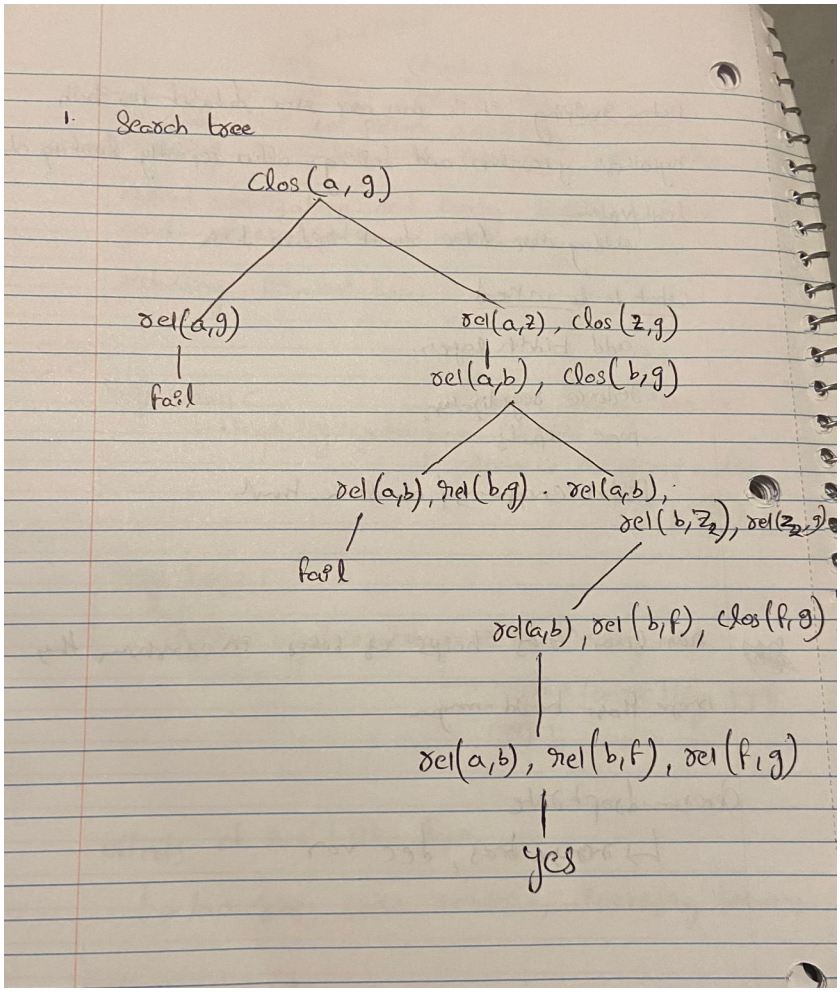
Programming Languages HW6

Sai Mannava

May 13, 2025

1 Search Tree



2 Prolog Unification

Determine success/failure and bindings:

1. `p(X,Y) = p(Y,X).`
   Succeeds. Solution:
   $$X = Y.$$

2. `q(X,X) = q(1,2).`
   Fails. Because $1 \neq 2$.

3. `m(f(X),Y) = m(f(a),b).`
   Succeeds. Bindings:
   $$X = a, \quad Y = b.$$

4. `k(X,Y) = k(a).`
   Fails. Arity mismatch: `k/2` vs. `k/1`.

5. `[A,B | X] = [1,2].`
   Succeeds. Bindings:
   $$A = 1, \quad B = 2, \quad X = [].$$

3 Peano Arithmetic (`exp/3`)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Peano arithmetic formalism
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

is_number(0).
is_number(s(X)) :- is_number(X).

add(0, N, N) :- is_number(N).
add(s(N), M, s(Y)) :- add(N, M, Y).

mul(0, N, 0) :- is_number(N).
mul(s(N), M, Y) :-
  mul(N, M, YY),
  add(M, YY, Y).

% Anything to the 0th power is 1 (which is s(0) in Peano)
exp(_, 0, s(0)).

% X^(s(Y)) = X * (X^Y)
exp(X, s(Y), Z) :-
    exp(X, Y, Z1),
    mul(X, Z1, Z).

% Example execution:
% swipl peano.pl
% ?- exp(s(s(0)), s(s(0)), Z).
% Z = s(s(s(s(0)))).
% ?- exp(s(s(0)), s(s(s(0))), Z).
% Z = s(s(s(s(s(s(s(s(0))))))))).
```

4 Binary Search Trees

```
% empty
% tree(Key, VaL, left, right)

% insert(key, val, in_tree, out_tree)
insert(K, V, empty, tree(K, V, empty, empty)).
insert(K, V, tree(TK, TV, Left, Right), tree(TK, TV, New_left, Right)) :-
  K < TK, insert(K, V, Left, New_left).
insert(K, V, tree(TK, TV, Left, Right), tree(TK, TV, Left, New_right)) :-
  K > TK, insert(K, V, Right, New_right).
insert(K, V, tree(K, _, Left, Right), tree(K, V, Left, Right)).

% lookup
% lookup(K, T, V)
% lookup(_, empty, no_val).
lookup(K, tree(K, V, _, _), V).
lookup(K, tree(TK, _, Left, _), V) :- K < TK, lookup(K, Left, V).
lookup(K, tree(TK, _, _, Right), V) :- K > TK, lookup(K, Right, V).

ltree(
    tree(3, c,
      tree(2, b,
        tree(1,a,empty,empty),
        empty
      ),
      empty
    )
).

ltreeR(
    tree(2,b,
      tree(1,a,empty,empty),
      tree(3,c,empty,empty)
    )
).

rtree(
    tree(1,a,
      empty,
      tree(2,b,
        empty,
        tree(3,c,
          empty,
          empty
        )
      )
    )
).

% Right rotation: promote the left child to the root
rotateRight(
  tree(K1, V1,
       tree(K2, V2, L2, R2),
       R1),
  tree(K2, V2,
       L2,
       tree(K1, V1, R2, R1))
).

% Left rotation: promote the right child to the root
rotateLeft(
  tree(K1, V1,
       L1,
       tree(K2, V2, L2, R2)),
  tree(K2, V2,
       tree(K1, V1, L1, L2),
```

```
        R2)
).

% Example execution:
% ?- ltree(T1), rotateRight(T1, T2).
% T1 = tree(3, c, tree(2, b, tree(1, a, empty, empty), empty), empty),
% T2 = tree(2, b, tree(1, a, empty, empty), tree(3, c, empty, empty)).
% ?- rtree(T1), rotateLeft(T1, T2).
% T1 = tree(1, a, empty, tree(2, b, empty, tree(3, c, empty, empty))),
% T2 = tree(2, b, tree(1, a, empty, empty), tree(3, c, empty, empty)).
```