Summary of last lecture

- Some facts on logic
- Representing functions as input/output lists/tables, decision problems and languages
- Computing devices solving problems by accepting (recognizing) languages

Comparison of computation models

- We would like to claim “Model A can compute function every function B can, and further it can compute a function f, but model B cannot’’ which would imply model A is strictly more powerful than B.
- What we will (typically) prove is “Model A accepts every language model B accepts and in addition a language L that model B cannot accept”

Proof by contradiction - 2

The Pigeonhole Principle

- If $n+1$ or more objects are placed into n boxes, then there is at least one box containing two or more of the objects

  In a set of any 27 English words, at least two words must start with the same letter

- If n objects are placed into k boxes, then there is at least one box containing $\lceil n/k \rceil$ objects

Recursively defined sets

Close relationship to induction

Example: set of all palindromes

- $\varepsilon \in P$; $\forall a \in \Sigma$, $a \in P$;
- $\forall a \in \Sigma\ \forall x \in P$, $axa \in P$
- No other strings are in P

More definitions

Definition of Σ^*:

- $\varepsilon \in \Sigma^*$;
- $\forall$ a $\in \Sigma$, $\forall$ x $\in \Sigma^*$, xa $\in \Sigma^*$;
- No other strings are in Σ^*.

Exercise

Suppose $\Sigma = \{a,b\}$. Define L as

- $a \in L$;
- $\forall x \in L$, $ax \in L$
- $\forall x, y \in L$, bxy, xby and xyb are in L
- No other strings are in P

- Prove that this is the language of strings with more a's than b's.