Bootcamp

Christoph Thiele

Summer 2012

0.1 An example of a primitive universe

A primitive universe consists of primitive objects and primitive sets. This allows to form primitive statements as to which objects are contained in which sets. A primitive statement takes exactly on truth value, it is either true or false. For our purpose of discussion it will suffice to work with one very concrete set of primitive objects and primitive sets.

The primitive objects are any one of a finite list of specifically declared things. To agree on which things one declares primitive objects, one may simply list these things. Each object in the list has its own identity, no two objects in the list are equal. For later purpose, the order in which these primitive obects are listed, is important.

We shall entirely work with one example and adopt all lower case letters of the alphabet as primitive objects, all written in a specific fond.

$$\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f}, \mathbf{g}, \mathbf{h}, \mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{l}, \mathbf{m}, \mathbf{n}, \mathbf{o}, \mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s}, \mathbf{t}, \mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}$$

We have listed them here in alphabetical order, separated by commata. It is not exactly important which list of objects we choose here. It could be any sets of objects in real life say all apples in a basket of apples. However, in such a situation we would have to agree on a coding, since we cannot insert these apples into a text. We would have to choose symbols or strings of symbols in the text to represent these objects. Even in the current list one could view the typesetted **a** as a symbol for first letter in the alphabet, an object that has an identity independent of a specific choice of font for example.

One impotant feature of our coding scheme is that the symbol code we use for the object does not include symbols that we have to reserve for our later discussion, such as $\{, \}, \in, \notin, (,), \wedge, \vee, \bigwedge, \bigvee$ as well as - to be safe - all capital letters of the alphabet in normal fond, plus a few more symbols that we may want to introduce.

A primitive set collects primitive objects together, for example the set

$$\{\mathbf{aeiou}\}$$

collects the five vowels together. In fact, this primitive set may be a code for the concept of a vowel, though ofcourse this notion of vowel does not as of yet describe well any intricacies of sounds or other interesting linguistic aspects of vowels. However, just putting objects together into sets is a very powerful start of making a concept.

We choose to list the objects without commata, this is because with our present list of primitive objects it is not difficult to determine when the symbol for one object stops and when the next symbol starts. If we had chosen different objects, for example words written in the letters of the alphabet, we would have to be more careful.

To have a formal definition of a primitive set, we say a primitive set is a string of symbols as follows: We start with an opening "set bracket" $\{$, then we list primitive objects in alphabetical order, and not repeating the same primitive object, i.e. each object appears at most once. We allow the empty list which does not list any primitive object and looks like $\{\}$.

Note that there are only finitely many primitive sets one can form this way. To be precise one has be 2^{26} such primitive sets, since when forming the list of a primitive set one may for each of the primitive elements decide independently whether the primitive element should be in a primitive set or not.

Given a primitive object and a primitive set, one may want to discuss the question whether the primitive object appears in the list of the primitive set or not. To discuss such issues we introduce primitive statements, which are strings consisting of an object first, then the symbol $\in$ or $\notin$ and then the set. For example

$$\mathbf{e} \in \{\mathbf{aeiou}\}$$

$$\mathbf{x} \in \{\mathbf{aeiou}\}$$

$$\mathbf{e} \notin \{\mathbf{aeiou}\}$$

$$\mathbf{x} \notin \{\mathbf{aeiou}\}$$

are four primitive statements. If the primitive object actually does appear in the list of the primitive set and the middle symbol is $\in$, we say that the primitive statement is true. If the primitive object does not appear in the list of the primitive set and the symbol is $\in$, we say that the primitive statement is false. If the primitive object does appear in the list of the primitive set and the symbol is $\notin$, we say that the primitive statement is false. If the primitive object does not appear in the list of the primitive set and the symbol is $\notin$, we say that the primitive statement is true. Thus each primitive statement is either true or false.

Note that there are $26 \times 2 \times 2^{26}$ primitive statements. namely there are 26 choices of choosing the primitive object, there are 2 choices of choosing the symbol, and there are 2^{26} choices of choosing the primitive set. Thus we have assembled some finite collection of primitive statements.

To facilitate the discussion of the situation, we adopt the use of variables. A variable for a primitive object is a symbol, often a letter, that will stand for any object we may choose at a later point. Similarly a variable for a primitive set is a symbol that will stand for any primitive object we may choose. We then have the following definition of a primitive statement:

Definition 1 (Primitive statements) *A primitive statement is of the form $A \in B$ where A is a primitive object and B is a primitive set.*

Lemma 1 (Truth value of primitive statement) *Every primitive statement is true or false, not both at the same time.*

Proof: By inspection of the definitions.□

We may also use variables for primitive statements, and resreve the latters P, Q, R, S for that. If P is a primitive statement, then it is of the form $A \in B$ for some primitive object A and some primitive set B or it is of the form $A \notin B$ for some primitive object A and some primitive set B.

Remark:

To talk about the operation of switching the symbol of a primitive statement to the other symbol, thus producing another one of the primitive statements, we introduce the negation operation.

Definition 2 (Negation of primitive statements) *If P is a primitive statement, we write $\overline{P}$ for*

1. *the statement $A \notin B$ if P is of the form $A \in B$,*
2. *the statement $A \in B$ if P is of the form $A \notin B$,*

Note that we have used the common practice that specific symbol for a variable stands for the same primitive object of primitive set throughout the duration of a particular discussion, definition, or argument.

0.2 Propositional logic for $\wedge$ and $\vee$

Propositional logic is a method of producing new "propositional" statements from a collection of primitive statements by means of certain logical operations. We shall use the operations $\wedge$ and $\vee$ as basis for our discussion, this is sufficient to later express all commonly used propositional operations by means of these.

Propositional statements are produced recursively, that means a propositional statement is formed from other propositional statements that were formed previously. there is no limit as to how often one may apply these procedures, which will lead to an infinitude of propositional statements. This infinitude of propositional statements is a new phenomenon compared ot the previously discussed primitive statements.

Generally, a propositional statement is a string formed from the symbols $\in, \notin, \wedge, \vee$ and the opening bracket (and closing bracket) and primitive objects and primitive sets (which themselves are strings using certain symbols discussed before.

Definition 3 (Propositional statements)

1. *Every primitive statement is a propositional statement.*

2. *If P and Q are propositional statements, then $(P) \wedge (Q)$ is a propositional statement.*

3. *If P and Q are propositional statements, then $(P) \vee (Q)$ is a propositional statement.*

For example,

$$((\mathbf{a} \in \{\mathbf{ab}\}) \wedge (\mathbf{b} \in \{\mathbf{bc}\})) \vee (\mathbf{a} \in \{\mathbf{bc}\})$$

is a propositional statement.

Note that given a propositional statement, one can easily determine the number of steps of type 2 or 3 of the recurision used to arrive at the propositional statement. Namely, this is simply the number of symbols $\wedge$ and $\vee$ used in the propositional statement. In particular, it makes sense to call a propositional statement primitive only if that number is zero, and to call it non-primitive if this number is not zero. We are guaranteed that this is well defined since we cannot arrive again at a primitive string after a nonzero number of recursions.

One can also determine the last step used in the recursion to form a certain propositional statement. Note that each propositional statement contains as many opening brackets as closing brackets. Namely, this property is preserved throughout the recursion. Define the bracket balance of a symbol ina string the difference between the number of opening brackets and the number of closing brackets in the initial part of the string up to including the symbol in question. Note that this balance is always a nonnegative integer. If a the propositional statement is not primitive, then this balance is equal to zero for exactly three symbols: the last bracket before the last symbol $\vee$ or $\wedge$ added, that symbol itself, and the very last bracket in the whole string. This shows that one can determine the most recent step in the recursion using the bracket balance.

Just as for primitive statements, one would like to have a truth value for each propositional statement. To make sure each propositional statement has a truth value, we revisit the recursive process. Note that by the previous remarks about uniqueness of the previous step in the recursion, all these definitions are well defined.

Definition 4 (Propositional statements) 1. *If a propositional statement is primitive, then the truth value of the propositional statement is that of the primitive statement.*

2. *If a propositional statement is of the form $(P) \wedge (Q)$ with propositional statements P and Q, then it is true if both P and Q are true and it is false if at least one of P and Q is false.*

3. *If a propositional statement is of the form $(P) \vee (Q)$ with propositional statements P and Q, then it is true if at least one of P and Q is true, and it is false if both P and Q are false.*

It is easy to verify inductively:

Exercise 1 *Each propositional statement is true or false, but not both true and false.*

It turns out that truth value of a propositional statement can be deduced from other propositional statements via certain rules that are different from the pure recursional definition. Some rules say that under certain circumstances two propositional statements have the same truth value. Other say that a value of one propositional statement determines the value of another propositional statement, but not the other way around. It will be an important observation that the verification of these rules will be based puerely on the recursive rules, and have nothing to do with the structure of underlying primitive statements.

Definition 5 *For any propositional statementas P and Q and R we have the following:*

1. *Trivial $\wedge$: $(P) \wedge (P)$ is tautological to P*
2. *Trivial $\vee$: $(P) \vee (P)$ is tautological to $(Q \vee P)$*
3. *Commutativity of $\wedge$: $(P) \wedge (Q)$ is tautological to $(Q \wedge P)$*
4. *Commutativity of $\vee$: $(P) \vee (Q)$ is tautological to $(Q \vee P)$*
5. *Associativity of $\wedge$: $((P) \wedge (Q)) \wedge (R)$ is tautological to $(P) \wedge ((Q) \wedge (R))$*
6. *Associativity of $\vee$: $((P) \vee (Q)) \vee (R)$ is tautological to $((P) \vee (Q)) \vee (R)$*
7. *Distributive law for $\wedge(\vee)$: $(P) \wedge ((Q) \vee (R))$ is tautological to $((P) \wedge (Q)) \vee ((P) \wedge (R))$*
8. *Distributive law for $\vee(\wedge)$: $(P) \vee ((Q) \wedge (R))$ is tautological to $((P) \vee (Q)) \wedge ((P) \vee (R))$*

More generally, one may define two propositional statements of there is a sequecne of steps, either using one of the tautology ruels above or recursion steps that produce the tautological propositional statements. We omit details here.

We now turn to rules how to infer from truth value of one or more propositional statements the truth value of another propositional statement, without in general being able to infer the opposite direction. There are two types of such rules, we may infer that some propositional statement is true, or we may infer that a propositional statement is false.

1. If a propositional statement if the form $(P) \wedge (Q)$ is true, then also P is true.
2. If a propositional statement if the form $(P) \vee (Q)$ is true and the statement P is false, then Q is true.
3. If a propositional statement if the form $(P) \wedge (Q)$ is false and P is true, then Q is false.

4. If a propositional statement if the form $(P) \vee (Q)$ is false, then also P is false.

Remark:
The symmetry of these can be understood by means of negation:

Definition 6 *If P is a propositional statement, we define $\overline{P}$ as follows:*

1. *If P is a primitive statement, then $\overline{P}$ has been defined before.*

2. *If P is of the form $(Q) \wedge (R)$ for some propositional statements Q and R, then we defined $\overline{P}$ to be $(\overline{Q}) \vee (\overline{R})$.*

3. *If P is of the form $(Q) \vee (R)$ for some propositional statements Q and R, then we defined $\overline{P}$ to be $(\overline{Q}) \wedge (\overline{R})$.*

Again, that these definitions are well defined, we need the observations made before that given a propositional statement one can uniquely determine whether it is primitive or not, and in the latter case what was the last step in the recursion to produce the propositional statement.

We have the followng more concise rule to negate a propositional statement

Exercise 2 *A propositional statement is negated by replacing all symbols $\in$ by $\notin$ and vice versa and replacing all symbols $\wedge$ by $\vee$ and vice versa.*

Remark:
The rules of commutativity and associativity make it appropriate to talk about n-fold connection of polynomials by the connectives such as

$$P \wedge Q \wedge R$$

$$P \vee Q \vee R$$

For $n = 1$ such product is the statement P itself, for $n = 0$ one introduces to more stetements, the statement $\overset{0}{\wedge}$ being always truue and the statement and $\overset{0}{\vee}$ being always false.

We have the following rules for these symbols:

Exercise 3 *For any statement P,*

1. *$(\overset{0}{\wedge} \wedge P)$ is tautological to P*

2. *$(\overset{0}{\wedge} \vee P)$ is tautological to $\overset{0}{\wedge}$*

3. *$(\overset{0}{\vee} \wedge P)$ is tautological to $\overset{0}{\vee}$*

4. *$(\overset{0}{\vee} \vee P)$ is tautological to P*

0.3 Predicate logic

Picking the primitive object a for the moment, the propositional statement

$$(\mathbf{a} \in B) \vee (\mathbf{a} \notin B)$$

holds for every choice of primitive set B, as follows from one of the inference rules. To express this fact in propositional logic, we would need to form these statements for every single primitive set in place of B, in our case 2^{26} of them, and combine all these statements with the symbol $\wedge$ to one single statement.

Predicate logic is an extended form of logic where one can formulate such statement in more concise form, having later the additional benefit that in principle on can form a statement that express a combination for an infinite choices of sets. However, only such combinations of infinite statements will be allowed where all the statement slook alike in some fashion to be made precise. This at present is hypothetical though, since we only do have finitely many primitive sets, albeit a very large number.

Predicate statement will use in addition to the symbols known from propositional logic as well the symbols $\bigwedge$ and $\bigvee$, which are read "for all" and "there exists". In our finite primitive universe, they are informally ways of combining all choices of some variable with $\wedge$ or $\vee$. These symbols are called quantifyers.

In principle we could alse quantify over the primitive elements. However, this would lead to a cumbersome case distinction, and we will not have need to do so.

We shall need variables, and we will agree that the variables will be written

$$B_0, B_1, B_2, B_3, \ldots$$

We are already used to having infinitely many propositional statements. Since predicate logic will be defined using a similar recursion, we will need an infinite supply of variables, whence the above list. The full strength of this infinitude of variables will however only play out once we start talking about infinite universes.

Our rules will hold for every choice of variable, so to address the infinitude of variables by names, we will introduce variables X, Y, Z for these variables.

An expression of the form

$$\mathbf{a} \in B_0$$

does not not have a definite value of truth, but the truth will depend on which set we pick for B_1 This expression will therefore not be called a statement. There are however statements

$$\bigwedge B_0(\mathbf{a} \in B_0)$$

$$\bigvee B_0(\mathbf{a} \in B_0)$$

The first stating that for all sets B_0 the statement $\mathbf{a} \in B_0$ is true, clearly a false statement since $\mathbf{a} \in \{\mathbf{bc}\}$ is not true. The second is stating that there exists a set that we can insert for the variable B_0, a true statement since $\mathbf{a} \in \{\mathbf{ab}\}$.

Expressions will contain variables, and variable will be called free or bound. Informally, a variable X is called free in an expression if the expression does not have a substring of the form $\bigwedge X$ or $\bigvee X$, and it is called bound otherwise. Formally, we will define containment of variables as well as freeness and boundedness as part of the recursion.

The following defines expressions.

1. For every primitive object A and primitive set B, $A \in B$ is an expression. It contains no variables, neither free nor bound.

2. For every primitive object A and variable X, $A \in X$ is an expression. It contains only the variable X and this variable is free.

3. If P and Q are expressions such that whenever one variable is bound in one of the expressions, then it is not contained in the other expression, then $(P) \wedge (Q)$ is an expression. This expressions contains all variables that are cpontained in P or Q, and each such variables is bound if it is bound in one of P or Q and it is free otherwise.

4. If P and Q are expressions such that whenever one variable is bound in one of the expressions, then it is not contained in the other expression, then $(P) \vee (Q)$ is an expression. This expressions contains all variables that are cpontained in P or Q, and each such variables is bound if it is bound in one of P or Q and it is free otherwise.

5. If P is an expression and X is a variable that may or may not occur in P, but if it does occur, then it is free in P, then

   $$\bigwedge X(P)$$

   is an expression.

   This expression contains the variable X as well as any variable that is contained in Q. The variable X is bound, while all other variables contained in P are bound if they are bound in Q, and they are free otherwise.

6. If P is an expression and X is a variable that may or may not occur in P, but if it does occur, then it is free in P, then

   $$\bigvee X(P)$$

   is an expression.

   This expression contains the variable X as well as any variable that is contained in Q. The variable X is bound, while all other variables contained in P are bound if they are bound in Q, and they are free otherwise.

Exercise 4 *Prove that on can determine the number of reecursion steps that were used to define a certain expression. Prove that one can determine the last step used to form the expression.*

Exercise 5 *If X is a variable and P an expression, then each of the strings $\bigwedge X$ and $\bigvee X$ appears at most once in the expression, and they cannot both appear in the expression.*

Definition 7 *An expression is called a statement, if every variable that is contained in the expression is bound in the expression.*

Note that in particular the primitive statements are statements, while expressions of the form $A \in X$ for some primitive object A and some variable X is not a statement.

Lemma 2

1. *If P is a statement if the form $(Q) \wedge (R)$ or the form $(Q) \vee (R)$ with expressions Q and R, then Q and R are statements as well.*

2. *If P is a statement that contains an expression of the form $\bigwedge X(Q)$ or of the form $\bigvee X(Q)$ for some expression Q, and if B is a particular primitive set, then if we replace any occurence of X in Q with the primitive set B, then the thus formed string Q' is a statement.*

Proof: We consider the first claim. If Y is some variable in Q, then it is also a variable in P and hence it is bound. One string of type $\bigwedge Y$ or $\bigvee Y$ appears in the expression P, and since by the recursive definition Y does not appear in R, it appears in Q. Hence Y is boudn in Q. Since Y was aribitrary, this shows that Q is a statement. Similarly one proves that P is a statement. We consider the second claim. That Q is an expression follows by the recursive rules and induction on numebr of steps needed to produce Q. In essence whenever $A \in X$ is a basic input into the recursion, it is replaced by $A \in B$. To see that it is a statement, let Y be a variable contained in Q'. It is not equal X since we have replaced all occurrences of X by B. Then Y is in P and since P is a statement, there occurs a string $\bigwedge Y$ or $\bigvee Y$ inside P. Since Y is not X, this string has to occur inside Q. Since this string does not contain any X, it was not changed in the transition from Q to Q', so it is still contained in Q'. Since Y was arbitrary, this proves that Q' is a statement. $\square$

Exercise 6 *Profde that if Q is a sustring of P that is a statmeent, then Q was used in the recursive process to produce P.*

Exercise 7 *Formulate and prove a lemma as above where Q is a statement that is a substring of P, and Q is of the form $\bigwedge X(R)$ or $\bigvee X(R)$ and all ofccurerences of X in R are replaced by a variable bound in P (specify which variables are allowed) but not bound in Q.*

We now define recursively truth values for statements.The recursion is by the number of variables appearing in a statement.

1. If P is of the form $(Q)\wedge(R)$ for some expressions Q and R, then these expressions are statements. Then P is true if both of these statements are true, and it is false if at least one of these statements is false.

2. If P is of the form $(Q)\vee(R)$ for some expressions Q and R, then these expressions are statements. Then P is true if at least one of these statements is true, and it is false if both of these statements is false.

3. Assume P is of the form $\bigwedge X(Q)$ for some variable X and some expression Q. Then P is true if for every primitives set B, the statement Q' is true, where Q' as in the previous lemma is obtained by replacing all occurrences of X in Q by the set B. The statement P is false if there exists some primiitve set B such that the string Q' thus obtained is false.

4. Assume P is of the form $\bigvee X(Q)$ for some variable X and some expression Q. Then P is true if there exists some set B for which the statement Q' is true, where Q' as in the previous lemma is obtained by replacing all occurrences of X in Q by the set B. The statement P is false if for all primitive sets B the statement Q' thus obtained is false.

Now we turn to tautologies.

Exercise 8 *All tautologies listed for the propositional logic remain tautologies in the obvious corresponding forms.*

Next we list tautologies involving quantifyers.

1. Commutativity/Associativity of $\bigwedge$: If P is a statement of the form $\bigwedge X(\bigwedge Y(Q))$ then also $\bigwedge X(\bigwedge Y(Q))$ is a statement and the two are tautological.

2. Commutativity/Associativity of $\bigvee$: If P is a statement of the form $\bigvee X(\bigvee Y(Q))$ then also $\bigvee X(\bigvee Y(Q))$ is a statement and the two are tautological.

3. Commutativity/Associativity of $\bigwedge$ and $\wedge$: If P is a statement not containing the variable X and $\bigwedge X(Q)$ is statement, then if both of the following expressions are statements and $(P) \wedge (\bigwedge X(Q))$ is tautological to $\bigwedge X((P) \wedge (Q))$

4. Commutativity/Associativity of $\bigvee$ and $\vee$: If P is a statement not containing the variable X and $\bigvee X(Q)$ is statement, then if both of the following expressions are statements and $(P) \vee (\bigvee X(Q))$ is tautological to $\bigvee X((P) \vee (Q))$

5. "Distributive law" of $\bigwedge$: If P is a statement not containing the variable X and $\bigwedge X(Q)$ is a statement, then if either of the following is a statement and $P \vee (\bigwedge X(Q))$ is tautological to $\bigwedge X((P) \vee (Q))$

6. "Distributive law" of $\bigvee$: If P is a statement not containing the variable X and $\bigvee X(Q)$ is a statement, then if either of the following is a statement and $P \wedge (\bigvee X(Q))$ is tautological to $\bigvee X((P) \wedge (Q))$

There is a specific tautology related to the use of variables for primitive sets:

1. If P is a statement containing a variable X and not containing a variable Y, Then the expression P' obtained by replacing all occurrences of X in P with Y is tautological to P.

Next we list rules of inference.

1. If either of $\bigwedge X(\bigvee Y(P))$ and $\bigvee Y(\bigwedge X(P))$ is a statement, then so is the other one. If $\bigwedge X(\bigvee Y(P))$ is true then $\bigvee Y(\bigwedge X(P))$ is also true.

2. If either of $\bigwedge X(\bigvee Y(P))$ and $\bigvee Y(\bigwedge X(P))$ is a statement, then so is the other one. If $\bigvee Y(\bigwedge X(P))$ is false then $\bigwedge X(\bigvee Y(P))$ is false.

3. If $\bigwedge X(Q)$ is an expression inside a statement P, and B is a specific primitive set, P being true implies P' being true where P' comes about by replacing the expression $\bigwedge X(Q)$ with Q' and Q' is the expression Q where all occurrences of X are replaced by B.

4. If $\bigwedge Y(R)$ or $\bigvee Y(R)$ is an expression inside a statement P, and $\bigwedge X(Q)$ is a statement inside R, then P being true implies that P' is true where P' comes about by replacing $\bigwedge X(Q)$ by Q' and Q' is the statement where all occurrences of X are replaced by Y.

5. If Q is an expression inside a statement P, and Y a set that occurs in Q, then P being false implies P' being false where P' comes about by replacing the expression Q by $\bigwedge X(Q')$ with Q' and Q' is the expression Q where some number of occurrence of the set B is replaced by X.

6. If $\bigwedge Y(R)$ or $\bigvee Y(R)$ is an expression inside a statement P, and Q is an expression inside R, then P being false implies P' being false where P' comes about by replacing the expression Q by $\bigwedge X(Q')$ and Q' is the expression Q where some number of occurrences of Y is replaced by X.

7. If $\bigvee X(Q)$ is an expression inside a statement P, and B is a specific primitive set, then P being false implies P' being false where P' comes about by replacing the expression $\bigvee X(Q)$ with Q' and Q' is the expression Q where all occurrences of X are replaced by B.

8. If $\bigwedge Y(R)$ or $\bigvee Y(R)$ is an expression inside a statement P, and $\bigvee X(Q)$ is a statement inside R, then P being false implies that P' is false where P' comes about by replacing $\bigvee X(Q)$ by Q' and Q' is the statement where all occurrences of X are replaced by Y.

9. If Q is an expression inside a statement P, and Y a set that occurs in Q, then P being true implies P' being true where P' comes about by replacing the expression Q by $\bigvee X(Q')$ with Q' and Q' is the expression Q where some number of occurrence of the set B is replaced by X.

10. If $\bigwedge Y(R)$ or $\bigvee Y(R)$ is an expression inside a statement P, and Q is an expression inside R, then P being true implies P' being true where P' comes about by replacing the expression Q by $\bigvee X(Q')$ and Q' is the expression Q where some number of occurrences of Y is replaced by X.

The universe of founded sets

Recall that every primitive set. Our next development is that we wopuld like to talk within our theory also concepts about primitive sets. In other words we would like to add primitive sets to the collection of objects in addition to the primitive objects. These collections of primitive sets will form what we call founded sets.

For example

$$\{\{\mathbf{ab}\}\{\mathbf{cd}\}\}$$

is a founded set contianing the two founded sets $\{\mathbf{ab}\}$ and $\{\mathbf{cd}\}$. A founded set may also contain objects of different types, for example

$$\{\mathbf{ab}\{\mathbf{ab}\}\{\}\}$$

is a founded set containing the four objects a, b, $\{\mathbf{cd}\}$ and $\{\}$. Note that readability of these strings is enhanced by the use of the concept of bracket balance. A new object of the global founded set in question starts whenever the bracket balance is 2 and stops when the bracket balance goes down to 1.

Note that an issue occurs with the order in which we list the elements of the founded set. For example the founded set

$$\{\{\mathbf{cd}\}\{\mathbf{ab}\}\}$$

contains the same elements as a previously displayed founded set, but given by a different string. In the case of primitive sets we solved this issue by standardizing the notation for primitive sets, only allowing the primitive objects be listed in a prescribed order. It is possible but cumbersome to define an order in the theory of founded sets we envision, but we shall not have the need to do so and instead agree that different strings may denote the same founded set in a manner to be made precise below.

Note that we have now infinitely many objects, as the list

$$\{\}, \{\{\}\}, \{\{\{\}\}\}, \ldots$$

suggests. Note that this list is obtained without the use of the primitive objects. These infinitely many objects are sufficient to encode any number of things we like to talk about, without having to use specific symbols for primitive objects. We shall thus now change our primitive universe to contain no primitive objects. Then the only primitive founded set is the empty set $\{\}$.

We begin a formla definiiton with a recursive definiiton of valid strings:

Definition 8 *1. $\{\}$ is a valid string*

2. *If C is a valid string, then $\{C\}$ is a valid string.*

3. *If C agnd D are valid strings of the form $\{E\}$ and $\{\{F\}\}$ where E is a valid string while F is not necessarily valid (and thus may be the empty string), then $\{E\{F\}\}$ is a valid string.*

For the sake of recursive definitions we make the following observation:

Lemma 3 *Given a valid string, one can determine the last step in the recursion to define the string. The two valid strings used to define the new valid string have shorter length than the new valid string.*

Lemma 4 *A string containing only opening and closing bracket $\{$ and $\}$ is a valid string if the bracket balance is never negative and it takes the value 0 only with the very last symbol.*

A statement about valid strings is of the tyupe $C \in D$ or $C \notin D$ for valid strings C and D. Each such statement has a truth value defined by the following recursion (which is well defined as foir a valid string we can determine the last step in the recursion.

Definition 9 1. *For every valid string C we have that $c \in \{\}$ is false and $C \notin \{\}$ is true.*

2. *If C is a valid string of the form $\{D\}$ for some valid string D, then $E \in C$ is true if E is the string D and it is false if E is not the string D. Moreover $E \notin C$ is false if E is the string D and it is true if E is not the string D.*

3. *If C is of the form $\{E\{F\}\}$ for some valid string E and some valid string $\{F\}$, then $D \in C$ is true if D is the string E or $D \in \{F\}$ is true, and it is false if $D \in C$ is true if D is not string E and $D \in \{F\}$ is false. Moreover $D \notin C$ is true if $D \in C$ is false and $D \notin C$ is false if $D \in C$ is true.*

We now define an equivalence relation $\sim$ on valid strings

Definition 10 *let C and D be two valid strings.*

1. *If C is $\{\}$, then $C \sim D$ if D is also the string $\{\}$.*

2. *If C is of the form $\{E\}$ for some valid string E, then $C \sim D$ if D is also the string $\{E\}$*

3. *If C is of the form $\{E\{F\}\}$ with a valid string E and an arbitrary string F, then $C \sim D$ if D is the same string $\{E\{F\}\}$ or there exist two valid strings G and $\{H\}$ such that $\{F\} \sim \{G, \{H\}\}$ and D is the string $\{G\{K\}\}$ with $\{K\} \sim \{E\{H\}\}$.*

Exercise 9 *The relation $\sim$ satisfies the following properties. For all valid strings C, D, E, we have*

1. $C \sim C$
2. *If $C \sim D$, then $D \sim C$*
3. *If $C \sim D$ and $D \sim E$ then $C \sim E$*

Exercise 10 *If $C \sim D$ and $E \sim F$ then $C \in E$ implies $D \in F$. If $C \sim D$ and $E \sim F$ then $C \notin E$ implies $D \notin F$.*

Exercise 11 *Two equivalent valid strings have the same length.*

Definition 11

1. *A founded set consists of a nonempty collection of valid strings that are all equivalent to each other, and no string equivalent to any of them is not in the collection.*
2. *If AS and B are founded stes, we define $A \in B$ if $C \in D$ for any members representibg A and B.*
3. *Let AS and B are founded sets. We define $A \in B$ to be true if $C \in D$ is true for some members representing A and B. We define $A \in B$ to be false if $C \in D$ is false for some members representing A and B. We define $A \notin B$ to be true if $C \notin D$ is true for some members representing A and B. We define $A \notin B$ to be false if $C \notin D$ is false for some members representing A and B.*

Exercise 12 *If A and B are founded sess, then exactly one of $A \in B$ and $A \notin B$ is true and exactly one of them is false. If one is true, then the other one is false.*

An axiomatic approach to finite set theory

In the previosu section we have introduced a new universe of statements of the type $A \in B$ or $A \notin B$ with two founded sets A and B. One can develop the propositional and predicate logic for this universe just as well. For the propositional logic we did not use any particularities of the particular collection of statements used. For predicate logic we only used that statements involved primitive sets and we could quantify over all primitive sets. Now we have statements involving founded sets and we may quantify over all founded sets. That founded sets enter somewhat differently into the statements, namely on both sides of the symbol $\in$ or $\notin$ is an invisible change with respect to the buildup of predicate logic. Thus we have all of predicate logic still in place. One added difficulty is that we now have infinitely many elementary statements (those that did ot involve any symbols) $\wedge$, $\vee$, or $\bigwedge$, $\bigvee$ and in particular quantifying over a variable is a combination of infinitely many statements. It may then be impossibly to check all these statements indivuidually. Nowhere however appears this difference in the explicit buildup of predicate logic.

There are statements that are true about founded sets.

Lemma 5

$$\bigvee X(\bigwedge Y(Y \notin X))$$

Note: This lemma in essence proclaims the existence of (at least one) empty set. Proof: We need to insert an element for X, we choose $\{\}$. The we have to prove $\bigwedge Y(Y \notin \{\})$. But Y is a string of length at least two, and if $C \in D$ for some representatives sets, then the length of C is alwasy less than the lenght of D. □

Exercise 13

$$\bigwedge X(\bigvee Y(\bigwedge Z(\bigwedge U((U \notin Z) \vee (((U \in Z) \wedge (U \in E)) \vee ((U \notin Z) \wedge (U \notin X)))))))$$

Note: This lemma proclaims informally for the set $[C]$ the existence of some set that has the same properties as the set $[\{C\}]$ where C denotes a valid string and for any valid string D the symbol $[D]$ denotes the equivalence class of the valid string D.

Lemma 6 *Formulate and prove a statement that proclaims the existence of a set that has the same property as the one coming about from the third method of construction of a valid string.*

Proof: exercise □

We now could turn more abstract and ignore any knowledge about the collection of things (so far primitive objects and primitive sets in one example and founded sets in another example) that appear on either side of the sign $\in$ or $\notin$ and how we deduce truth value of such statements. For simplicity we assume to have only one kind of things that are put on both sides of the sign $\in$ or $\notin$. Such things we shall call classes.

If in any theory of classes we have the statement of Lemma **??** be true, then we are guaranteed a class that looks like the empty set in the theory of founded sets. Similarly, if ina ddition the statements of Lemma 13 and Lemma 6 are true, then intuitivley (to be proven) we may build up classes that look like any of the founded sets. This is the axiomatic approach to the theory of founded sets, which is also called finite set theory.

We make the followign definition of a class

Definition 12 *Given a statement P of the form*

$$\bigvee X(\bigwedge Y(\bigwedge Z((Q) \wedge (((\overline{P}) \vee ((X \in Z) \wedge (Y \in Z)) \vee ((X \notin Z) \wedge (Y \notin Z)))\wedge$$

$$((\overline{P}) \vee ((Z \in X) \wedge (Z \in Y)) \vee ((Z \notin X) \wedge (Z \notin Y)))))))$$

where Q is an expression not containing Y and P is an expression not containing X and P comes about from replacing all occurrences of X in Q by Y, if this statement is true, we say Q is a preclass.

Definition 13 *Two preclasses are equivalent, $P \sim Q$, if ...*