1 Plan

1. Review the issues involved in expressing arithmetic in FOL, eg. treating equality, role of explicit recursive function definitions, e.g. add(x, y), mult(x, y), obtaining zero and successor without constants in the base language.
2. Arithmetic as a general example of specification. Projects could include specifying lists, finite sets, finite bags, graphs, trees, etc.

2 Axioms for Arithmetic from textbooks

Kleene's language explicitly introduces 0 and successor $s(x)$, which he denotes as a' for $s(a)$.

82 A FORMAL SYSTEM CH. IV

GROUP A. Postulates for the predicate calculus.

GROUP A1. Postulates for the propositional calculus.

1a. $A \supset (B \supset A)$.

1b. $(A \supset B) \supset ((A \supset (B \supset C)) \supset (A \supset C))$.

2. $\dfrac{A,\ A \supset B}{B.}$

3. $A \supset (B \supset A \,\&\, B)$.

4a. $A \,\&\, B \supset A$.

4b. $A \,\&\, B \supset B$.

5a. $A \supset A \vee B$.

5b. $B \supset A \vee B$.

6. $(A \supset C) \supset ((B \supset C) \supset (A \vee B \supset C))$.

7. $(A \supset B) \supset ((A \supset \neg B) \supset \neg A)$.

8°. $\neg\neg A \supset A$. [Boolean Logic]

GROUP A2. (Additional) Postulates for the predicate calculus.

9. $\dfrac{C \supset A(x)}{C \supset \forall x A(x).}$

10. $\forall x A(x) \supset A(t)$.

11. $A(t) \supset \exists x A(x)$.

12. $\dfrac{A(x) \supset C}{\exists x A(x) \supset C.}$

GROUP B. (Additional) Postulates for number theory.

13. $A(0) \,\&\, \forall x(A(x) \supset A(x')) \supset A(x)$.

14. $a'=b' \supset a=b$.

15. $\neg a'=0$.

16. $a=b \supset (a=c \supset b=c)$.

17. $a=b \supset a'=b'$.

18. $a+0=a$.

19. $a+b'=(a+b)'$.

20. $a \cdot 0=0$.

21. $a \cdot b'=a \cdot b+a$.

Figure 1: Kleene *Introduction to Metamathematics* 1952

We can write the combination of his 14 and 17 as

$$a = b \Longleftrightarrow a' = b'$$

We express this as

$$\forall x_1, x_2, y_1, y_2.(Succ(x_1, y_1)\ \&\ Succ(x_2, y_2)) \Rightarrow (Eq(x_1, x_2) \Longleftrightarrow Eq(y_1, y_2))$$

Consider his Axiom 15 $\neg(a' = 0)$. We can express it as

$$\forall x, y.(Succ(x, y) \Rightarrow \neg Eq(y, 0)).$$

Notice that we can define False as $Eq(0, s(0))$, i.e. as $0 = 1$. We then have results like

$$\exists x Eq(x, s(x)) \Rightarrow Eq(0, s(0)) \text{ which is}$$

$$\exists x Eq(x, s(x)) \Rightarrow False \text{ which is}$$

$$\neg \exists x Eq(x, s(x))$$

We can also prove

$$\forall x(Eq(x, s(x)) \Rightarrow Eq(0, s(0)).$$

3 Specifying Addition with Axioms

Recall this definition from the Oct 6 lecture.

$$add(0, y) = y$$

$$add(s(x), y) = s(add(x, y))$$

We can symbolize the atomic relation $add(x, y) = z$ as $Add(x, y, z)$. We need to express the equations for the primitive recursive definition without the constants. Here is one way to do it.

$$Z(x) \Rightarrow A(x, y, y)$$

$$(A(x, y, z)\ \&\ Succ(x, x')) \Rightarrow A(x', y, z')\ \&\ Succ(z, z')$$

A good exercise is to write a similar equation for defining Mult(x, y, z) using Add(x, y, z). Our definition for $A(x, y, z)$ could be the one we want for Add(x, y, z). What do you think?

Can we prove these properties?

$$\forall x, y.\exists z.Add(x, y, z)$$

$$\forall x, x', y, y', z, z'.(Eq(x, x')\ \&\ Eq(y, y')\ \&\ Add(x, y, z)\ \&\ Add(x', y', z')) \Rightarrow Eq(z, z')$$

Here we left z, z' as free variables. We could also have used the quantifier prefix $\forall x, x', y, y', z, z'$.