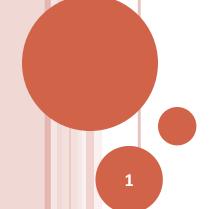
CS F364 Design & Analysis of Algorithms

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
 - o 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
- If matching succeeds, add each of the other sub-clauses as a query; continue;
- 3. <u>If matching fails or sub-query fails</u>, **backtrack**;
- 4. If <u>no more rules to backtrack</u> fail

PROLOG PROGRAM AND QUERY

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

EVALUATION OF PROLOG QUERIES - EXAMPLE

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

EVALUATION OF PROLOG QUERIES — EXAMPLE [CONTD..]

- "father(bebe,Z)" fails to match with father(bart, cathiss);
 - 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)
- **O** ...
- Exercise: Completely resolve this example query.