13.3 How to do Proofs: "Proof Machine"

There *is* a certain "formula" or method to doing proofs. Some of the guidelines are given below. The most important factor in learning to do proofs is practice, just as when one is learning a new language.

1. There are very few words needed in the structure of a proof. Organized in rows by synonyms they are:

   ```
   To show
   Assume, Let, Suppose, Define, If
   Since, Because, By
   Then, Thus, So
   There exists, There is
   Recall, We know, But
   ```

   Do not use '`for all`' or '`for each`'. These can always be replaced by '`if`' to achieve greater clarity, accuracy and efficiency.

   Do not use the phrase '`for some`'. It can always be replaced by '`There exists`' to achieve greater clarity, accuracy and efficiency.

2. The overall structure of a proof is a block structure like an outline. For example:

   ```
   To show:  If __A__ then __B__ and __C__.
       Assume:  __A__.
   [itemsep=-0.2em]
   To show:  (a) __B__.
             (b) __C__.
   (a) To show:  __B__.
           _______.
           _______.
           _______.
       Thus __B__.
   (b) To show:  __C__.
           _______.
           _______.
           _______.
       Thus __C__.
   So __B__ and __C__.
   So, if __A__ then __B__ and __C__.
   ```

3. Any proof or section of proof begins with one of the following:

   (a) `To show:  If __A__ then __B__.`
   (b) `To show:  There exists __C__ such that __D__.`
   (c) `To show:  __E__.`

4. Immediately following this, the next step is

Case (a) Assume the ifs and 'To show' the thens. The next lines are

- `Assume __A__.`
- `To show: __B__.`

Case (b) To show an object exists you must find it. The next lines are

- `Define __xxx = ________.`
- `To show: __xxx__ satisfies __D__.`

Case (c) Rewrite the statement in __E__ by using a definition. The next line is

- `To show:` __E′__.

There are some kinds of proofs which have a special structure.

(E) Proofs of equality: LHS=RHS.

```
To show: __A=B__.
    Left Hand Side: __A= ···______
                    ____= ···_____
                    ____= ···_____
                    ____= ···_____
                    ____= expression____
    Right Hand Side: __B= ···______
                     ____= ···_____
                     ____= ···_____
                     ____= ···_____
                     ____= THE SAME expression____
```

(F) Counterexamples: Proofs of falseness

To show that a statement, "If ____ then ____", is false you *must* give an example.

`To show: There exists a __xxx__ such that`
(a) `xxx` satisfies the `ifs` of the statement that you are showing is false,
(a) `xxx` satisfies the opposite of some assertion in the `then`s of the statement that you are showing is false.

(U) Proofs of uniqueness.

To show that an object is unique you must show that if there are two of them then they are really the same.

`To show:` A THING `is unique.`
`Assume` X_1 `and` X_2 `are both` THINGs.
`To show:` $X_1 = X_2$.

(I) Proofs by induction.

A statement to be proved by induction *must* have the form

`If` n `is a positive integer then __A__.`

The proof by induction should have the form

`Proof by induction.`
`Base case:`
`To show: If` $n = 1$ `then __A__.`
`______.`
`______.`
`______.`
`Thus, if` $n = 1$ `then __A__.`
`Induction step:`
`Let` ℓ `be a positive integer and assume that if` n `is a positive integer and` $n < \ell$ `then __A__.`
`To show: __A__.`

The mechanics of proof by induction is an unwinding of the *definition* of $\mathbb{Z}_{>0}$.

(CP) Proofs by contrapositive.

`To show: If __A__ then __B__.`
`To show: If __not B__ then __not A__.`

(BAD) Proofs by contradiction.

(*) `Assume` the opposite of what you want to show.
________.
________.
________.
End up showing the opposite of some assumption (not necessarily the (*) assumption).
`Contradiction to` specify exactly what assumption is being contradicted.
`Thus` assumption (*) is wrong and what you want to show is true.

PROOFS BY CONTRADICTION ARE STRONGLY DISCOURAGED. In all known cases they can be replaced by a proof by contrapositive for greater clarity, direction and efficiency.