CS F364
Design & Analysis of Algorithms

ALGORITHMS – DESIGN TECHNIQUES

Exact Solutions

- Search with Backtracking

- Application: Prolog

Evaluation of Prolog Queries (given a program):

- Prolog resolves queries against a given program :
  - A program is a set of rules
  - Resolution is achieved by
    - matching the query with rules to generate sub-queries;
    - a query is resolved if all sub-queries (generated recursively) are resolved.
  - Rules are searched for matching
    - When search fails for a sub-query or matching fails the resolver backtracks and searches another path.
      - i.e. ***backtracking is built into Prolog search engine***

Evaluation of Prolog Queries – Resolution steps

1. Match query term with the head of a rule
2. If matching succeeds, add each of the other sub-clauses as a query; continue;
3. If matching fails or sub-query fails, **backtrack**;
4. If no more rules to backtrack fail

Prolog Program and Query

- Evaluation of Prolog Queries (given a Prolog program):
  - A program is a set of rules in Horn Clause form
    - e.g.
      - grandparent(X,Y):-parent(X,Z), parent(Z,Y).
      - parent(X,Y):-father(X,Y).
      - parent(X,Y):-mother(X,Y).
      - mother(ada,bebe).
      - mother(bebe,bart).
      - father(bart,catniss).
  - A sample query:
    - grandparent(bebe,catniss)?

Evaluation of Prolog Queries - Example

- match “grandparent(bebe,catniss)” with “grandparent(X,Y)”
  - Step 1
- add “parent(bebe, Z)” and “parent(Z, catniss)” to list of queries
  - Step 2
- for resolving “parent(bebe, Z)” add “father(bebe,Z) as query.
  - Step 2

Evaluation of Prolog Queries – Example [contd..]

- "father(bebe,Z)" fails to match with father(bart, catniss) ;
  - so backtrack and look for another "father" rule. (Step 3)
- "father(bebe,Z)" fails;
  - so backtrack; (Step 4)
- this is subquery for "parent(bebe,Z)"
  - backtrack (step 3)
- add "mother(bebe,Z)" to list of queries
  - (step 2)
- ...

- Exercise: Completely resolve this example query.