Newman's Lemma

a case study in proof automation

Marc Bezem
`www.ii.uib.no/~bezem`
Department of Computer Science
Bergen University
Norway

1 Introduction, summary and conclusions

A directed graph G is called *confluent* if any two diverging paths having the same start node can be extended into paths leading to the same end node. Directed graphs are ubiquitous, but often go by different names:

- abstract rewriting systems (underlying functional languages, etc.)
- labelled transition systems (underpinning operational semantics)
- Kripke frames (the basis of the semantics of modal logic)

In the respective cases, confluence is a key notion for:

- uniqueness of normal forms (independence of the evaluation strategy)
- certain forms of determinism (in operational semantics)
- quantifier shifts (in dynamic logic)

The graph G is called *locally confluent* if any two different *steps* (paths of length one) departing from the same node can be extended into paths leading to the same end node. As we will see in the next section, local confluence is strictly weaker than confluence. Newman's Lemma[1] (NL hereafter) states that local confluence implies confluence provided there are no infinite paths. We have automated the combinatorial core of Huet's proof[2] of NL using the resolution theorem prover Otter (`www-unix.mcs.anl.gov/AR/otter/`). We discuss the various

[1]On theories with a combinatorial definition of 'equivalence', *Annals of Mathematics* **43**(2), pp. 223–243, 1942.

[2]Confluent reductions: Abstract properties and applications to term rewriting systems, *Journal of the ACM* **27**(4), pp. 797–821, 1980.

sensitivities regarding the precise formulation in relation to the theorem proving procedure. Among the things we learned underway are:

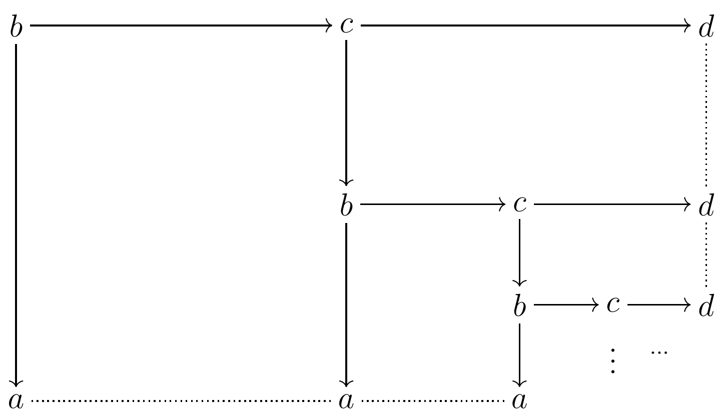
- The constructive formulation of 'no infinite paths' is good for automatic theorem proving.
- Transitive closure is first-order definable in the absence of infinite paths.

2 Local confluence is weaker than confluence

Consider the following directed graph attributed by Hindley to Kleene.

$$\begin{tikzcd} a & b \arrow[l] \arrow[r, bend left] & c \arrow[l, bend left] \arrow[r] & d \end{tikzcd}$$

Obviously, we have local confluence here since there are no diverging steps from a nor d, and diverging steps from b and c can be extended to paths ending in a and d, respectively, by using the cycle. However, the graph is not confluent, since the diverging paths $b \to c \to d$ and $b \to a$ cannot be extended to join. Attempts to deny this fact by applying a tiling analysis leads to the diagram below, where the cycle has been unwound into an infinite path. Here the dotted lines from a to a and from d to d represent paths of length zero for the sole purpose to keep things orthogonal.



3 Proving Newman's Lemma

NL provides an interesting test case for several reasons. First, it consists of a mix of first-order and higher-order aspects. The higher-order aspects are the transitive closure and the finiteness of the paths. This makes the identification of the first-order combinatorial core of the proof non-trivial. Second, the proof of NL is not completely trivial, as experienced by everybody seeing it for the first time. As such it provides a challenge to many theorem proving systems. In particular the automation of the constructive proof seems for the moment to be out of reach.

For a precise analysis we have to introduce some notions from rewriting theory.

Definition 3.1 Let $\to$ be a binary relation on a set S and let $\twoheadrightarrow$ be the reflexive–transitive closure of $\to$. In other words, we consider a directed graph $(S, \to)$ with path relation $\twoheadrightarrow$.

1. We say that $x \in S$ is *confluent* if, for all $x_1, x_2 \in S$, $x \twoheadrightarrow x_1$ and $x \twoheadrightarrow x_2$ implies that $x_1 \twoheadrightarrow y$ and $x_2 \twoheadrightarrow y$ for some $y \in S$. Such a y is called a *common reduct* of x_1 and x_2. We say that $\to$ is *confluent* if every $x \in S$ is confluent.

2. We say that $x \in S$ is *locally confluent* if, for all x_1, x_2, $x \to x_1$ and $x \to x_2$ implies that $x_1 \twoheadrightarrow y$ and $x_2 \twoheadrightarrow y$ for some $y \in S$. Note $x \to x_i$ here and $x \twoheadrightarrow x_i$ above. We say that $\to$ is *locally confluent* if every $x \in S$ is locally confluent.

3. We say that $\to$ is *terminating* if there is no infinite sequence $x_0 \to x_1 \to x_2 \to \cdots$ in S.

One of the classical proofs of NL is by contradiction. Let $\to$ be locally confluent and terminating. Assume there is an x which is not confluent, that is, there exist $x_1, x_2 \in S$ such that $x \twoheadrightarrow x_1$ and $x \twoheadrightarrow x_2$ and x_1 and x_2 have no common reduct. Since $\to$ is terminating, we may assume without loss of generality that x is an $\to$-maximal[3] non-confluent element. If not, there would be a non-confluent x' with $x \to x'$, and if that x' is not $\to$-maximal, then there would be a non-confluent x'' with $x' \to x''$ and so on, leading to a sequence contradicting the termination of $\to$. (This part is difficult to explain, it actually uses the Axiom of Dependent Choice, DC.) Hence we have that every $y \in S$ with $x \to y$ is confluent. From the fact that x_1 and x_2 have no common reduct, it follows that we do not have $x = x_1$ or $x = x_2$, so there must exist intermediate points i_1, i_2 such that $x \to i_1 \twoheadrightarrow x_1$ and $x \to i_2 \twoheadrightarrow x_2$. To x and these intermediate points we can apply local confluence to obtain a common reduct of the intermediate points. By the maximality of x we can then complete the diagram below. This is a contradiction and hence NL has been proved.

[3]If the transitive closure of $\to$ is viewed as a *greater than* ordering, then it would be natural to speak of $\to$*-minimal* instead.

$$\begin{tikzcd}
x \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{local confluence}"] & i_1 \arrow[r, two heads] \arrow[d, two heads] & x_1 \arrow[d, two heads] \\
i_2 \arrow[r, two heads] \arrow[d, two heads] & \cdot \arrow[r, two heads] & \cdot \arrow[d, two heads] \\
x_2 \arrow[rr, two heads] & & \cdot
\end{tikzcd}$$

The formalization of the above argument requires higher-order logic (to express transitive closure), DC and three-sorted first-order logic: one sort for the set S, one for the natural numbers and one for infinite sequences of elements of S. There exist many alternative proofs of NL, for example, based on the multiset extension of $\rightarrow$. All those proofs are a bit complicated by the fact that the absence of infinite paths has to be turned into an inductive argument, which again requires DC and many-sorted logic.

An important improvement, essentially due to Huet, is obtained by taking a constructive reformulation of NL as point of departure. In this formulation the infinite sequences such as used in the definition of termination and in DC are avoided by using an inductively defined predicate called accessibility[4].

Definition 3.2 Let $\rightarrow$ be a binary relation on a set S. The unary predicate $Acc_{\rightarrow}$ is inductively defined by the closure property $(*)$ if $Acc_{\rightarrow}(y)$ for all $y \in S$ such that $x \rightarrow y$, then $Acc_{\rightarrow}(x)$. By $Acc_{\rightarrow}(S)$ we express that $Acc_{\rightarrow}(x)$ for all $x \in S$.

In other words, all $\rightarrow$-maximal elements are accessible, as well as all elements whose successors are all $\rightarrow$-maximal, and so on. The inductive definition stipulates $Acc_{\rightarrow}$ to be the *smallest* unary predicate closed under $(*)$. For this to be well-defined, one easily checks that the intersection of an arbitrary set of unary predicates which are closed under $(*)$ is itself closed under $(*)$. Furthermore, any infinite sequence $x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \cdots$ consists of elements that are not accessible. The reason is that they can be left out without violating the closure property $(*)$. On the other hand, if there exists a non-accessible $x_0 \in S$, then there must be a non-accessible $x_1 \in S$ with $x_0 \rightarrow x_1$, and hence a non-accessible $x_2 \in S$ with $x_1 \rightarrow x_2$, and so on. Thus one proves using DC that $\rightarrow$ is terminating if and only if all elements of S are accessible, that is, if $Acc_{\rightarrow}(S)$.

[4] Can be traced back to Gentzen ('erreichbar'): Die Widerspruchsfreiheit der reinen Zahlentheorie. *Mathematische Annalen* **112**, pp. 493–565, 1936.

The point is now that we avoid the above equivalence proof by adopting $Acc_{\rightarrow}(S)$ straightaway as the 'positive' formulation of the absence of infinite paths. The advantages of using $Acc_{\rightarrow}(S)$ instead of the traditional formulation of termination are three-fold.

- DC is not needed anymore in the proof of NL.
- The sorts for the natural numbers and for infinite sequences are not needed.
- We can reason by induction on $Acc_{\rightarrow}(x)$, the induction step being first-order.

These reasons motivate the following reformulation of NL.

Lemma 3.3 *If $Acc_{\rightarrow}(S)$, then confluence of $\rightarrow$ follows from local confluence.*

We could have added a fourth advantage to the three advantages above, namely that the proof of NL in the formulation with the accessibility predicate can be done constructively, as shown in the next paragraph. The constructive proof is not more complicated than the classical one, it is actually shorter, but the relevant point here is that we do not know a theorem prover that is able to generate the constructive proof. Instead we had to use a resolution theorem prover, which first clausifies the negation of the statement and then searches for a resolution refutation of the resulting clauses.

We will sketch the constructive argument. By induction one proves that every accessible $x \in S$ is confluent. By $Acc_{\rightarrow}(S)$ we then obtain confluence. The induction step we have to prove is that confluence is preserved under the closure property $(*)$ in the inductive definition of $Acc_{\rightarrow}$. In other words, we have to prove that x is confluent if the induction hypothesis (IH) holds, that is, if every $y \in S$ such that $x \rightarrow y$ is confluent. Assume IH and let $x_1, x_2 \in S$ such that $x \twoheadrightarrow x_1$ and $x \twoheadrightarrow x_2$. If $x = x_1$ or $x = x_2$ then x_2 or x_1 is a common reduct of x_1, x_2. Otherwise, actually appealing to the inductive definition of the reflexive–transitive closure, there exist intermediate points as in the classical proof above. Now a common reduct can be obtained in exactly the same way as in the classical proof, with IH replacing the $\rightarrow$-maximality of x. This proves the induction step.

The above proof of the induction step is completely first-order and can be casted in the form of a refutation problem, namely the inconsistency of the conjunction below. Note that $=, \rightarrow, \twoheadrightarrow$ are uninterpreted binary relations here, with (local) confluence shorthand and $\rightarrow\cdot\twoheadrightarrow$ the composition of (first) $\rightarrow$ and (last) $\twoheadrightarrow$.

$$\bigwedge \begin{cases} 1. = \text{ is reflexive and symmetric} \\ 2. \twoheadrightarrow \text{ includes } = \text{ and } \rightarrow \text{ and is transitive} \\ 3. \twoheadrightarrow \text{ is included in the union of } = \text{ and } \rightarrow\cdot\twoheadrightarrow \\ 4. \rightarrow \text{ is locally confluent} \\ 5. \text{ there exists a non-confluent } x \text{ such that all } y \leftarrow x \text{ are confluent} \end{cases}$$

Otter found a refutation within some 15 minutes (on a Sun Ultra 10), with a combination of positive hyperresolution and unit-resulting resolution. The Otter input and output files match `~bezem/NL/NL1.*`. Some remarks on the conjuncts 1–3 and 5 above are in order here.

1. Transitivity of $=$ is not needed, which was obvious from the human proof.

2. Taking $\twoheadrightarrow$ to include $=$ is a convenient way of stating that $\twoheadrightarrow$ is reflexive. One can then conclude $a \twoheadrightarrow b$ from $a = b$ without any appeal to congruence axioms.

   Otter didn't use the fact that $\twoheadrightarrow$ includes $\rightarrow$. This came, admittedly, as a surprise to me, although it also follows from a close analysis of the human proof. It implies that NL can be strengthened to a (diamond) result for any reflexive–transitive relation satisfying conjunct 3. This strengthening is in itself not very interesting, but results in a surprising speed-up of the proof search by a factor 5.

3. It is well-known that transitive closure is not first-order definable. The conjunction of 2 and 3 is weaker than the reflexive–transitive closure. However, one can prove by induction on $Acc_{\rightarrow}(x)$ that conjunct 3 implies that $\twoheadrightarrow$ is a subset of the reflexive–transitive closure of $\rightarrow$. So in the presence of $Acc_{\rightarrow}(S)$ the conjunction of 2 and 3 exactly axiomatizes the reflexive–transitive closure. The observation that transitive closure is first-order definable for terminating relations may also be of use in other cases.

5. This is the negation of the induction step.

The refutation found by Otter is quite close to a human proof, see the diagram chase below. The arrows are labelled by numbers indicating the clause representing that arrow. A negative number means that the clause refutes the arrow. An ill-understood phenomenon is why it took so long after clause 477 to infer 24727 by transitivity from 45 and 70. The Otter input and output files match `~bezem/NL/NL2.*`. Otter can produce many other refutations of the same set of clauses, but they all seem to be inessential variations of the same diagram chase due to the many symmetries.

$$
\begin{tikzcd}
x \arrow[r, "27"] \arrow[d, "28"'] \arrow[dr, phantom, "\text{local confluence}"] & i_1 \arrow[r, two heads, "31"] \arrow[d, two heads, "45"'] \arrow[dd, bend left, two heads, "24727"] & x_1 \arrow[dd, two heads, "25185"] \\
i_2 \arrow[r, two heads, "48"] \arrow[d, two heads, "32"'] & \cdot \arrow[d, two heads, "70"'] & \\
x_2 \arrow[r, two heads, "477"] \arrow[rr, bend right, two heads, "-25536"'] & \cdot \arrow[r, two heads, "25291/-25719"] & \cdot
\end{tikzcd}
$$

An obvious difficulty for the proof search is the symmetry of the formulation of NL. For example, by applying local confluence, any step $x \to y$ gives rise to useless common reducts of y and y. It is well-known that it is possible to distinguish between 'horizontal' and 'vertical' steps in the formulation of both confluence and local confluence. This leads to an asymmetrical version of Newman's Lemma (aNL), which can be proved by the same proof with all the steps properly labelled as either 'horizontal' or 'vertical'. NL can easily be recovered from aNL by removing the distinction. The asymmetrical analogues of confluence and local confluence are known in the literature as commutativity and weak commutativity, respectively.

Definition 3.4 Let $\to_h$ and $\to_v$ be binary relations on a set S, with reflexive–transitive closures $\twoheadrightarrow_h$ and $\twoheadrightarrow_v$, respectively.

1. We say that x is *commutative* if, for all $x_1, x_2 \in S$, $x \twoheadrightarrow_h x_1$ and $x \twoheadrightarrow_v x_2$ implies that $x_1 \twoheadrightarrow_v y$ and $x_2 \twoheadrightarrow_h y$ for some $y \in S$. We say that $\to_h$ and $\to_v$ *commute* if every $x \in S$ is commutative.

2. We say that x is *weakly commutative* if, for all $x_1, x_2 \in S$, $x \to_h x_1$ and $x \to_v x_2$ implies that $x_1 \twoheadrightarrow_v y$ and $x_2 \twoheadrightarrow_h y$ for some $y \in S$. We say that $\to_h$ and $\to_v$ *commute weakly* if every $x \in S$ is weakly commutative.

The precise statement of aNL is that $\to_h$ and $\to_v$ commute if they commute weakly, provided $Acc_{\to_{hv}}(S)$. Here $\to_{hv}$ is the union of $\to_h$ and $\to_v$. A glance at the diagram below tells us that we need the induction hypothesis both for i_1 with $x \to_h i_1$ and for i_2 with $x \to_v i_2$.

$$
\begin{tikzcd}
x \arrow[r, "h"] \arrow[d, "v"'] \arrow[dr, phantom, "\text{weak commutativity}"] & i_1 \arrow[r, two heads, "h"] \arrow[d, two heads, "v"] & x_1 \arrow[d, two heads, "v"] \\
i_2 \arrow[r, two heads, "h"] \arrow[d, two heads, "v"'] & \cdot \arrow[r, two heads, "h"] & \cdot \arrow[d, two heads, "v"] \\
x_2 \arrow[rr, two heads, "h"] & & \cdot
\end{tikzcd}
$$

The proof of aNL follows the pattern of the proof of NL, but is based on the inconsistency of the following conjunction:

$$
\bigwedge \left\{ \begin{array}{l}
= \text{ is reflexive and symmetric} \\
\twoheadrightarrow_h \text{ and } \twoheadrightarrow_v \text{ include } = \text{ and are transitive} \\
\twoheadrightarrow_h \text{ is included in the union of } = \text{ and } \rightarrow_h \cdot \twoheadrightarrow_h \\
\twoheadrightarrow_v \text{ is included in the union of } = \text{ and } \rightarrow_v \cdot \twoheadrightarrow_v \\
\rightarrow_h \text{ and } \rightarrow_v \text{ are weakly commutative} \\
\text{there exists a non-commutative } x \text{ such that all } y \leftarrow_{hv} x \text{ are commutative}
\end{array} \right.
$$

Proof search in the asymmetrical case is two orders of magnitude faster than in the symmetrical case: less than a second. The relevant Otter files match `~bezem/NL/NLa.*`.

Automated theorem proving is one of the many fascinating subjects in between logic and computer science. There is a lot to learn and to improve in this kind of experiments with theorem provers. And last but not least: it's fun!

Acknowledgement

The author is indebted to Wilfried Buchholz, Dimitri Hendriks, Gérard Huet and Tobias Nipkow for valuable comments.