Backward Chaining General Notes 3

Testcase 1: ((p) (q (p)) (r (q s t)) (s (p u)) (r (q h)) (t) (h (t)) (u (v)))
Query: r

1. Goes to (p) not added to goal list because head does not equal top of goal list
2. Goes to (q (p)) not added to goal list because head does not equal top of goal list
3. Goes to (r (q s t)) adds q s t to goal list because head is equal to top of goal list
4. Recurses with q as car(goal)
5. Goes to (p) not added to goal list because head does not equal top of goal list
6. Goes to (q (p)) p added to goal list because head = a
7. Recurses with p as car(goal)
8. Goes to (p), proven because fact.
9. s = car(goal) now, goes through rules not adding until it gets to (s (p u)) adds p u to goal list because head = a
10. u = car(goal), goes through rule list not adding until (u (v)) adds v to goal list because head = a
11. v = car(goal), goes through list but can't prove v.
12. Returns back, with r = car(goal) goes to (r (q h)) adds q h to goal list because head = a
13. q = car(goal), goes to (q (p)) adds p to goal list because head = a
14. p = car(goal), p proven because fact.
15. h = car(goal), goes to (h (t)) adds t because head = a
16. t = car(goal), goes through list until (t), t is proven a fact
17. No more goals remain, success.

Testcase 2: ((q (p)) (p (l m)) (m (b l)) (l (a p)) (l (a b)) (a) (b))
Query: a

1. Goes through rules list until (a), a is proven because it is a fact.
2. No more goals remain, success.

Testcase 3: ((a) (b) (p (a b c d e)) (q (c e)) (c (a b)) (r (a b d)) (d))
Query: q

1. Goes through list not adding until (q (c e)) adds c e to goal list because head = a
2. c = car(goal) goes through list until (c (a b)), adds a b to goal list because head = a
3. a = car(goal) goes to (a), a proven because fact
4. b = car(goal) goes to (b), b proven because fact
5. e = car(goal) goes through list but cannot prove e
6. Cannot prove q, fail.