CS 427

Homework 4

Baker, Alex

# Problem 1

## Part A

1. unify((p X Y), (p a A)) return {a/X, A/Y}

2. unify((p), (p)) return {}

3. unify(p, p) return {}

4. unify((), ()) return {}

5. unify((X Y), (a A)) return {a/X, A/Y}

6. unify((X), (a)) return {a/X}

7. unify(X, a) return {a/X}

8. unify((), ())) return {}

9. unify((Y), (A)) return {A/Y}

10. unify(Y, A) return {A/Y}

11. unify((), ()) return {}

return {a/X, A/Y}

## Part B

1. unify((ancestor X (father X)), (ancestor david george)) return FAIL

2. unify((ancestor), (ancestor)) return {}

3. unify(ancestor, ancestor) return {}

4. unify((), ()) return {}

5. unify((X (father X), (david, george)) return FAIL

6. unify((X), (david)) return {david/X}

7. unify(X, david) return {david/X}

8. unify((), ()) return {}

9. unify((father david), (george)) return FAIL

10. unify((father), (george)) return FAIL

11. unify(father, george) return FAIL

return FAIL

## Part C

1. unify((q x), (not (q x))) return FAIL

2. unify((q), (not)) return FAIL

3. unify(q, not) return FAIL

return FAIL

# Problem 2

|  |  |
| --- | --- |
| 9. amount\_saved(15000) |  |
| 10. earnings(30000, steady) |  |
| 11. dependents(4) |  |
| 12. earnings(30000, steady) ∧ dependents(4) | 10 and 11 |
| 13. earnings(X, steady) ∧ dependents(Y) | unify with 12 {30000/X, 4/Y} |
| 14. earnings(30000, steady) ∧ dependents(4) ∧ ¬greater(30000, 31000) → income(inadequate) | minincome(4) = 31000  unify with 7 {30000/X, 4/Y} |
| 15. income(inadequate) | modus ponens with 10, 11, and def of greater |
| 16. amount\_saved(15000) ∧ dependents(4) | 9 and 11 |
| 17. amount\_saved(X) ∧ dependents(Y) | unify with 16 {15000/X, 4/Y} |
| 18. amount\_saved(15000) ∧ dependents(4) ∧ ¬greater(15000, 20000) → savings(inadequate) | minsavings(4) = 20000  unify with 5 {15000/X, 4/Y} |
| 19. savings(inadequate) | modus ponens with 9, 11, and def of greater |
| 20. investment(savings) | modus ponens with 19 and 1 |

# Problem 3

1. battery(dead)
   1. Battery does not have a charge
2. starter(working)
   1. The engine starter is working
3. lights(working)
   1. The lights are working
4. gas\_tank(empty)
   1. Gas tank is empty
5. engine(wont\_start)
   1. engine will not start
6. ¬battery(dead) ∧ engine(wont\_start) → ¬starter(working)
7. (engine(wont\_start) ∧ starter(working)) ∨ ¬lights(working) → battery(dead)
8. engine(wont\_start) ∧ starter(working) → gas\_tank(empty)