Chained Infixed Formal Relations

Introduction and Terminology

In our discussion of the proof that $\sqrt{2}$ is irrational, I made the following observation:

> When we write e.g.
>
> $$2 = (\sqrt{2})^2 = \left(\frac{a}{b}\right)^2 = \frac{a^2}{b^2}$$
>
> this is just a shorthand for
>
> $$2 = (\sqrt{2})^2 \text{ and } (\sqrt{2})^2 = \left(\frac{a}{b}\right)^2 \text{ and } \left(\frac{a}{b}\right)^2 = \frac{a^2}{b^2}.$$ [a]
>
> When we come to providing semantics for compressed equations of this kind, we will break them apart into their 'longhand' versions and consider those individually; i.e. we will give semantics for each short equation like '$2 = (\sqrt{2})^2$' rather than the compressed whole.
>
> [a] This compression of formal relations actually occurs more generally, as in e.g.
>
> $$T_1 = T_2 < T_3 <= T_4$$
>
> there are specific restrictions on what formal relations can be chained in this way, but as they are not relevant to the proof at hand we will leave discussion of them to another time.

This note is an expansion of the comments made in the footnote just quoted, i.e. it is a discussion of the general compression of formal relations and restrictions thereon. Let us start by introducing a little terminology. Since there are mathematical objects called 'relations', we will use the metalanguage term *(infixed) formal relation* to refer to any sequence of tokens which may be placed between two terms so as to form a formula. So, for example, we have:

$$\underbrace{k}_{\text{Term}} \; \underset{\substack{\uparrow \\ \text{Formal Relation}}}{=} \; \underbrace{Ae^{-\frac{E_a}{RT}}}_{\text{Term}}$$

$$\underbrace{\left(\sum_{i=1}^{n} x_i y_i\right)^2}_{\text{Term}} \; \underset{\substack{\uparrow \\ \text{Formal Relation}}}{\leqslant} \; \underbrace{\left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} y_i^2\right)}_{\text{Term}}$$

Under certain conditions, to be discussed below, one may compress two or more formulae involving terms T_i and infixed formal relations R_i by writing

$$T_0 \; R_0 \; T_1 \; R_1 \; T_2 \; \ldots \; T_{n-1} \; R_{n-1} \; T_n$$

as a shorthand equivalent for

$$T_0\ R_0\ T_1 \text{ and } T_1\ R_1\ T_2 \text{ and } \dots \text{ and } T_{n-1}\ R_{n-1}\ T_n$$

We will refer to this compression as the *chaining* of infixed formal relations. A typical example is:[1]

$$\underbrace{|f(x) - f(x_0)|}_{\text{Term}} \underset{\substack{\uparrow\\ \text{Formal Relation}}}{\leqslant} \underbrace{|f(x) - f_n(x)| + |f_n(x) - f_n(x_0)| + |f_n(x_0) - f(x_0)|}_{\text{Term}} \underset{\substack{\uparrow\\ \text{Formal Relation}}}{<} \underbrace{\epsilon}_{\text{Term}}$$

standing as a shorthand for the conjunction of

$$\underbrace{|f(x) - f(x_0)|}_{\text{Term}} \underset{\substack{\uparrow\\ \text{Formal Relation}}}{\leqslant} \underbrace{|f(x) - f_n(x)| + |f_n(x) - f_n(x_0)| + |f_n(x_0) - f(x_0)|}_{\text{Term}}$$

and

$$\underbrace{|f(x) - f_n(x)| + |f_n(x) - f_n(x_0)| + |f_n(x_0) - f(x_0)|}_{\text{Term}} \underset{\substack{\uparrow\\ \text{Formal Relation}}}{<} \underbrace{\epsilon}_{\text{Term}}$$

.

Long chained sequences of this kind are often typeset over several lines, with each formal relations starting a new line and with such relations aligned vertically. For example:[2]

$$\begin{aligned} d(T^n x_0, x_0) &\leqslant d(T^n x_0, T^{n-1}x_0) + d(x_0, T^{n-1}x_0) \\ &\leqslant q^{n-1} d(Tx_0, x_0) + \frac{1-q^{n-1}}{1-q}a \\ &= \frac{q^{n-1} - q^n}{1-q}a + \frac{1-q^{n-1}}{1-q}a \\ &= \frac{1-q^n}{1-q}a \end{aligned}$$

$\uparrow$ Aligned Column of Formal Relations

Such chaining of former relations is utterly ubiquitous; it would be difficult to find any mathematical text that did not use this piece of sugar.

[1] Formula retrieved from http://planetmath.org/encyclopedia/ LimitOfAUniformlyConvergentSequenceOfContinuousFunctionsIsContinuous.html on 20 January 2008.

[2] Formula retrieved from http://planetmath.org/encyclopedia/ ProofOfBanachFixedPointTheorem.html on 20 January 2008.

Licensing

As noted above, it does not seem to be possible to chain arbitrary relations together; for example, although both $x < y < z$ and $x > y > z$ are perfectly legal, one is not allowed to write $x < y > z$. The general condition that determines whether or not a particular chain is valid is technical and may be hard to read, so we will start by looking at a much simpler special case:[3]

> **Ideal Licensing Condition for Chaining (Two Relations)**
> In a chained sequence with two relations, the first term must be related to the last term by one of the two relations in the chain.

So, for example, in:

$$|f(x) - f(x_0)| \leqslant |f(x) - f_n(x)| + |f_n(x) - f_n(x_0)| + |f_n(x_0) - f(x_0)| < \epsilon$$

the condition states that $|f(x) - f(x_0)|$ and ϵ must be related by one of $\leqslant$ and $<$; and, indeed, the transitivity of order relations means that

$$|f(x) - f(x_0)| < \epsilon$$

does hold.

In a longer sequence, the licensing condition is most easily described by a procedure. One starts by taking the initial segment of the chain consisting of three terms and two relations:

$$\underbrace{T_0\ R_0\ T_1\ R_1\ T_2}_{\text{Initial Segment of Chain}} \quad R_2\ T_3\ \ldots\ T_{n-1}\ R_{n-1}\ T_n$$

Then one tries to apply the simple criterion above to this sub-chain; in other words, one tries to determine whether T_0 and T_2 are related by R_0 or R_1. If this is not the case, then the complete chain is illegal. But if T_0 and T_2 are related, then one replaces $T_0\ R_0\ T_1\ R_1\ T_2$ by whichever of $T_0\ R_0\ T_2$ and $T_0\ R_1\ T_2$ holds.[4] Supposing that, say, $T_0\ R_0\ T_2$ holds, we would obtain:

$$\underbrace{T_0\ R_0\ T_2\ R_2\ T_3}_{\text{Initial Segment of New Chain}} \quad R_3\ T_4\ \ldots\ T_{n-1}\ R_{n-1}\ T_n$$

Then one repeats the process with the initial segment of this chain; that is, one tries to determine whether T_0 and T_3 are related by R_0 or R_2, etc. . If this process can be continued until one has something of the form $T_0\ R_i\ T_n$, then the chain is licensed.[5]

---

[3] We use the word *ideal* in the material below to refer to the version we would use if effective computability were not a concern. In the next section we will construct *actual* versions which are effectively (and even efficiently) computable.

[4] It is possible that both hold; so this procedure is nondeterministic. As we are not interested in implementational considerations here, this is not a cause for concern.

[5] It is generally only the fact that $T_0\ R_i\ T_n$ that is relevant to the rest of the proof. We could take the opportunity to pass a hint to this effect to the theorem prover. The fact that we do not know which R_i is in use is only a minor hindrance; since we need modal logic anyway, to cope with background objects, we can emit the presupposition

$$\Box(T_0\ R_0\ T_n) \vee \Box(T_0\ R_1\ T_n) \vee \ldots \vee \Box(T_0\ R_{n-1}\ T_n).$$

Turning this description into a formal condition takes a little work:

Ideal Licensing Condition for Chaining
We define a notion of *ideal collapsibility* inductively as follows. Any unchained formula $T_0\ R_0\ T_1$ is trivially collapsible to itself, i.e. to $T_0\ R_0\ T_1$. A chained formula

$$T_0\ R_0\ T_1\ R_1\ T_2\ \ldots\ T_{n-1}\ R_{n-1}\ T_n$$

is ideally collapsible to $T_0\ R\ T_n$ precisely when there is a formal relation R' such that $T_0\ R_0\ T_1\ \ldots\ T_{n-1}$ is ideally collapsible to $T_0\ R'\ T_{n-1}$ and it is the case that

$$\text{If } T_0\ R'\ T_{n-1} \text{ and } T_{n-1}\ R_{n-1}\ T_n\text{, then } T_0\ R\ T_n.$$

where $R = R'$ or R_{n-1}.

A chained formula is licensed precisely if it is ideally collapsible to some unchained formula.

Example

We would validate our long example of chaining,

$$\begin{aligned} d(T^n x_0, x_0) &\leqslant d(T^n x_0, T^{n-1}x_0) + d(x_0, T^{n-1}x_0) \\ &\leqslant q^{n-1}d(Tx_0, x_0) + \frac{1-q^{n-1}}{1-q}a \\ &= \frac{q^{n-1}-q^n}{1-q}a + \frac{1-q^{n-1}}{1-q}a \\ &= \frac{1-q^n}{1-q}a, \end{aligned}$$

using the following steps:

1. It is trivially the case that

$$d(T^n x_0, x_0) \leqslant d(T^n x_0, T^{n-1}x_0) + d(x_0, T^{n-1}x_0)$$

is collapsible to itself.

2. We now consider

$$\begin{aligned} d(T^n x_0, x_0) &\leqslant d(T^n x_0, T^{n-1}x_0) + d(x_0, T^{n-1}x_0) \\ &\leqslant q^{n-1}d(Tx_0, x_0) + \frac{1-q^{n-1}}{1-q}a \end{aligned}$$

By 1. and the transitivity of $\leqslant$, this is collapsible to

$$d(T^n x_0, x_0) \leqslant q^{n-1}d(Tx_0, x_0) + \frac{1-q^{n-1}}{1-q}a$$

3. We now need to determine whether

$$\begin{aligned} d(T^n x_0, x_0) &\leqslant d(T^n x_0, T^{n-1}x_0) + d(x_0, T^{n-1}x_0) \\ &\leqslant q^{n-1}d(Tx_0, x_0) + \frac{1-q^{n-1}}{1-q}a \\ &= \frac{q^{n-1}-q^n}{1-q}a + \frac{1-q^{n-1}}{1-q}a \end{aligned}$$

is collapsible. By 2. and the fact that

$$d(T^n x_0, x_0) \leqslant q^{n-1}d(Tx_0, x_0) + \frac{1-q^{n-1}}{1-q}a$$

and

$$q^{n-1}d(Tx_0, x_0) + \frac{1-q^{n-1}}{1-q}a = \frac{q^{n-1}-q^n}{1-q}a + \frac{1-q^{n-1}}{1-q}a$$

together imply

$$d(T^n x_0, x_0) \leqslant \frac{q^{n-1}-q^n}{1-q}a + \frac{1-q^{n-1}}{1-q}a$$

the quoted equation is indeed collapsible, to

$$d(T^n x_0, x_0) \leqslant \frac{q^{n-1}-q^n}{1-q}a + \frac{1-q^{n-1}}{1-q}a.$$

4. Finally, we need to determine whether the complete equation is collapsible. The working is essentially identical to the previous step.

Notes

Most instances of change turn out to be licensed by transitivity or properties of '=' as in the above example, but there are exceptions. For example, one may write an equation

$$a \in X \subseteq X^c$$

which is licensed because it implies that $a \in X^c$; this example is not related to transitivity in any way.

Parsing

The chaining of infixed formal relations is a piece of syntactic sugar and accordingly needs to be handled by the parser. This raises all of the usual complications related to type and parsing; to quote from an earlier document,

> ...we have set up an unusually circular system: syntax depends on type, and type depends on semantics, but (as in any other compositional system) semantics depends on syntax. Untangling this knot accounts for a significant part of the complexity of the parsing process.

In the present context, the issue is that the parser is not capable of deciding general semantic questions; that is, when it encounters, say,

$$a \in X \subseteq X^c$$

it cannot decide whether or not $a \in X^c$ or $a \subseteq X^c$ hold. So the parser cannot pursue a perfect strategy which admits precisely the licensed instances of chaining and rejects the others. Instead, it must approximate the answer using the restricted semantic information that is available to it, namely the information about type.

Given that we cannot parse the construct perfectly, the first issue that we face is this: broadly speaking, is it worse to admit some unlicensed examples or reject some licensed examples.[6] In this case, the ubiquity of the construction leads us to aim for a solution that errs on the wide side, i.e. that admits some unlicensed examples. We formalise this approach as follows:

> **(Actual) Licensing Condition for Chaining**
> We define a notion of *(actual) collapsibility* inductively as follows. Any unchained formula $T_0\ R_0\ T_1$ is trivially collapsible to itself, i.e. to $T_0\ R_0\ T_1$. A chained formula
>
> $$T_0\ R_0\ T_1\ R_1\ T_2\ \ldots\ T_{n-1}\ R_{n-1}\ T_n$$
>
> is actually collapsible to $T_0\ R\ T_n$ precisely when there is a formal relation R' such that $T_0\ R_0\ T_1\ \ldots\ T_{n-1}$ is collapsible to $T_0\ R'\ T_{n-1}$ and the system has *explicitly* seen the fact that
>
> $$\text{If } T_0'\ R'\ T_{n-1}' \text{ and } T_{n-1}'\ R_{n-1}\ T_n', \text{ then } T_0'\ R\ T_n'$$
>
> where $R = R'$ or R_{n-1}, and T_0', T_{n-1}', T_n' are some objects which have the same type as T_0, T_{n-1}, T_n respectively.
>
> A chained formula is (actually) licensed precisely if it is actually collapsible to some unchained formula.

We have effected two changes to produce this condition. First, by only checking collapsibility for *some objects* of the same types as those actually present, we have effectively taken the ideal licensing condition and lifted it up to a very

[6] With all questions of this kind, one must remember that the system may emit presuppositions to cause the theorem prover to double-check all the accepted instances and reject the duds.

coarse semantic grain, namely that of type;[7] this process is necessarily lossy, and we have chosen to err on the generous side.[8]

Second, by requiring that the system has *explicitly* seen the fact licensing chaining, we are narrowing the criterion further, and some licensed examples will not be caught. This is necessary, because parsing should not depend on the reasoning strength of the parser — i.e. a stronger reasoner should not be able to pick up more readings, for if it could, we could never be confident that we had found all the readings of a sentence. It is also to a degree desirable, in that it cuts down the frequency of spurious 'pseudo-chaining' parses. We believe that relatively few explicit assertions are needed, and so that the increased load on the user will not be too large.

Examples

The simplest examples are all licensed by assertions about transitivity; for example, after an explicit assertion that

> If A, B, C are sets such that $A \subset B$ and $B \subset C$, then $A \subset C$.

the system will accept chains of the form

$$X \subset Y \subset Z$$

for any sets X, Y, Z. In this case, the explicit assertion is type-universal so that the actual licensing condition and the ideal licensing condition coincide; no inappropriate examples are accidentally accepted by the parser. A more complex example arises from an explicit assertion that:

> If A, B, C are Von Neumann ordinals such that $A \in B$ and $B \in C$, then $A \in C$.

Because Von Neumann ordinals are sets, but not all sets are Von Neumann ordinals, this assertion is not type-universal; nevertheless it licenses all chains of the form

$$X \in Y \in Z$$

for *any* sets X, Y, Z.

[7] This is a slightly misleading formulation, in that type is not quite purely semantic; but I find this particular formulation more illuminating than a stricter alternative hedged with caveats. More generally, I have a strong belief that looking at type as a coarse-grained quotient of semantics is a very illuminating perspective on formal languages of all kinds — but this is not the place for that digression.

[8] The condition we have adopted is in fact very generous; it licences many illegal examples of chaining. Nevertheless, it rules out a huge amount compared to the simplest 'generous' solution, which is to allow chaining of any formal relations. We hope that the restriction is sufficient to prevent ambiguity arising from spurious 'pseudo-chaining' parses.

An Unresolved Issue

It is at least arguable that at some point in the development of mathematics, the standard ordering ‘$<$’ on the real numbers is reanalysed from being part of a rule to being an object in its own right, more specifically, a mathematical relation. If relations have the type ‘Set’, as most analyses suggest, the reanalysis is from

Formula → Term:Number $<$ Term:Number

to

Formula → Term:Object Term:Set Term:Object

We have not yet determined how to handle the induced interaction between reanalysis and chaining in examples like:

$$0 < 1 < 2 < 3$$