Review

• Union/Find in OCaml
• Review of Homomorphisms
• Equality
• Congruence Closure

Outline

• Rewriting
• Termination
• Completion

Sources:
Harrison, John. *Introduction to Logic and Automated Theorem Proving*. Unpublished manuscript. Used by permission.

Rewriting

From now on, when we write \( \Gamma \models \phi \), it means that \( \phi \) holds in all normal models satisfying each formula in \( \Gamma \).

Consider again the general problem of establishing \( E \models s = t \).

Congruence closure handles the case when all equations are ground.

There cannot be a simple procedure for the more general case because first order logic with equality is, in general, undecidable.

However, often the kind of equational reasoning done in mathematics is straightforward: equations are used in a predictable direction to simplify expressions.

Using equations in a directional fashion is called rewriting, and there are indeed cases when this technique gives us a decision procedure.
Rewriting

Suppose \( t \) is a term and \( l = r \) is an equation. We say that \( t' \) results from rewriting \( t \) with \( l = r \) iff there is a subterm \( s \) of \( t \) and an instantiation \( \theta \) so that \( s = [\theta]l, s' = [\theta]r \) and \( t' \) is the result of replacing an instance of \( s \) by \( s' \) in \( t \).

Given a set \( R \) of rewrite rules, we write \( t \rightarrow_R t' \) iff there is some equation \((l = r) \in R \) which rewrites \( t \) to \( t' \).

\[ \begin{align*}
&\text{Rewriting} \\
&\text{Theorem (Soundness)} \\
&\text{If } t \rightarrow_R t', \text{ then } R \models t = t'.
\end{align*} \]

\[ \begin{align*}
&\text{Proof} \\
&\text{Every rewrite can be duplicated by a single instantiation followed by a chain of congruences.} \\
&\text{What about completeness?} \\
&\text{It depends on the rewrite rules.} \\
&A \text{ set of equations } R \text{ is said to constitute a canonical or convergent rewrite system when the question of whether } R \models s = t \text{ can be decided by rewriting.} \\
&\text{We will make this more precise later.}
\end{align*} \]

Abstract Reduction Relations

An abstract reduction relation is simply a binary relation on a set \( X \). We will denote a generic abstract reduction relation by \( \rightarrow \).

When we are talking about the specific case of the rewrite relation on terms induced by a set of equations \( R \), we will use the notation \( \rightarrow_R \).

\( \leftarrow \) is the inverse of \( \rightarrow \) (i.e. \( x \rightarrow y \) iff \( y \leftarrow x \)).

\( \rightarrow^+ \) is the transitive closure of \( \rightarrow \).

\( \rightarrow^* \) is the reflexive-transitive closure of \( \rightarrow \).

An element \( x \in X \) is said to be in normal form (NF) iff there is no \( y \in X \) such that \( x \rightarrow y \).

An abstract reduction relation is said to be terminating, strongly normalizing (SN), or noetherian iff there is no infinite reduction sequence:

\[ x_0 \rightarrow \cdots \rightarrow x_n \rightarrow \cdots \]

Note that \( \rightarrow \) is terminating iff \( \leftarrow \) is well-founded.

Confluence

\( \rightarrow \) has the \textit{diamond property} iff whenever \( x \rightarrow y \) and \( x \rightarrow y' \), there is a \( z \) such that \( y \rightarrow z \) and \( y' \rightarrow z \).

\( \rightarrow \) is \textit{confluent} or \textit{Church-Rosser} (CR) if \( \rightarrow^* \) has the diamond property.

\( \rightarrow \) is \textit{weakly confluent} or \textit{weakly Church-Rosser} (WCR) if whenever \( x \rightarrow y \) and \( x \rightarrow y' \), there is a \( z \) such that \( y \rightarrow^* z \) and \( y' \rightarrow^* z \).

These notions are closely related: the diamond property implies confluence which implies weak confluence.
Confluence

Weak confluence does not in general imply confluence, but adding termination changes the story.

Newman’s Lemma

If \( \longrightarrow \) is terminating and weakly confluent, then it is confluent.

Proof

It suffices to show that if \( x \rightarrow^* y \) and \( x \rightarrow^* y' \) with \( y \) and \( y' \) in normal form, then \( y = y' \). (Why?) This can be proved by well-founded induction. Suppose \( x \) is the minimal (according to \( \longrightarrow \)) element which rewrites to two different normal forms: \( y \) and \( y' \). We cannot have \( x = y \) or \( x = y' \), so suppose \( x \rightarrow w \rightarrow^* y \) and \( x \rightarrow w' \rightarrow^* y' \). By weak confluence, there must be a \( z \) such that \( w \rightarrow^* z \) and \( w' \rightarrow^* z \). But \( x \) is the minimal element for which confluence fails. Since \( w \) and \( w' \) are successors of \( x \), there must be a \( u \) such that \( y \rightarrow^* u \) and \( z \rightarrow^* u \), and a \( u' \) such that \( z \rightarrow^* u' \) and \( y' \rightarrow^* u' \). Similarly, there must be a \( z' \) such that \( y \rightarrow^* z' \) and \( u \rightarrow^* z' \). But \( y' \) and \( y' \) are in normal form, so \( y = u = z' = u' = y' \). \( \square \)

Rewriting

A simple implementation of rewriting is in rewrite.ml.

We have shown how a decision procedure is easily obtained from a terminating and weakly confluent rewrite system.

We now look at each of these properties more closely.

We begin with termination.

Canonical Rewrite Systems

Let \( \longrightarrow^*_R \) denote the reflexive-symmetric-transitive closure of \( \longrightarrow_R \).

Theorem

If \( R \) is a set of rewrites, then for all terms \( s \) and \( t \), \( s \leftarrow^*_R t \) iff \( R \models s = t \).

Let \( x \downarrow_R y \) denote that there exists a \( z \) such that \( x \rightarrow^*_R z \) and \( y \rightarrow^*_R z \). If \( x \downarrow_R y \), we say that \( x \) and \( y \) are joinable.

Theorem

If \( \longrightarrow_R \) is confluent, then for any \( x \) and \( y \), \( x \leftarrow^*_R y \) iff \( x \downarrow_R y \).

Corollary

If \( \longrightarrow_R \) is terminating and weakly confluent, then it is canonical (i.e. \( R \models s = t \) can be decided by rewriting \( s \) and \( t \) to normal forms and comparing them).

Proof

By Newman’s Lemma, termination and weak confluence imply confluence. Also, termination implies the existence of normal forms. Thus, by the above theorems, \( s \) and \( t \) have the same normal forms iff \( s \downarrow_R t \) iff \( R \models s = t \). \( \square \)

Termination Orderings

A binary relation \( > \) on terms is said to be a rewrite order if it is transitive and irreflexive and is closed under instantiation and simple congruences, i.e.

- It is never the case that \( t > t \).
- If \( s > t \) and \( t > u \), then \( s > u \).
- If \( s > t \), then \([\theta]s > [\theta]t\) for any instantiation \( \theta \).
- If \( s > t \), then
  \[ f(u_1, \ldots, u_{i-1}, s, u_{i+1}, \ldots, u_n) > f(u_1, \ldots, u_{i-1}, t, u_{i+1}, \ldots, u_n). \]

A rewrite order that is also well-founded is said to be a reduction order.

Lemma

If \( > \) is a reduction order and \( l > r \) for each equation \( l = r \) in \( R \), then the rewrite relation \( \longrightarrow_R \) is terminating.

Proof

It is not hard to see that if \( s \longrightarrow_R t \), then \( s > t \). Thus, because \( > \) is well-founded, \( \longrightarrow_R \) must be terminating. \( \square \)
Measure-based Orders

Let us denote by $|t|$ the number of variables and function symbols in $t$.

We might hope to define a reduction order $s > t$ by $|s| > |t|$. However, this fails the instantiation property:

If $s > t$, then $[\theta]s > [\theta]t$ for any instantiation $\theta$.

Example

$f(x, x, x) > g(x, y)$

but $[y := f(x, x, x)]f(x, x, x) < [y := f(x, x, x)]g(x, y)$.

What can we do to fix this?

Let $|t|_x$ denote the number of occurrences of $x$ in $t$.

Define $s > t$ if $|s| > |t|$ and $|s|_x \geq |t|_x$ for each free variable $x$ in $t$.

This defines a reduction ordering.

Code dealing with orderings is in order.ml.

Properties of the LPO

- If $s > t$ then $\text{free}(t) \subseteq \text{free}(s)$.

- The LPO is transitive.

- The LPO has the subterm property, i.e. if $t$ is a proper subterm of $s$, then $s > t$.

- The LPO is closed under instantiations: if $s > t$, then for any instantiation $\theta$, $[\theta]s > [\theta]t$.

- The LPO is a congruence with respect to the function symbols, i.e. if $s > t$, then $f(u_1, \ldots, u_{i−1}, s, u_{i+1}, \ldots, u_n) > f(u_1, \ldots, u_{i−1}, t, u_{i+1}, \ldots, u_n)$.

- The LPO is irreflexive ($t > t$ never holds).

- The LPO (over a finite set of function symbols) is well-founded.

Any rewrite order with the subterm property is called a simplification order.

Simplification orders are always well-founded.

Lexicographic Path Orders

The simple reduction ordering we defined fails for common kinds of equations, such as the associative and distributive laws:

- $(x * y) * z = x * (y * z)$
- $x * (y + z) = x * y + x * z$

For such rules, we can use a more sophisticated lexicographic ordering. A sequence $s_1, \ldots, s_m$ is lexicographically greater than a sequence $t_1, \ldots, t_m$ if there is some $1 \leq n \leq m$ such that $s_i = t_i$ for all $i < n$ and $s_n > t_n$.

The lexicographic path order (LPO) is defined as follows:

- $f(s_1, \ldots, s_m) > f(t_1, \ldots, t_m)$ if $s_1, \ldots, s_m$ is lexicographically greater than $t_1, \ldots, t_m$.
- $f(s_1, \ldots, s_m) > t$ whenever $s_i \geq t$.
- $f(s_1, \ldots, s_m) > g(t_1, \ldots, t_m)$ if $f > g$ according to some specified precedence ordering of the function symbols and $f(s_1, \ldots, s_m) > t_i$ for each $1 \leq i \leq n$.

Completion

Term orderings such as the LPO can be used to show that a rewrite system is terminating.

If we can in addition establish that the rewrite system is weakly confluent, we can conclude that it is confluent and hence canonical.

Example

Consider the group axioms:

- $(x * y) * z = x * (y * z)$
- $1 * x = x$
- $i(x) * x = 1$

Suppose we use these equations and some LPO to form a terminating rewrite system $R$. Is it confluent?

Consider the two terms $(1 * x) * y$ and $(i(x) * x) * y$.

The first can be rewritten to two terms which are joinable, but the second can be rewritten to different terms that are not joinable.

Thus, $R$ is not confluent.
Critical Pairs

In general, given termination, we can decide weak confluence (and hence confluence) by discovering whether any starting term \( s \) can be rewritten to different normal forms.

Suppose \( s \rightarrow_R t_1 \) and \( s \rightarrow_R t_2 \). There are three possible situations:

- The two rewrites apply to disjoint subterms, for example \((1 \cdot x) \cdot (i(y) \cdot y)\) to \(x \cdot (i(y) \cdot y)\) and to \( (1 \cdot x) \cdot 1 \).
- One rewrite applies to a term that is at or below a position corresponding to a variable in the other rewrite. For example, \((y \cdot z) \cdot (1 \cdot x)\) to \(y \cdot (z \cdot (1 \cdot x))\) or to \((y \cdot z) \cdot x\).
- One rewrite applies to a term that is inside the term to which the other rewrite applies, but is not at or below a variable position. For example, \((1 \cdot x) \cdot y\) to \(x \cdot y\) or \(1 \cdot (x \cdot y)\).

It is not hard to see that the first two cases cannot break weak confluence. Thus, it is only the third case that must be considered.

Critical Pairs

**Lemma**

Let \( l_1 = r_1 \) and \( l_2 = r_2 \) be two equations with no common variables. If \( s \rightarrow l_1 = r_1 \) \( t_1 \) and \( s \rightarrow l_2 = r_2 \) \( t_2 \) not joinable, then \( t_1 \) and \( t_2 \) differ only in two subterms \( u_1 \) and \( u_2 \) such that either \((u_1, u_2)\) or \((u_2, u_1)\) is an instance of a critical pair.

**Theorem**

A term rewriting system is weakly confluent iff all its critical pairs are joinable.

The code for computing and checking critical pairs is in `completion.ml`.

Critical Pairs

Suppose \( l_1 = r_1 \) and \( l_2 = r_2 \) are two rewrite rules (we assume \( \text{free}(l_1) \cap \text{free}(l_2) = \emptyset \)). If \( x_1 \) is a non-variable subterm of \( l_1 \) (it could be \( l_1 \) itself) and \( \theta \) is a most general unifier for \( x_1 \) and \( l_2 \), then \([\theta] r_1 \) and the result of replacing \([\theta] x_1 \) in \([\theta] l_1 \) by \([\theta] r_2 \) are a critical pair.

**Example**

What are the critical pairs for the group axioms?

1. \((x_1 \cdot y) \cdot z = x_1 \cdot (y \cdot z)\)
2. \(1 \cdot x_2 = x_2\)
3. \(i(x_3) \cdot x_3 = 1\)

1 and 2 with \( x_1 := 1 \) \( y := x_2 \) gives \((1 \cdot (x_2 \cdot z), x_2 \cdot z)\).

1 and 3 with \( x_1 := i(x_3), y := x_3 \) gives \((i(x_3) \cdot (x_3 \cdot z), 1 \cdot z)\).

1 and 1' with \( x_1 := x_1', y := y', z := z' \) gives
\((x_1' \cdot y'), (z' \cdot z') \cdot (x_1' \cdot (y' \cdot z')).\)

The first and third pairs are joinable, but the second is not. Thus this rewrite system is not weakly confluent.

Completion

It is straightforward to check whether each critical pair is joinable.

However, we can be more ambitious.

Suppose \((s, t)\) is a critical pair which is not joinable, that is, the normal form of \( s \) is \( s' \) and the normal form of \( t \) is \( t' \), and \( s' \neq t' \).

Instead of giving up, we can imagine adding \( s' = t' \) or \( t' = s' \) to our rewrite system to achieve confluence.

The process of repeatedly adding normalized critical pairs to the rewrite system is known as completion.

Two things can go wrong:

- It may not be possible to add \( s' = t' \) or \( t' = s' \) while respecting the term ordering.
- The completion process may run forever.

However, often completion is successful. We will continue looking at `completion.ml`.

Completion
Interreduction

Completion often results in a large set of rewrites. A natural question is whether the set can be reduced.

**Theorem**

Let \( \rightarrow_R \) be a canonical (i.e. terminating and confluent) abstract reduction relation on a set \( X \). Suppose another abstract reduction relation \( \rightarrow_S \) has the following two properties:

- For any \( x, y \in X \), if \( x \rightarrow_S y \), then \( x \rightarrow_R y \).
- For any \( x, y \in X \), if \( x \rightarrow_R y \), then there is a \( y' \in X \) with \( x \rightarrow_S y' \).

Then \( \rightarrow_S \) is also canonical and defines the same equivalence.

**Corollary** If \( R \) is a canonical rewrite system and \( (l = r) \in R \), then if \( l \) is reducible by the other equations, the system \( R - \{l = r\} \) is also canonical and defines the same equational theory.

**Corollary** If \( R \) is a canonical rewrite system and \( (l = r) \in R \), let \( S \) be the result of replacing the equation \( l = r \) in \( R \) with \( l = r' \) where \( r' \) is the \( R \)-normal form of \( r \). Then \( S \) is also canonical and defines the same equational theory.