NOTES ON PROLOG EXECUTION

ERIC MARTIN

1. Program

```
concat(a, Y, Y).
concat(f(X), Y, f(Z)) :- concat(X, Y, Z).
concat(g(X), Y, g(Z)) :- concat(X, Y, Z).

compress(a, a).
compress(f(a), f(a)).
compress(g(a), g(a)).
compress(f(f(X)), Y) :- compress(f(X), Y).
compress(g(g(X)), Y) :- compress(g(X), Y).
compress(f(g(X)), f(Y)) :- compress(g(X), Y).
compress(g(f(X)), g(Y)) :- compress(f(X), Y).

accumulator_reverse(a, X, X).
accumulator_reverse(f(X), Y, Z) :- accumulator_reverse(X, f(Y), Z).
accumulator_reverse(g(X), Y, Z) :- accumulator_reverse(X, g(Y), Z).
reverse(X, Y) :-  accumulator_reverse(X, a, Y).

relation(X, Y, Z) :- concat(X, Y, U), compress(U, V), reverse(V, Z).
relation(X, Y, Z) :- concat(Y, X, U), compress(U, V), reverse(V, Z).
```

2. Query: relation(f(f(f(g(f(a))))), f(f(g(g(a)))), X)

```
goals_solution_list: [([relation(f(f(g(f(a)))), f(g(g(a))), X)], {X: X})]
    goals: [relation(f(f(g(f(a)))), f(g(g(a))), X)]    solution: {X: X}
    goal: relation(f(f(g(f(a)))), f(g(g(a))), X)
    clause: relation(X, Y, Z) :- concat(X, Y, U), compress(U, V), reverse(V, Z).    variable renaming: {X: X_0}
    head: relation(X_0, Y, Z)
    mgu: {X: Z, Y: f(g(g(a))), X_0: f(f(g(f(a))))}
    clause: relation(X, Y, Z) :- concat(Y, X, U), compress(U, V), reverse(V, Z).    variable renaming: {X: X_0}
    head: relation(X_0, Y, Z)
    mgu: {X: Z, Y: f(g(g(a))), X_0: f(f(g(f(a))))}

goals_solution_list: [([concat(f(f(g(f(a)))), f(g(g(a))), U), compress(U, V), reverse(V, Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [concat(f(f(g(f(a)))), f(g(g(a))), U), compress(U, V), reverse(V, Z)    solution: {X: Z}
    goal: concat(f(f(g(f(a)))), f(g(g(a))), U)
    clause: concat(f(X), Y, f(Z)) :- concat(X, Y, Z).    variable renaming: {X: X_0, Z: Z_0}
    head: concat(f(X_0), Y, f(Z_0))
    mgu: {U: f(Z_0), Y: f(g(g(a))), X_0: f(g(f(a)))}

goals_solution_list: [([concat(f(g(f(a))), f(g(g(a))), Z_0), compress(f(Z_0), V), reverse(V, Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [concat(f(g(f(a))), f(g(g(a))), Z_0), compress(f(Z_0), V), reverse(V, Z)]    solution:{X: Z}
    goal: concat(f(g(f(a))), f(g(g(a))), Z_0)
    clause: concat(f(X), Y, f(Z)) :- concat(X, Y, Z).    variable renaming: {X: X_0, Z: Z_1}
    head: concat(f(X_0), Y, f(Z_1))
    mgu: {Z_0: f(Z_1), Y: f(g(g(a))), X_0: g(f(a))}
```

Date: Session 1, 2018.

```
goals_solution_list: [([concat(g(f(a)), f(g(g(a))), Z_1), compress(f(f(Z_1)), V), reverse(V, Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [concat(g(f(a)), f(g(g(a))), Z_1), compress(f(f(Z_1)), V), reverse(V, Z)]    solution: {X: Z}
    goal: concat(g(f(a)), f(g(g(a))), Z_1)
    clause: concat(g(X), Y, g(Z)) :- concat(X, Y, Z).    variable renaming: {X: X_0, Z: Z_0}
    head: concat(g(X_0), Y, g(Z_0))
    mgu: {Z_1: g(Z_0), Y: f(g(g(a))), X_0: f(a)}

goals_solution_list: [([concat(f(a), f(g(g(a))), Z_0), compress(f(f(g(Z_0))), V), reverse(V, Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [concat(f(a), f(g(g(a))), Z_0), compress(f(f(g(Z_0))), V), reverse(V, Z)]    solution: {X: Z}
    goal: concat(f(a), f(g(g(a))), Z_0)
    clause: concat(f(X), Y, f(Z)) :- concat(X, Y, Z).    variable renaming: {X: X_0, Z: Z_1}
    head: concat(f(X_0), Y, f(Z_1))
    mgu: {Z_0: f(Z_1), Y: f(g(g(a))), X_0: a}

goals_solution_list: [([concat(a, f(g(g(a))), Z_1), compress(f(f(g(f(Z_1)))), V), reverse(V, Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [concat(a, f(g(g(a))), Z_1), compress(f(f(g(f(Z_1)))), V), reverse(V, Z)]    solution: {X: Z}
    goal: concat(a, f(g(g(a))), Z_1)
    clause: concat(a, Y, Y).    variable renaming: {}
    head: concat(a, Y, Y)
    mgu: {Z_1: Y, Y: f(g(g(a)))}

goals_solution_list: [([compress(f(f(g(f(f(g(g(a))))))), V), reverse(V, Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [compress(f(f(g(f(f(g(g(a))))))), V), reverse(V, Z)]    solution: {X: Z}
    goal: compress(f(f(g(f(f(g(g(a))))))), V)
    clause: compress(f(f(X)), Y) :- compress(f(X), Y).    variable renaming: {X: X_0}
    head: compress(f(f(X_0)), Y)
    mgu: {V: Y, X_0: g(f(f(g(g(a)))))}

goals_solution_list: [([compress(f(g(f(f(g(g(a)))))), Y), reverse(Y, Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [compress(f(g(f(f(g(g(a)))))), Y), reverse(Y, Z)]    solution: {X: Z}
    goal: compress(f(g(f(f(g(g(a)))))), Y)
    clause: compress(f(g(X)), f(Y)) :- compress(g(X), Y).    variable renaming: {X: X_0, Y: Y_0}
    head: compress(f(g(X_0)), f(Y_0))
    mgu: {Y: f(Y_0), X_0: f(f(g(g(a))))}

goals_solution_list: [([compress(g(f(f(g(g(a))))), Y_0), reverse(f(Y_0), Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [compress(g(f(f(g(g(a))))), Y_0), reverse(f(Y_0), Z)]    solution: {X: Z}
    goal: compress(g(f(f(g(g(a))))), Y_0)
    clause: compress(g(f(X)), g(Y)) :- compress(f(X), Y).    variable renaming: {X: X_0}
    head: compress(g(f(X_0)), g(Y))
    mgu: {Y_0: g(Y), X_0: f(g(g(a)))}

goals_solution_list: [([compress(f(f(g(g(a)))), Y), reverse(f(g(Y)), Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [compress(f(f(g(g(a)))), Y), reverse(f(g(Y)), Z)]    solution: {X: Z}
    goal: compress(f(f(g(g(a)))), Y)
    clause: compress(f(f(X)), Y) :- compress(f(X), Y).    variable renaming: {X: X_0, Y: Y_0}
    head: compress(f(f(X_0)), Y_0)
    mgu: {Y: Y_0, X_0: g(g(a))}

goals_solution_list: [([compress(f(g(g(a))), Y_0), reverse(f(g(Y_0)), Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [compress(f(g(g(a))), Y_0), reverse(f(g(Y_0)), Z)]    solution: {X: Z}
    goal: compress(f(g(g(a))), Y_0)
    clause: compress(f(g(X)), f(Y)) :- compress(g(X), Y).    variable renaming: {X: X_0}
    head: compress(f(g(X_0)), f(Y))
    mgu: {Y_0: f(Y), X_0: g(a)}
```

```
goals_solution_list: [([compress(g(g(a)), Y), reverse(f(g(f(Y))), Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [compress(g(g(a)), Y), reverse(f(g(f(Y))), Z)]    solution: {X: Z}
    goal: compress(g(g(a)), Y)
    clause: compress(g(g(X)), Y) :- compress(g(X), Y).    variable renaming: {X: X_0, Y: Y_0}
    head: compress(g(g(X_0)), Y_0)
    mgu: {Y: Y_0, X_0: a}

goals_solution_list: [([compress(g(a), Y_0), reverse(f(g(f(Y_0))), Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [compress(g(a), Y_0), reverse(f(g(f(Y_0))), Z)]    solution: {X: Z}
    goal: compress(g(a), Y_0)
    clause: compress(g(a), g(a)).    variable renaming: {}
    head: compress(g(a), g(a))
    mgu: {Y_0: g(a)}

goals_solution_list: [([reverse(f(g(f(g(a)))), Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [reverse(f(g(f(g(a)))), Z)]    solution: {X: Z}
    goal: reverse(f(g(f(g(a)))), Z)
    clause: reverse(X, Y) :- accumulator_reverse(X, a, Y).    variable renaming: {X: X_0}
    head: reverse(X_0, Y)
    mgu: {Z: Y, X_0: f(g(f(g(a))))}

goals_solution_list: [([accumulator_reverse(f(g(f(g(a)))), a, Y)], {X: Y}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [accumulator_reverse(f(g(f(g(a)))), a, Y)]    solution: {X: Y}
    goal: accumulator_reverse(f(g(f(g(a)))), a, Y)
    clause: accumulator_reverse(f(X), Y, Z) :- accumulator_reverse(X, f(Y), Z).    variable renaming: {X: X_0, Y: Y_0}
    head: accumulator_reverse(f(X_0), Y_0, Z)
    mgu: {Y: Z, Y_0: a, X_0: g(f(g(a)))}

goals_solution_list: [([accumulator_reverse(g(f(g(a))), f(a), Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [accumulator_reverse(g(f(g(a))), f(a), Z)]    solution: {X: Z}
    goal: accumulator_reverse(g(f(g(a))), f(a), Z)
    clause: accumulator_reverse(g(X), Y, Z) :- accumulator_reverse(X, g(Y), Z).    variable renaming: {X: X_0, Z: Z_0}
    head: accumulator_reverse(g(X_0), Y, Z_0)
    mgu: {Z: Z_0, Y: f(a), X_0: f(g(a))}

goals_solution_list:[([accumulator_reverse(f(g(a)), g(f(a)), Z_0)], {X: Z_0}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [accumulator_reverse(f(g(a)), g(f(a)), Z_0)]    solution: {X: Z_0}
    goal: accumulator_reverse(f(g(a)), g(f(a)), Z_0)
    clause: accumulator_reverse(f(X), Y, Z) :- accumulator_reverse(X, f(Y), Z).    variable renaming: {X: X_0}
    head: accumulator_reverse(f(X_0), Y, Z)
    mgu: {Z_0: Z, Y: g(f(a)), X_0: g(a)}

goals_solution_list: [([accumulator_reverse(g(a), f(g(f(a))), Z)], {X: Z}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [accumulator_reverse(g(a), f(g(f(a))), Z)]    solution: {X: Z}
    goal: accumulator_reverse(g(a), f(g(f(a))), Z)
    clause: accumulator_reverse(g(X), Y, Z) :- accumulator_reverse(X, g(Y), Z).    variable renaming: {X: X_0, Z: Z_0}
    head: accumulator_reverse(g(X_0), Y, Z_0)
    mgu: {Z: Z_0, Y: f(g(f(a))), X_0: a}

goals_solution_list: [([accumulator_reverse(a, g(f(g(f(a)))), Z_0)], {X: Z_0}),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [accumulator_reverse(a, g(f(g(f(a)))), Z_0)]    solution: {X: Z_0}
    goal: accumulator_reverse(a, g(f(g(f(a)))), Z_0)
    clause: accumulator_reverse(a, X, X).    variable renaming: {X: X_0}
    head: accumulator_reverse(a, X_0, X_0)
    mgu: {Z_0: X_0, X_0: g(f(g(f(a))))}

goals_solution_list: [([], X: g(f(g(f(a))))),
                      ([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: []    solution: {X: g(f(g(f(a))))}
    YIELDED SOLUTION: {X: g(f(g(f(a))))}
```

```
goals_solution_list: [([concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)], {X: Z})]
    goals: [concat(f(g(g(a))), f(f(g(f(a)))), U), compress(U, V), reverse(V, Z)]    solution: {X: Z}
    goal: concat(f(g(g(a))), f(f(g(f(a)))), U)
    clause: concat(f(X), Y, f(Z)) :- concat(X, Y, Z).    variable renaming: {X: X_0, Z: Z_0}
    head: concat(f(X_0), Y, f(Z_0))
    mgu: {U: f(Z_0), Y: f(f(g(f(a)))), X_0: g(g(a))}

goals_solution_list: [([concat(g(g(a)), f(f(g(f(a)))), Z_0), compress(f(Z_0), V), reverse(V, Z)], {X: Z})]
    goals: [concat(g(g(a)), f(f(g(f(a)))), Z_0), compress(f(Z_0), V), reverse(V, Z)]    solution: {X: Z}
    goal: concat(g(g(a)), f(f(g(f(a)))), Z_0)
    clause: concat(g(X), Y, g(Z)) :- concat(X, Y, Z).    variable renaming: {X: X_0, Z: Z_1}
    head: concat(g(X_0), Y, g(Z_1))
    mgu: {Z_0: g(Z_1), Y: f(f(g(f(a)))), X_0: g(a)}

goals_solution_list:  [([concat(g(a), f(f(g(f(a)))), Z_1), compress(f(g(Z_1)), V), reverse(V, Z)], {X: Z})]
    goals: [concat(g(a), f(f(g(f(a)))), Z_1), compress(f(g(Z_1)), V), reverse(V, Z)]    solution: {X: Z}
    goal: concat(g(a), f(f(g(f(a)))), Z_1)
    clause: concat(g(X), Y, g(Z)) :- concat(X, Y, Z).    variable renaming: {X: X_0, Z: Z_0}
    head: concat(g(X_0), Y, g(Z_0))
    mgu: {Z_1: g(Z_0), Y: f(f(g(f(a)))), X_0: a}

goals_solution_list: [([concat(a, f(f(g(f(a)))), Z_0), compress(f(g(g(Z_0))), V), reverse(V, Z)], {X: Z})]
    goals: [concat(a, f(f(g(f(a)))), Z_0), compress(f(g(g(Z_0))), V), reverse(V, Z)]    solution: {X: Z}
    goal: concat(a, f(f(g(f(a)))), Z_0)
    clause: concat(a, Y, Y).    variable renaming: {}
    head: concat(a, Y, Y)
    mgu: {Z_0: Y, Y: f(f(g(f(a))))}

goals_solution_list: [([compress(f(g(g(f(f(g(f(a))))))), V), reverse(V, Z)], {X: Z})]
    goals: [compress(f(g(g(f(f(g(f(a))))))), V), reverse(V, Z)]    solution: {X: Z}
    goal: compress(f(g(g(f(f(g(f(a))))))), V)
    clause: compress(f(g(X)), f(Y)) :- compress(g(X), Y).    variable renaming: {X: X_0}
    head: compress(f(g(X_0)), f(Y))
    mgu: {V: f(Y), X_0: g(f(f(g(f(a)))))}

goals_solution_list: [([compress(g(g(f(f(g(f(a)))))), Y), reverse(f(Y), Z)], {X: Z})]
    goals: [compress(g(g(f(f(g(f(a)))))), Y), reverse(f(Y), Z)]    solution: {X: Z}
    goal: compress(g(g(f(f(g(f(a)))))), Y)
    clause: compress(g(g(X)), Y) :- compress(g(X), Y).    variable renaming: {X: X_0, Y: Y_0}
    head: compress(g(g(X_0)), Y_0)
    mgu: {Y: Y_0, X_0: f(f(g(f(a))))}

goals_solution_list: [([compress(g(f(f(g(f(a))))), Y_0), reverse(f(Y_0), Z)], {X: Z})]
    goals: [compress(g(f(f(g(f(a))))), Y_0), reverse(f(Y_0), Z)]    solution: {X: Z}
    goal: compress(g(f(f(g(f(a))))), Y_0)
    clause: compress(g(f(X)), g(Y)) :- compress(f(X), Y).    variable renaming: {X: X_0}
    head: compress(g(f(X_0)), g(Y))
    mgu: {Y_0: g(Y), X_0: f(g(f(a)))}

goals_solution_list: [([compress(f(f(g(f(a)))), Y), reverse(f(g(Y)), Z)], {X: Z})]
    goals: [compress(f(f(g(f(a)))), Y), reverse(f(g(Y)), Z)]    solution: {X: Z}
    goal: compress(f(f(g(f(a)))), Y)
    clause: compress(f(f(X)), Y) :- compress(f(X), Y).    variable renaming: {X: X_0, Y: Y_0}
    head: compress(f(f(X_0)), Y_0)
    mgu: {Y: Y_0, X_0: g(f(a))}

goals_solution_list: [([compress(f(g(f(a))), Y_0), reverse(f(g(Y_0)), Z)], {X: Z})]
    goals: [compress(f(g(f(a))), Y_0), reverse(f(g(Y_0)), Z)]    solution: {X: Z}
    goal: compress(f(g(f(a))), Y_0)
    clause: compress(f(g(X)), f(Y)) :- compress(g(X), Y).    variable renaming: {X: X_0}
    head: compress(f(g(X_0)), f(Y))
    mgu: {Y_0: f(Y), X_0: f(a)}

goals_solution_list: [([compress(g(f(a)), Y), reverse(f(g(f(Y))), Z)], {X: Z})]
    goals: [compress(g(f(a)), Y), reverse(f(g(f(Y))), Z)]    solution: {X: Z}
    goal: compress(g(f(a)), Y)
    clause: compress(g(f(X)), g(Y)) :- compress(f(X), Y).    variable renaming: {X: X_0, Y: Y_0}
    head: compress(g(f(X_0)), g(Y_0))
    mgu: {Y: g(Y_0), X_0: a}
```

```
goals_solution_list: [([compress(f(a), Y_0), reverse(f(g(f(g(Y_0)))), Z)], {X: Z})]
    goals: [compress(f(a), Y_0), reverse(f(g(f(g(Y_0)))), Z)]    solution: {X: Z}
    goal: compress(f(a), Y_0)
    clause: compress(f(a), f(a)).    variable renaming: {}
    head: compress(f(a), f(a))
    mgu: {Y_0: f(a)}

goals_solution_list: [([reverse(f(g(f(g(f(a))))), Z)], {X: Z})]
    goals: [reverse(f(g(f(g(f(a))))), Z)]    solution: {X: Z}
    goal: reverse(f(g(f(g(f(a))))), Z)
    clause: reverse(X, Y) :- accumulator_reverse(X, a, Y).    variable renaming: {X: X_0}
    head: reverse(X_0, Y)
    mgu: {Z: Y, X_0: f(g(f(g(f(a)))))}

goals_solution_list: [([accumulator_reverse(f(g(f(g(f(a))))), a, Y)], {X: Y})]
    goals: [accumulator_reverse(f(g(f(g(f(a))))), a, Y)]    solution: {X: Y}
    goal: accumulator_reverse(f(g(f(g(f(a))))), a, Y)
    clause: accumulator_reverse(f(X), Y, Z) :- accumulator_reverse(X, f(Y), Z).    variable renaming: {X: X_0, Y: Y_0}
    head: accumulator_reverse(f(X_0), Y_0, Z)
    mgu: {Y: Z, Y_0: a, X_0: g(f(g(f(a))))}

goals_solution_list: [([accumulator_reverse(g(f(g(f(a)))), f(a), Z)], {X: Z})]
    goals: [accumulator_reverse(g(f(g(f(a)))), f(a), Z)]    solution: {X: Z}
    goal: accumulator_reverse(g(f(g(f(a)))), f(a), Z)
    clause: accumulator_reverse(g(X), Y, Z) :- accumulator_reverse(X, g(Y), Z).    variable renaming: {X: X_0, Z: Z_0}
    head: accumulator_reverse(g(X_0), Y, Z_0)
    mgu: {Z: Z_0, Y: f(a), X_0: f(g(f(a)))}

goals_solution_list: [([accumulator_reverse(f(g(f(a))), g(f(a)), Z_0)], {X: Z_0})]
    goals: [accumulator_reverse(f(g(f(a))), g(f(a)), Z_0)]    solution: {X: Z_0}
    goal: accumulator_reverse(f(g(f(a))), g(f(a)), Z_0)
    clause: accumulator_reverse(f(X), Y, Z) :- accumulator_reverse(X, f(Y), Z).    variable renaming: {X: X_0}
    head: accumulator_reverse(f(X_0), Y, Z)
    mgu: {Z_0: Z, Y: g(f(a)), X_0: g(f(a))}

goals_solution_list: [([accumulator_reverse(g(f(a)), f(g(f(a))), Z)], {X: Z})]
    goals: [accumulator_reverse(g(f(a)), f(g(f(a))), Z)]    solution: {X: Z}
    goal: accumulator_reverse(g(f(a)), f(g(f(a))), Z)
    clause: accumulator_reverse(g(X), Y, Z) :- accumulator_reverse(X, g(Y), Z).    variable renaming: {X: X_0, Z: Z_0}
    head: accumulator_reverse(g(X_0), Y, Z_0)
    mgu: {Z: Z_0, Y: f(g(f(a))), X_0: f(a)}

goals_solution_list: [([accumulator_reverse(f(a), g(f(g(f(a)))), Z_0)], {X: Z_0})]
    goals: [accumulator_reverse(f(a), g(f(g(f(a)))), Z_0)]    solution: {X: Z_0}
    goal: accumulator_reverse(f(a), g(f(g(f(a)))), Z_0)
    clause: accumulator_reverse(f(X), Y, Z) :- accumulator_reverse(X, f(Y), Z).    variable renaming: {X: X_0}
    head: accumulator_reverse(f(X_0), Y, Z)
    mgu: {Z_0: Z, Y: g(f(g(f(a)))), X_0: a}

goals_solution_list: [([accumulator_reverse(a, f(g(f(g(f(a))))), Z)], {X: Z})]
    goals: [accumulator_reverse(a, f(g(f(g(f(a))))), Z)]    solution: {X: Z}
    goal: accumulator_reverse(a, f(g(f(g(f(a))))), Z)
    clause: accumulator_reverse(a, X, X).    variable renaming: {X: X_0}
    head: accumulator_reverse(a, X_0, X_0)
    mgu: {Z: X_0, X_0: f(g(f(g(f(a)))))}

goals_solution_list: [([], {X: f(g(f(g(f(a)))))})]
    goals: []    solution: {X: f(g(f(g(f(a)))))}
    YIELDED SOLUTION: {X: f(g(f(g(f(a)))))}
```