cs.cornell.edu

Calculational logic

5–7 minutes

Calculational propositional logic is a product of researchers in the field of the formal development of algorithms. (Click here for a short history.) C is sound and complete. The emphasis in proofs is on substitution of equals for equals, instead of modus ponens. Equality, or equivalence, assumes an important role instead of being a bit player, as in most propositional logics.

We use & for conjunction, | for disjunction, and ~ for negation (not).

In our notation, b = c denotes equality, for b and c of the same type, while b == c, or equivalence, is defined only for b and c of type boolean. For b and c of type boolean, b = c and b == c have the same meaning.

One reason for having two symbols for equality over the booleans is that equality = is usually viewed as being conjunctional:

b = c = d   is an abbreviation for   b = c & c = d

while b == c can be used associatively, which is of great importance for us:

b == c == d   is equal to b == (c == d)   and to   (b == c) == d .

Below is a typical proof in calculational logic. The line numbers are present only for later discussion. This example is chosen because it is simple but shows all aspects of the formal proof system (as discussed later). The theorem numbers refer to theorems of A Logical Approach to Discrete Math. Associativity and symmetry of operators are treated transparently, without mention; this results in a large reduction in the number of theorems to be presented and simplifies manipulation. Finally, note that the style is just a formalization of the style of calculation used in high school algebra, and indeed by many scientists in their work --that is why the style is so useful.

Here is a proof of ~p == p == false.

(0) ~p == p == false
(1) = < (3.9), ~(p == q) == ~p == q, with q:= p >
(2) ~(p == p) == false
(3) = < Identity of == (3.9), with q:= p >
(4) ~true == false --(3.8)

Inference rules of calculational logic

Here are the four inference rules of logic **C**. (P[x:= E] denotes textual substitution of expression E for variable x in expression P):

- Substitution: If P is a theorem, then so is P[x:= E].

```
|- P    --->  |- P[x:= E]
```

- Leibniz: If P = Q is a theorem, then so is E[x:= P] = E[x:= Q].

```
|- P = Q    --->  |- E[x:= P] = E[x:= Q]
```

- Transitivity: If P = Q and Q = R are theorems, then so is P = R.

```
|- P = Q,  |- Q = R   --->  |- P = R
```

- Equanimity: If P and P == Q are theorems, then so is Q.

  ```
  |- P, |- P == Q   --->  |- Q
  ```

Explanation of proof format

We explain how the four inference rules are used in proofs, using the proof of `~p == p == false`.

```
    (0)     ~p == p == false


    (1)  =     < (3.9), ~(p == q) == ~p == q, with
q:= p >


    (2)     ~(p == p) == false


    (3)  =     < Identity of == (3.9), with q:= p >


    (4)     ~true == false     --(3.8)
```

First, lines (0)-(2) show a use of inference rule Leibniz:

```
(0) = (2)
```

is the conclusion of Leibniz, and its premise ~(p == p) == ~p == p is given on line (1). In the same way, the equality on lines (2)-(4) are substantiated using Leibniz.

The "hint" on line (1) is supposed to give a premise of Leibniz, showing what substitution of equals for equals is being used. This premise is theorem (3.9) with the substitution p:= q, i.e.

```
(~(p == q) == ~p == q)[p:=q]
```

This shows how inference rule Substitution is used within hints.

From (0) = (2) and (2) = (4), we conclude by inference rule Transitivity that (0) = (4). This shows how Transitivity is used.

Finally, note that line (4), ~true == false, is a theorem, as indicated by the hint to its right. Hence, by inference rule Equanimity, we conclude that line (0) is also a theorem. And (0) is what we wanted to prove.

This proof format has several advantages.

- The use of each inference rule is determined by the proof format, so the names of the inference rules need not be mentioned. This reduces the amount of reading and writing in a proof.
- The proof is completely rigorous, but without complexity overwhelming. Further, the proof is simple enough for students to comprehend. In fact, it formalizes the style of calculation taught in high school.
- In general, there is little copying of formulas from one line to another --one of the disadvantages of Hilbert-style proofs.
- With this format, useful proof principles and strategies can be taught.
- The proof can be read forward or backward.