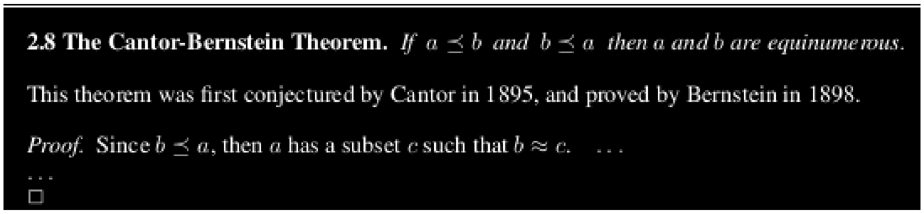
1 The Requirements Document

1. Phases of the Life Cycle of Program Development:
   - system analysis- feasibility study of system we want to construct.
   - **requirements document- states clearly what the functions and constraints of the system are.(Written in Natural language)**
   - technical specification- contains structured formalization of the previous document using some modelling technique.
   - design- develops the previous phase by taking and justfying the decisions which impelment the previous specification, and also defines the architecture of the future systems.
   - implementation- contains the translation of the outcome of the previous phase into hardware and software components.
   - tests- consists of the experimental verifications of the final system.
   - maintenance- contains the system upgrading.
2. Difficulties with the Requirements Doc?
   - What is difficult for the reader of the requirement document is to make distinctions between which part of text is devoted to **explanations** and which is devoted to genuine **requirements**.
   - Explanations are needed initially for the reader to understand the future system. But when the reader is more acquainted with the purpose of the system, explanations are less important.
3. In mathematical texts, requirements are Defintions and Theorems.
   - Such items are usually easily recognizable because they are labeled by their function(defition, lemma, theorem).
   - Also, numbered in systematic fashion.
   - Usually differs in font which differs from that used elsewhere in the book
   - 

     2.8 The Cantor-Bernstein Theorem. *If $a \preceq b$ and $b \preceq a$ then a and b are equinumerous.*

     This theorem was first conjectured by Cantor in 1895, and proved by Bernstein in 1898.

     Proof. Since $b \preceq a$, then a has a subset c such that $b \approx c$. . . .

     . . .

     □

   -
     ```
     we can clearly see the \requirement" as in-dicated on the first line:
     the theorem number, the theorem name, and the theorem statement (written
     Next are the associated \explanations": historical comments and proof.
     ```

4. the idea of stucturing the reqequirements doc is to have our requirement document organized around two texts embedded in each other: the explanatory text and the reference text.
    - These two texts should be immediately separable, so that it is possible to summarize the reference text independently.
    - reference text takes the form of labeled and numbered short statements written using natural language, , which must be very easy to read independently from the explanatory text.
    - We shall use a special font for the reference text.
    - These fragments must be self contained without the explanations.
    - They together form the requirements.
    - The explanations are just there to give some comments which could help a first reader. But after an initial period, the reference text is the only one that counts.
5. Labeling and Numbering the requiremnts FUN: for functional requirements; ENV:for environment requiremnts, is important for traceability.
    - So that it will be easy to recognize how each requirement has indeed been taken into account during the construction of our system and in its final operational version.
6.