Supplementary Notes for the MAC-lecture September 18

September 18, 2003

Classical axioms, Mac axioms and Mac rules

In classical mathematics we often take some things for **granted** (e.g. that 117 is a natural number), whereas we want **proofs** for statements which seem 'less obvious'. This has made some mathematicians interested in finding a minimal set of **basic assumptions (called axioms)** from which all known classical mathematical statements can be proved.
As an example, we may consider two of the axioms of the Italian mathematician Peano (1858 – 1932):

1) $1 \in \mathbf{N}$

2) $x \in \mathbf{N} \vdash x + 1 \in \mathbf{N}$ (i.e. if x is a natural number then so is x + 1)

From these axioms we may now **prove** that 117 as well as 2, 3, 4,.. etc. are natural numbers, **provided the two assumptions/axioms are true**!
Hence, in *classical* mathematics axioms are basic assumptions which *cannot* be proved!

In the *Mac System* **axioms are equations that hold unconditionally**, e.g.

$0 \in \mathbf{N} \equiv \mathrm{T}$

(Note that 0 is a natural number in the Mac System!)
Rules stating that equations hold **under certain conditions** are called **inference rules**, and one of the Mac inference rules is:

$x \in \mathbf{N} \vdash (x^{+} \in \mathbf{N} \equiv \mathrm{T})$

A **Mac rule** is either an axiom or an inference rule stated **without proof**. Hence, a Mac rule is a *basic assumption* in the Mac System.

Basic rules are most often introduced to Map by means of the directive

$[\ x\ \textbf{rule}\ y : z\ ]^{\cdot}$

where $[x]^{\cdot}$ is the mathematical system, and $[y]^{\cdot}$ is a *macro* for the rule $[z]^{\cdot}$.
For example (cf. page 547):

[**Mac rule PeanoA**: $0 \in \mathbf{N} \equiv \mathrm{T}$]

[**Mac rule PeanoB**: $x \in \mathbf{N} \vdash x^{+} \in \mathbf{N} \equiv \mathrm{T}$]

The **type tables** in Volume 3 are compact versions of basic inference rules, and thus (row 2, coloumn 3) in the table on page 607 should be read as:

[**Mac rule TypeB+D**: $x \in \mathbf{B} \vdash y \in \mathbf{D} \vdash x + y \in \mathbf{X} \equiv \mathrm{T}$]

If we extend the Mac System by **defining new operators**, e.g.

$\mathrm{fplustwo}(x) \doteq x + 2$

we automatically get a new basic Mac rule, viz. the axiom:

$\mathrm{fplustwo}(x) \equiv x + 2$

If we have **an equation comparing two terms without variables**, e.g. $[2 + 2 \equiv 4]$, the computer on which Map runs has no difficulties in establishing the validity/failure of the equation, and thus we also have basic computation axioms of the form:

$2 + 2 \equiv 4$

$2 + 2 = 5 \equiv \mathrm{F}$

$(1 :: 2 :: <>)\ \mathrm{tail} \equiv 2 :: <>$

Using the **basic** Mac rules described above, we may **prove** the *validity* of **new rules** (called **lemmas**), whereas the *failure* of any new rule (called antilemmas) requires knowledge of the Mac antirule, and will be postponed until Chapter 16.
By an algebraic proof one may show the validity of new **axioms**, whereas new inference rules are proved by techniques introduced in Chapters 11 and 12.

Algebraic Proofs

Since axioms are equations that hold unconditionally, we may start by considering the following equations in the Mac System:

a) x + 1 - 1 ≡ x + 0

b) x + 0 ≡ x

c) x - x ≡ 2

d) < 1 - 1, y, z > head tail ≡ 0

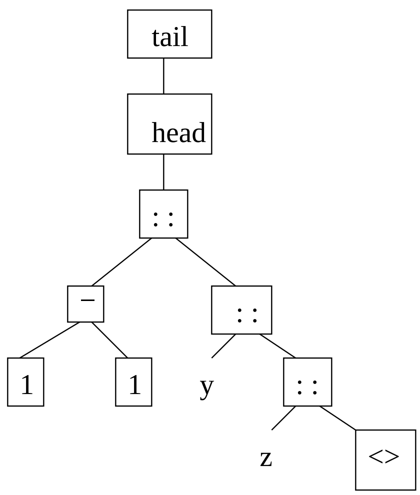
Although these equations seem rather 'obvious', and everybody can see that equation **c)** does not hold, many people will probably not see that equations **a)** and **b)** does not hold (unconditionally) *either*! (take e.g. x ≡ Ø.1F and x ≡ T, respectively, as counter-examples). If they try to make an algebraic proof for the *validity* of **a)** or **b)**, they will, however, never succeed.

How can **d)** be proved to be an axiom/lemma?

Well, in an algebraic proof one shows that

the syntax tree of the left hand side can be transformed to the syntax tree of the right hand side, using basic rules or lemmas!

Hence, we consider the syntax tree of [< 1 - 1, y, z > head tail]˙:



The Map program always looks at terms in the form of syntax trees, and thus Map has no difficulties in accepting our claim that the rule

[**Mac rule HeadPair**: (x :: y) head ≡ x]

can be applied to transform the term [< 1 - 1, y, z > head tail]˙ to [(1 - 1) tail]˙. This claim may be written in several ways in our proof:

Replace ▹ HeadPair ▹ (1 - 1) tail

or

Replace ▹ (x :: y) head ≡ x ▹ (1 - 1) tail

or

Replace ▹ HeadPair ▹ (((1 - 1) tail))

since Map performs term and macro reductions before checking the validity of our proof lines.

Further reduction of the syntax tree is easy, since we now have a term *without variables*!
Hence, we may use the computation axiom [(1 - 1) tail ≡ 0]:

Replace ▷ (1 - 1) tail ≡ 0 ▷ 0

Having obtained the syntax tree of the right hand side (one node containing 0), the proof is finished, and we may inform Map about our new rule/lemma:

[**Mac lemma No.1**: < 1 - 1, y, z > head tail ≡ 0]˙

and ask Map to check our algebraic proof in the Mac System:

[**Mac proof of No.1**:

Algebra	▷			< 1 - 1, y, z > head tail	;
Replace	▷	HeadPair	▷	(1 - 1) tail	;
Replace	▷	(1 - 1) tail ≡ 0	▷	0	]

If the reader of your proof is human (e.g. your teacher), he/she may not see the syntax trees when reading the terms in your proof, and thus you may help a human proof reader by stating explicitly that [< 1 - 1, y, z > head tail]˙ and [(1 - 1 :: < y, z >) head tail]˙ have the same syntax tree, i.e. are equal:

[**Mac proof of No.1**:

Algebra	▷			< 1 - 1, y, z > head tail	;
Reflexivity	▷			(1 - 1 :: < y, z >) head tail	;
Replace	▷	HeadPair	▷	(1 - 1) tail	;
Replace	▷	(1 - 1) tail ≡ 0	▷	0	]

Reflexivity is a macro for the axiom [x ≡ x]:

[**Mac rule Reflexivity**: x ≡ x]

If we want to use a rule of the 'axiom-form' $\mathcal{A} \equiv \mathcal{B}$, we must of course tell Map whether an occurrence/instance of $\mathcal{A}$ is to be substituted by the corresponding version of $\mathcal{B}$ **or** an occurrence/instance of $\mathcal{B}$ is to be substituted by the corresponding version of $\mathcal{A}$. In the first case the macro '**Replace**' is used (cf. the proof above), whereas the macro '**Reverse**' is used in the latter case. Since it does not make any difference, when the rule is **Reflexivity**, this rule is used without Replace/Reverse. Likewise, **Definition axioms** are used without Replace/Reverse, although they would have made sense here.
Note that replacements/reverse replacements follow the rules of **correct substitution of equals** described in a previous supplementary lecture note. So please remember any necessary parenthesis!

Let us extend the Mac System by defining a new operator:

fplustwo(x) ≐ x + 2

How should we prove that the following is a lemma?:

[**Mac lemma No.2**: 2 ≡ fplustwo(< 1 - 1, y, z > head tail)]˙

Well, we would probably like to reduce the term on the right hand side using lemma No.1, but in an algebraic proof one works on the left hand side!!
The trick is to (more or less) write the proof from below and up, and then swap 'Reverse's and 'Replace's.

[**Mac proof of No.2**:

Algebra	▷			2	;
Reverse	▷	0 + 2 ≡ 2	▷	0 + 2	;
Definition	▷			fplustwo(0)	;
Reverse	▷	No.1	▷	fplustwo(< 1 - 1, y, z > head tail)	]

Try to compare this proof with an algebraic proof of

[**Mac lemma No.3**: fplustwo(< 1 - 1, y, z > head tail) ≡ 2]˙