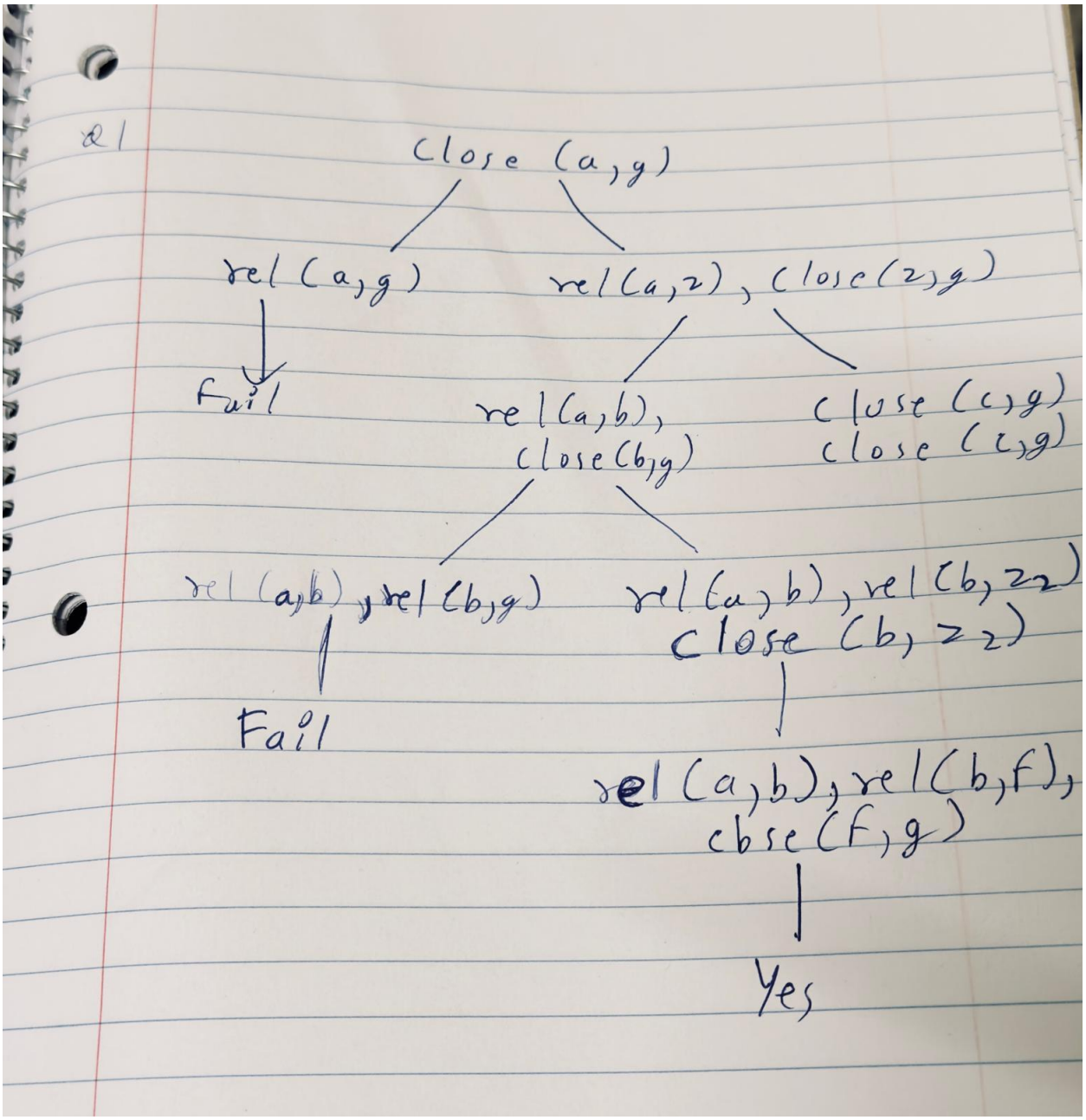
HW-6

Ans-1


Q1
close (a, g)
rel (a, g)
rel (a, z), close (z, g)
Fail
rel (a, b), close (b, g)
close (c, g)
close (c, g)
rel (a, b), rel (b, g)
rel (a, b), rel (b, z2)
close (b, z2)
Fail
rel (a, b), rel (b, f), cbse (f, g)
Yes

Ans-2:
1. p (X, Y) = p (Y, X) → Succeeds: X = Y
2. q (X, X) = q (1, 2) → Fails: X cannot both be 1 and 2
3. m(f(X), Y) = m(f(a), b) → Succeeds: X = a, Y = b
4. k (X, Y) = k(a) → Fails: Arity mismatch
5. [A, B | X] = [1, 2] → Succeeds: A = 1, B = 2, X = []

Ans-3:
In Peano arithmetic, we can formally define the predicate of exponentiation as:

```
%. Base case: any number raised to the power 0 equals 1.
exp(_, 0, s(0)).

% Recursive case: exp(X, Y, Z) if Y = s(Y1) => Z = X * exp(X, Y1)
exp(X, s(Y), Z) :-
exp(X, Y, Z1), mult(X, Z1, Z).
```

Ans-4:

Rotate left and right predicates for a BST:

```
rotateRight(tree(Z, KZ, tree(Y, KY, A, B), C), tree(Y, KY, A, tree(Z, KZ, B, C))).
rotateLeft(tree(X, KX, A, tree(Y, KY, B, C)), tree(Y, KY, tree(X, KX, A, B), C)).
ltree(tree(3, c, tree(2, b, tree(1, a, empty, empty), empty), empty)).
rtree(tree(1, a, empty, tree(2, b, empty, tree(3, c, empty, empty)))).
```