta_1, \Delta_2, A \multimap B \xrightarrow{=} C \setminus F_1 \wedge F_2} \multimap\mathrm{L}$$

$$\frac{}{\Gamma; \cdot \xrightarrow{=} \mathbf{1} \setminus \top} \mathbf{1}\mathrm{R} \qquad \frac{\Gamma; \Delta \xrightarrow{=} C \setminus F}{\Gamma; \Delta, \mathbf{1} \xrightarrow{=} C \setminus F} \mathbf{1}\mathrm{L}$$

Quantifiers. These are the critical rules. Since we residuate the existential choices entirely, the $\exists$R and $\forall$L rules instantiate a quantifier by a new *parameter*, which is existentially quantified in the residual formula in both cases. Similarly, the $\forall$R and $\exists$L rule introduce a parameter which is universally quantified in the residual formula.

$$\frac{\Gamma; \Delta \xrightarrow{=} [a/x]A \setminus [a/x]F}{\Gamma; \Delta \xrightarrow{=} \forall x.\, A \setminus \forall x.\, F} \forall\mathrm{R}^a \qquad \frac{\Gamma; \Delta, [a/x]A \xrightarrow{=} C \setminus [a/x]F}{\Gamma; \Delta, \forall x.\, A \xrightarrow{=} C \setminus \exists x.\, F} \forall\mathrm{L}^a$$

$$\frac{\Gamma; \Delta \xrightarrow{=} [a/x]A \setminus [a/x]F}{\Gamma; \Delta \xrightarrow{=} \exists x.\, A \setminus \exists x.\, F} \exists\mathrm{R}^a \qquad \frac{\Gamma; \Delta, [a/x]A \xrightarrow{=} C \setminus [a/x]F}{\Gamma; \Delta, \exists x.\, A \xrightarrow{=} C \setminus \forall x.\, A} \exists\mathrm{L}^a$$

The soundness of residuating equalities and existential choices in this manner is straightforward.

Theorem 4.4 (Soundness of Equality Residuation) *If* $\Gamma; \Delta \xrightarrow{=} A \setminus F$ *and* $\models F$ *then* $\Gamma; \Delta \xrightarrow{=} A$.

Proof: By induction on the structure of $\mathcal{R} :: (\Gamma; \Delta \xrightarrow{=} A \setminus F)$. We show the critical cases. Note how in the case of the $\exists$R rule the proof of $\models \exists x.\, F$ provides the essential witness term t.

Case: $\mathcal{R} = \dfrac{}{\Gamma; P' \xrightarrow{=} P \setminus P' \doteq P} \mathrm{I}.$

We know by assumption that $\models F$ which reads $\models P' \doteq P$. By inversion therefore $P' = P$ (since $\doteq$I is the only rule which applies to this judgment), and $\Gamma; P' \xrightarrow{=} P$ is a valid initial sequent.

Case: $\mathcal{R} = \dfrac{\begin{array}{c}\mathcal{R}_1 \\ \Gamma; \Delta \xrightarrow{=} [a/x]A_1 \setminus [a/x]F_1\end{array}}{\Gamma; \Delta \xrightarrow{=} \exists x.\, A_1 \setminus \exists x.\, F_1} \exists\mathrm{R}^a.$

By assumption, we have $\models \exists x.\, F_1$. By inversion, $\models [t/x]F_1$ for some t. By the proviso on the $\exists\mathrm{R}^a$ rule, $\mathcal{R}_1$ is parametric in a, so we can substitute

t for a in this derivation an obtain $[t/a]\mathcal{R}_1 :: (\Gamma; \Delta \stackrel{-}{\Longrightarrow} [t/x]A_1 \setminus [t/x]F_1)$. Applying the induction hypothesis to $[t/a]\mathcal{R}_1$ yields a $\mathcal{D}_1$ and we construct

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma; \Delta \stackrel{-}{\Longrightarrow} [t/x]A_1\end{array}}{\Gamma; \Delta \stackrel{-}{\Longrightarrow} \exists x.\ A_1}\,\exists\mathrm{R}$$

Case: $\mathcal{R} = \dfrac{\begin{array}{c}\mathcal{R}_1\\ \Gamma; \Delta \stackrel{-}{\Longrightarrow} [a/x]A_1 \setminus [a/x]F_1\end{array}}{\Gamma; \Delta \stackrel{-}{\Longrightarrow} \forall x.\ A_1 \setminus \forall x.\ F_1}\,\forall\mathrm{R}^a.$

By assumption, we have $\models \forall x.\ F_1$. By inversion, $\models [b/x]F_1$ for a new parameter b, and therefore also $\models [a/x]F_1$ by substitution. Hence we can apply the induction hypothesis to obtain a $\mathcal{D}_1$ and construct

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma; \Delta \stackrel{-}{\Longrightarrow} [a/x]A_1\end{array}}{\Gamma; \Delta \stackrel{-}{\Longrightarrow} \forall x.\ A_1}\,\forall\mathrm{R}^a$$

□

The opposite direction is more difficult. The desired theorem:

If $\Gamma; \Delta \stackrel{-}{\Longrightarrow} A$ *then* $\Gamma; \Delta \stackrel{-}{\Longrightarrow} A \setminus F$ *for some* F *with* $\models F$

cannot be proved directly by induction, since the premises of the two derivations are different in the $\exists$R and $\forall$L rules. However, one can be obtained from the other by substituting terms for parameters. Since this must be done simultaneously, we introduce a new notation.

$$\textit{Parameter Substitution} \quad \rho \quad ::= \quad \cdot \mid \rho, t/a$$

We assume all the parameters a substituted for by ρ are distinct to avoid ambiguity. We write $[\rho]A$, $[\rho]F$, and $[\rho]\Gamma$, for the result of applying the substitution ρ to a proposition, formula, or context, respectively.

Lemma 4.5 *If* $\Gamma; \Delta \stackrel{-}{\Longrightarrow} A$ *and* $[\rho]A' = A$, $[\rho]\Delta' = \Delta$, *and* $[\rho]\Gamma' = \Gamma$, *then* $\Gamma'; \Delta' \stackrel{-}{\Longrightarrow} A' \setminus F$ *for some* F *and* $\models [\rho]F$.

Proof: The proof proceeds by induction on the structure of $\mathcal{D} :: (\Gamma; \Delta \stackrel{-}{\Longrightarrow} A)$. We show only three cases, the second of which required the generalization of the induction hypothesis.

Case: $\mathcal{D} = \dfrac{}{\Gamma; P \stackrel{=}{\Longrightarrow} P}\, I$

and $[\rho]\Gamma' = \Gamma$, $[\rho]\Delta' = P$, and $[\rho]A' = P$. Therefore $\Delta' = P''$ with $[\rho]P'' = P$ and $A' = P'$ with $[\rho]P' = P$ and we construct

$$\frac{}{\Gamma'; P'' \stackrel{=}{\Longrightarrow} P' \setminus P'' \doteq P'}\, I \quad \text{and} \quad \frac{}{\models [\rho]P'' \doteq [\rho]P'} \doteq \mathrm{I}$$

Case: $\mathcal{D} = \dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma; \Delta \stackrel{=}{\Longrightarrow} [t/x]A_1\end{array}}{\Gamma; \Delta \stackrel{=}{\Longrightarrow} \exists x.\ A_1}\, \exists\mathrm{R}.$

We assumed $[\rho]A' = \exists x.\ A_1$, so $A' = \exists x.\ A'_1$ and $[\rho, t/a]([a/x]A'_1) = [t/x]A_1$ for a new parameter a. Since a is new, $[\rho, t/a]\Gamma' = [\rho]\Gamma'$ and similarly for Δ', so we can apply the induction hypothesis to $\mathcal{D}_1$ to obtain $\mathcal{R}_1$ and $\mathcal{U}_1$ and construct

$$\frac{\begin{array}{c}\mathcal{R}_1\\ \Gamma'; \Delta' \stackrel{=}{\Longrightarrow} [a/x]A'_1 \setminus [a/x]F_1\end{array}}{\Gamma'; \Delta' \stackrel{=}{\Longrightarrow} \exists x.\ A'_1 \setminus \exists x.\ F_1}\, \exists\mathrm{R}^a \quad \text{and} \quad \frac{\begin{array}{c}\mathcal{U}_1\\ \models [\rho, t/a]([a/x]F_1)\end{array}}{\models [\rho]\exists x.\ F_1}\, \exists\mathrm{I}.$$

Case: $\mathcal{D} = \dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma; \Delta \stackrel{=}{\Longrightarrow} [a/x]A_1\end{array}}{\Gamma; \Delta \stackrel{=}{\Longrightarrow} \forall x.\ A_1}\, \forall\mathrm{R}^a.$

We assume $[\rho]A' = \forall x.\ A_1$, so $A' = \forall x.\ A'_1$ and $[\rho, a/a']([a'/x]A'_1) = [a/x]A_1$ for an a' new in Γ', Δ' and $\forall x.\ A'_1$. We can then appeal to the induction hypothesis on $\mathcal{D}_1$ to obtain $\mathcal{R}_1$ and $\mathcal{U}_1$ and construct

$$\frac{\begin{array}{c}\mathcal{R}_1\\ \Gamma'; \Delta' \stackrel{=}{\Longrightarrow} [a'/x]A'_1 \setminus [a'/x]F_1\end{array}}{\Gamma'; \Delta' \stackrel{=}{\Longrightarrow} \forall x.\ A'_1 \setminus \forall x.\ F_1}\, \forall\mathrm{I}^{a'} \quad \text{and} \quad \frac{\begin{array}{c}\mathcal{U}_1\\ \models [\rho, a/a']([a'/x]F_1)\end{array}}{\models [\rho]\forall x.\ F_1}\, \forall\mathrm{I}^a.$$

□

Theorem 4.6 (Completeness of Equality Residuation) *If* $\Gamma; \Delta \stackrel{=}{\Longrightarrow} A$ *then* $\Gamma; \Delta \stackrel{=}{\Longrightarrow} A \setminus F$ *for some* F *and* $\models F$.

Proof: From Lemma 4.5 with $A' = A$, $\Delta' = \Delta$, $\Gamma' = \Gamma$, and ρ the identity substitution on the parameters in Γ, Δ, and A. □

Next we describe an algorithm for proving residuated formulas, that is, an algorithm for unification. We do this in two steps: first we solve the problem in the fragment without parameters and universal quantifiers and then we extend the solution to the general case.

There are numerous ways for describing unification algorithms in the literature. We describe the computation of the algorithm as the bottom-up search for the derivation of a judgment. We restrict the inference rules such that they are essentially deterministic, and the inference rules themselves can be seen as describing an algorithm. This algorithm is in fact quite close to the implementation of it in ML which is available together with these notes.[2]

In order to describe the algorithm in this manner, we need to introduce *existential variables* (often called *meta-variables* or *logic variables*) which are place-holders for the terms to be determined by unification. We use X to stand for existential variables.

The second concept we need is a *continuation*, which arises from the introduction rule for conjunction. This rule has two premises, which leaves the choice on how which premise to prove first when we work in a bottom-up fashion. Our algorithm commits to do the first conjunct first, but it has remember that the second conjunct remains to be proved. Equational formulas which have been postponed in this way are accumulated in the continuation, which is activated when there are no further equations to be solved. For now, a continuation is simply another formula denoted by S. Initially, we use $\top$ for S. Thus our main judgment describing the algorithm has the form "*F is satisfiable with continuation S*", written as $\models F \;/\; S$.

Continuations. The following rules introduce and manage the continuations.

$$\frac{\models F_1 \;/\; F_2 \wedge S}{\models F_1 \wedge F_2 \;/\; S}\wedge\mathrm{I} \qquad \frac{}{\models \top \;/\; \top}\top\mathrm{I}\top \qquad \frac{\models F \;/\; S}{\models \top \;/\; F \wedge S}\top\mathrm{I}\wedge$$

Existential Quantification. Existential variables are introduced for existential quantifiers. They must be new not only in F but also in S.

$$\frac{\models [X/x]F \;/\; S \quad X \text{ not in } F \text{ or } S}{\models \exists x.\, F \;/\; S}\exists\mathrm{I}$$

Despite the requirement on X to be new, the derivation of the premise is not parametric in X. That is, we cannot substitute an arbitrary term t for X in a derivation of the permiss and obtain a valid derivations, since the vr, rv, vv, and vv′ rules below require one or both sides of the equation to be an existential variable. Substituting for such a variables invalidates the application of these rules.

Predicate and Function Constants. An equation between the same function constant applied to arguments is decomposed into equations between the arguments. Unification fails if different function symbols are compared, but this

[2] [*perhaps at some point*]

is only indirectly reflected by an absence of an appropriate rule. Failure can also be explicitly incorporated in the algorithm (see Exercise **??**).

$$\frac{\models t_1 \doteq s_1 \wedge \cdots \wedge t_n \doteq s_n \;/\; S}{\models p(t_1, \ldots, t_n) \doteq p(s_1, \ldots, s_n) \;/\; S}\,\mathrm{pp} \qquad \frac{\models t_1 \doteq s_1 \wedge \cdots \wedge t_n \doteq s_n \;/\; S}{\models f(t_1, \ldots, t_n) \doteq f(s_1, \ldots, s_n) \;/\; S}\,\mathrm{rr}$$

These rules violate orthogonality by relying on conjunction in the premises for the sake of conciseness of the presentation. When f or p have no arguments, the empty conjunction in the premise should be read as $\top$.

Existential Variables. There are three rules for variables. We write r for terms of the form $f(t_1, \ldots, t_n)$. Existential variables always range over terms (and not propositions), so we do not need rules for equations of the form $X \doteq P$ or $P \doteq X$.

$$\frac{\models \top \;/\; [r/X]S \quad X \text{ not in } r}{\models X \doteq r \;/\; S}\,\mathrm{vr} \qquad \frac{\models \top \;/\; [r/X]S \quad X \text{ not in } r}{\models r \doteq X \;/\; S}\,\mathrm{rv}$$

These two rules come with the proviso that the existential variable X does not occur in the term t. This is necessary to ensure termination of these rules (when viewed as an algorithm) and to recognize formulas such as $\exists x.\, x \doteq f(x)$ as unprovable. This leaves equations of the form $X \doteq Y$ with to existential variables. We write two rules for this case to simplify the analysis.

$$\frac{\models \top \;/\; [Y/X]S}{\models X \doteq Y \;/\; S}\,\mathrm{vv} \qquad \frac{\models \top \;/\; S}{\models X \doteq X \;/\; S}\,\mathrm{vv}'$$

We now analyze these rules when viewed as an algorithm specification. First we observe that all rules have either no or one premise. Furthermore, for any judgment $\models F \;/\; S$ at most one rule is applicable, and in only one way (the choice of the new existential variable name X is irrelevant). Therefore these rules, when viewed as instructions for construction a derivation of a judgment $\models F \;/\; \top$ are deterministic, but may fail, in which case the formula is not provable.

Furthermore, the bottom-up search for a derivation of $\models F \;/\; S$ in this system will always terminate. The termination ordering involves five measures, ordered lexicographically as follows:

1. the number of free and quantified existential variables,
2. the number of predicate and function symbols,
3. the total number of logical symbols $\wedge$, $\top$, $\exists$ in F and S,
4. the number of logical symbols in F,
5. the number of equations.

This measure decreases in each rule:

$\wedge$I does not change (1)–(3) and decreases (4),

$\top$I$\top$ completes the search,

$\top$I$\wedge$ does not change (1)–(2) and decreases (3),

$\exists$I does not change (1)–(2) and decreases (3),

pp does not change (1) and decreases (2),

rr does not change (1) and decreases (2),

vr decreases (1) since X does not occur in r,

rv decreases (1) since X does not occur in r,

vv decreases (1), and

vv$'$ does not change (1)–(4) and decreases (5).

In some of these cases it is also possible that a measure of higher priority decreases (but never increases), preserving the strict decrease along the lexicographic ordering.

We also note that the continuation S is not completely general, but follows the grammar below.

$$\textit{Continuations} \quad S \quad ::= \quad \top \mid F \wedge S$$

In other words, it may be viewed as a stack of formulas. In the ML implementation, this stack is not represented explicitly. Instead we use the call stack of ML itself.

The desired soundness and completess theorems for this algorithm requires some generalizations based on substitutions for existential variables.

$$\textit{Ground Substitutions} \quad \theta \quad ::= \quad \cdot \mid \theta, t/X$$

We always assume that the terms t we assign to variables in substitutions do not contain existential variables. This assumption is reasonable, since we only use substitutions here to connect derivations for $\models F$ (which contains to existential variables) with derivations of $\models F' \,/\, S'$ (which contains existential variables).

Lemma 4.7 (Soundness Lemma for Unification) *If $\models F \,/\, S$ then there exists a ground substitution for the existential variables in F and S such that $\models [\theta]F$ and $\models [\theta]S$.*

Proof: By induction on the structure of $\mathcal{F} :: (\models F \,/\, S)$. □

The soundness theorem follows easily from this lemma.

Theorem 4.8 (Soundness of Unification) *If $\models F \;/\; \top$ and F contains no existential variables, then $\models F$.*

Proof: From Lemma 4.7 for $S = \top$ and $\theta = \cdot$. □

Lemma 4.9 (Completeness Lemma for Unification) *If $\models F$ and $\models S$, then for any formulas F', continuations S' and substitutions θ for the existential variables in F' and S' such that $F = [\theta]F'$ and $S = [\theta]S'$ we have $\models F \;/\; S$.*

Proof: By nested inductions on $\mathcal{F} :: (\models F)$ and $\mathcal{S} :: (\models S)$. This means that when we appeal to the induction hypothesis on a subderivation of $\mathcal{F}$, $\mathcal{S}$ may be larger. We distinguish cases for $\mathcal{F}$.

Case: $\mathcal{F} = \dfrac{}{\models \top}\,\top I$.

The we distinguish two subcases for $\mathcal{S}$. If $\mathcal{S}$ is $\top$I, the result is trivial by $\top$I$\top$. Otherwise

$$\mathcal{S} = \dfrac{\begin{array}{c}\mathcal{F}_1\\ \models F_1\end{array} \qquad \begin{array}{c}\mathcal{S}_2\\ \models S_2\end{array}}{\models F_1 \wedge S_2}\,\wedge I$$

where $S = F_1 \wedge S_2$ for some F_1 and S_2. Then

$\mathcal{F}_1' :: (\models F_1 \;/\; S_2)$	By ind. hyp. on $\mathcal{F}_1$ and $\mathcal{S}_2$
$\mathcal{F}' :: (\models \top \;/\; F_1 \wedge S_2)$	By $\top$I$\wedge$

Case: $\mathcal{F} = \dfrac{\begin{array}{c}\mathcal{F}_1\\ \models F_1\end{array} \qquad \begin{array}{c}\mathcal{F}_2\\ \models F_2\end{array}}{\models F_1 \wedge F_2}\,\wedge I$.

$\mathcal{F}_2' :: (\models F_2' \;/\; S')$	By ind. hyp. on $\mathcal{F}_2$ and $\mathcal{S}$
$\mathcal{S}_2 :: (\models F_2 \wedge S)$	By $\wedge$I from $\mathcal{F}_2$ and $\mathcal{S}$
$\mathcal{F}_1' :: (\models F_1' \;/\; F_2' \wedge S')$	By ind. hyp. on $\mathcal{F}_1$ and $\mathcal{S}_2$
$\mathcal{F}' :: (\models F_1' \wedge F_2' \;/\; S')$	By $\wedge$I from $\mathcal{F}_1'$.

Case: $\mathcal{F} = \dfrac{\begin{array}{c}\mathcal{F}_1\\ \models [t/x]F_1\end{array}}{\models \exists x.\, F_1}\,\exists I$.

$F' = \exists x.\, F_1'$ and $[\theta](\exists x.\, F_1') = \exists x.\, F_1$	By assumption
$[\theta, t/X]([X/x]F_1') = [t/x]F_1$ for X not in F' or S'	
$[\theta, t/X]S' = S$	Since X is new
$\mathcal{F}_1' :: (\models [X/x]F_1' \;/\; S')$	By ind. hyp. on $\mathcal{F}_1$ and $\mathcal{S}$
$\mathcal{F} :: (\models \exists x.\, F_1' \;/\; S')$	By $\exists$I

Case: $\mathcal{F} = \dfrac{}{\models t \doteq t} \doteq \mathrm{I}.$

Here we proceed by an auxiliary induction on the structure of t. By assumption $[\theta]F' = (t \doteq t)$, so we have t' and t'' such that $[\theta]t' = [\theta]t'' = t$. We distinguish cases on t' and t'', showing three. The remaining ones are similar.

Subcase: $t' = f(t'_1, \ldots, t'_n)$ and $t'' = f(t''_1, \ldots, t''_n)$, so also $t = f(t_1, \ldots, t_n)$.

$\models t'_n \doteq t''_n \;/\; S'$	By ind. hyp. on t_n and $\mathcal{S}$
$\mathcal{S}_n :: (\models t_n \doteq t_n \wedge S)$	By $\wedge$I from $\doteq$I and $\mathcal{S}$
$\models t'_{n-1} \doteq t''_{n-1} \;/\; t'_n \doteq t''_n \wedge S'$	By ind. hyp. on t_{n-1} and $\mathcal{S}_n$.
$\models t'_1 \doteq t''_1 \;/\; t'_2 \doteq t''_2 \wedge \cdots \wedge t'_n \doteq t''_n \wedge S'$	As above
$\models t'_1 \doteq t''_1 \wedge t'_2 \doteq t''_2 \wedge \cdots \wedge t'_n \doteq t''_n \;/\; S'$	by $\wedge$I
$\models f(t'_1, \ldots, t'_n) \doteq f(t''_1, \ldots, t''_n) \;/\; S'$	by rr.

Subcase: $t' = X$ and $t'' = r$ but contains X. This is impossible, since we assumed $[\theta]t' = [\theta]t'' = t$.

Subcase: $t' = X$ and $t'' = r$ does not contain X. Then $[\theta]([r/X]S') = [\theta]S' = S$ since $[\theta]r = [\theta]X = t$ and θ is a ground substitution. By distinguishing cases for $\mathcal{S}$ as for $F = \top$ above, we conclude

$\models \top \;/\; [r/X]S'$	
$\models X \doteq r \;/\; S'$	By rule vr

□

The completeness theorem follows easily from this lemma.

Theorem 4.10 (Completeness of Unification) *If* $\models F$ *(where* F *contains no existential variables) then* $\models F \;/\; \top$.

Proof: From Lemma 4.9 with $S = \top$, $S' = \top$, $F' = F$ and $\theta = \cdot$. □

The generalization of the algorithm above to account for universal quantifiers and parameters is not completely straightforward. The difficulty is that $\forall x.\ \exists y.\ y \doteq x$ is valid, while $\exists y.\ \forall x.\ y \doteq x$ is not. We show an attempt to derive the latter which must be ruled out somehow.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\models \top \;/\; [a/Y]\top}\,\top\mathrm{I}\top}{\models Y \doteq a \;/\; \top}\,\mathrm{vr}}{\models \forall x.\ Y \doteq x \;/\; \top}\,\forall\mathrm{I}^a}{\models \exists y.\ \forall x.\ y \doteq x \;/\; \top}\,\exists\mathrm{I}$$

In this derivation, the application of $\top\mathrm{I}\top$ is correct since $[a/Y]\top = \top$. The problem lies in the fact a is new in the application of the $\forall\mathrm{I}^a$ rule, but only because we have not instantiated Y with a yet, which is necessary to complete the derivation.

There are two ways to solve this problem. More or less standard in theorem proving is *Skolemization* which we pursue in Exercise **??**. The dual solution notes for each existential variable which parameters may occur in its substitution term. In the example above, Y was introduced at a point where a did not yet occur, so the substitution of a for Y should be rejected.

In order to describe this concisely, we add a *parameter context* Ψ to the judgment which lists distinct parameters.

$$\textit{Parameter Context} \quad \Psi \quad ::= \quad \cdot \mid \Psi, a$$

This step is analogous to the localization of the hypotheses and should be considered merely a change in notation, not an essential change in the judgment itself. We annotate each judgment with the parameter context and introduce the new judmgnet "*t is closed with respect to Ψ*", written as $\Psi \models t\,\mathrm{term}$. It is defined by the following rules.

$$\frac{}{\Psi_1, a, \Psi_2 \vdash a\,\mathrm{term}}\,\mathrm{parm} \qquad \frac{\Psi \vdash t_1\,\mathrm{term} \quad \cdots \quad \Psi \vdash t_n\,\mathrm{term}}{\Psi \vdash f(t_1, \ldots, t_n)\,\mathrm{term}}\,\mathrm{root}$$

We modify the validity judgment for unification formulas to guarantee this condition.

$$\frac{\Psi \vdash t\,\mathrm{term} \qquad \Psi \models [t/x]F}{\Psi \models \exists x.\, F}\,\exists\mathrm{I} \qquad \frac{\Psi, a \models [a/x]F}{\Psi \models \forall x.\, F}\,\forall\mathrm{I}^a$$

When an existential variable X is introduced during the search for a derivation of a unification formula, we annotate it with the parameter context so we keep track of the admissible substitutions for X.

$$\frac{\Psi \models [X_\Psi/x]F \;/\; S \quad X_\Psi \text{ not in } F \text{ or } S}{\Psi \models \exists x.\, F \;/\; S}\,\exists\mathrm{I}$$

Parameters are introduced in the rule for universal quantifiers as before.

$$\frac{\Psi, a \models [a/x]F \;/\; S}{\Psi \models \forall x.\, F \;/\; S}\,\forall\mathrm{I}^a$$

An equation $X_\Psi \doteq t$ could now be solved immediately, if all parameters of t are contained in Ψ and X does not occur in t. However, there is one tricky case. Consider the judgment

$$a \models X_\cdot \doteq f(Y_a) \wedge Y_a \doteq a \;/\; \top$$

where X cannot depend on any parameters and Y can depend on a. This should have no solution, since $X_\cdot$ would have to be equal to $f(a)$, which is not permissible. On the other hand,

$$a \models X_\cdot \doteq f(Y_a) \wedge Y_a \doteq c \;/\; \top$$

for a constant c has a solution where Y_a is c and $X_\cdot$ is $f(c)$. So when we process an equation $X_\Psi = t$ we need to restrict any variable in t so it can depend only on the parameters in Ψ. In the example above, we would substitute $Y'_\cdot$ for Y_a.

In order to describe the algorithm, we internalize the judgment $\Psi \vdash t\,\mathit{term}$ as a new formula, written as $t\,|_\Psi$. We define it as follows.

$$\frac{\Psi' \models \top \;/\; S \quad \text{if } a \text{ in } \Psi}{\Psi' \models a\,|_\Psi/\; S}\,|\,\mathrm{a} \qquad \frac{\Psi' \models t_1\,|_\Psi \wedge \cdots \wedge t_n\,|_\Psi/\; S}{\Psi' \models f(t_1, \ldots, t_n)\,|_\Psi/\; S}\,|\,\mathrm{f}$$

$$\frac{\Psi' \models \top \;/\; [Y_{\Psi_2 \cap \Psi_1}/Y_{\Psi_2}]S}{\Psi' \models Y_{\Psi_2}\,|_{\Psi_1}/\; S}\,|\,\mathrm{v}$$

Here, $\Psi_1 \cap \Psi_2$ denotes the intersection of the two contexts. In the rules for variables, this is invoked as follows.

$$\frac{\Psi' \models r\,|_\Psi/\; [r/X_\Psi]S \quad \text{where } X_\Psi \text{ not in } r}{\Psi' \models X_\Psi \doteq r \;/\; S}\,\mathrm{vr}$$

$$\frac{\Psi' \models r\,|_\Psi/\; [r/X_\Psi]S \quad \text{where } X_\Psi \text{ not in } r}{\Psi' \models r \doteq X_\Psi \;/\; S}\,\mathrm{vr}$$

where r stands for a term $f(t_1, \ldots, t_n)$ or a parameter a. The variable rules are modified similarly.

$$\frac{\Psi' \models Y_{\Psi_2}\,|_{\Psi_1}/\; [Y_{\Psi_2}/X_{\Psi_1}]S}{\Psi' \models X_{\Psi_1} \doteq Y_{\Psi_2} \;/\; S}\,\mathrm{vv} \qquad \frac{\Psi' \models \top \;/\; S}{\Psi' x \models X_\Psi \doteq X_\Psi \;/\; S}\,\mathrm{vv}'$$

The use of continuations introduces on final complication. Consider the case of $(\forall x.\; F_1) \wedge F_2$. Since we linearize bottom-up search the parameter context Ψ will contain the parameter introduced for x when F_2 is finally considered after F_1 has been solved. This introduces spurious dependencies. To prohibit those, we build *closures* consisting of a formula and its parameter context on the continuation stack.

$$\text{Continuations} \quad S \quad ::= \quad \top \mid \{\Psi, F\} \wedge S$$

The rules for continuations are modified as follows.

$$\frac{\Psi \models F_1 \;/\; \{\Psi, F_2\} \wedge S}{\Psi \models F_1 \wedge F_2 \;/\; S}\,\wedge\mathrm{I} \qquad \frac{}{\Psi \models \top \;/\; \top}\,\top\mathrm{I}\top \qquad \frac{\Psi \models F \;/\; S}{\Psi' \models \top \;/\; \{\Psi, F\} \wedge S}\,\top\mathrm{I}\wedge$$

The termination argument is only slightly more difficult, since the restriction operation is a structural recursion over the term r and does not increase the number of variables or equations.

The soundness and completeness theorems from above extend to the problem with parameters, but become more difficult. The principal new notion we need is an *admissible substitution* θ which has the property that for every existential variable X_Ψ, $\Psi \vdash [\theta]X_\Psi$ term (see Exercise **??**).

The ML implementation takes advantage of the fact that whenever a variable must be restricted, one of the two contexts is a prefix of the other. This is because every equation in a formula F lies beneath a path of possibly alternating quantifiers, a so-called *mixed quantifier prefix*. When we apply the rules above algorithmically, we instantiate each existentially quantified variable with a new free existential variable which depends on all parameters which were introduced for the universally quantified variables to its left. Clearly, then, for any two variables in the same equation, one context is a prefix of the other. Our ML implementation does take advantage of this observation by simplifying the intersection operation.

We can take this optimization a step further and only record with an integer (a kind of time stamp), which parameters an existential variable may depend on. This improves the efficiency of the algorithm even further, since we only need to calculate the minimum of two integers instead of intersecting two contexts during restriction. In the ML code for this class, we did not optimize to this extent.

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.

[Her30] Jacques Herbrand. Recherches sur la théorie de la démonstration. *Travaux de la Société des Sciences et de Lettres de Varsovic*, 33, 1930.

[Hue76] Gérard Huet. *Résolution d'équations dans des langages d'ordre* $1, 2, \ldots, \omega$. PhD thesis, Université Paris VII, September 1976.

[Kni89] Kevin Knight. Unification: A multi-disciplinary survey. *ACM Computing Surveys*, 2(1):93–124, March 1989.

[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.

[MM76] Alberto Martelli and Ugo Montanari. Unification in linear time and space: A structured presentation. Internal Report B76-16, Ist. di Elaborazione delle Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy, July 1976.

[MM82] Alberto Martelli and Ugo Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems*, 4(2):258–282, April 1982.

[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.

[PW78] M. S. Paterson and M. N. Wegman. Linear unification. *Journal of Computer and System Sciences*, 16(2):158–167, April 1978.

[Rob65] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, January 1965.

[Rob71] J. A. Robinson. Computational logic: The unification computation. *Machine Intelligence*, 6:63–72, 1971.

[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.