Stephen Chambers
smx227
October 15, 2015

Assignment 6 Writeup

1. Anything we should know about your KB and queries?
Not really. I mean, Andy Dalton actually CAN read, but the KB proves that he can't.

2. What is the time complexity of your unification implmentation?
O(n*p)

Where:
n = number of terms in predicate
p = number of predicate pairs

3. Describe any implmentation choices you made that you felt were important. Mention anything else we should know when evaluating your program.
I used ANTLR to parse the grammar. It requires the ANTLR jar to run. My make and run scripts set the classpath to .:<ANTLR jar> in compilation and execution.

4. What suggestions do you have for this assignment in the future?
This assignment took me at least 5x long to complete as any other assignment. I'm glad its broken up, and keep warning people that it will take a while and to start early.

KB And Query being solved by Assignment 7 reference:

Bill Belichek is intelligent

-bash-4.3$ ./wffs-to-cnf < football-kb.fol > output
-CanRead(x1) | Intelligent(x1)
-FootballPlayer(C1) | -CanRead(C1)
FootballPlayer(TomBrady)
-Quarterback(x1) | FootballPlayer(x1)
QuarterBack(AndyDalton)
Coach(BillBelichek)
-Coach(x1) | CanRead(x1)
-FootballPlayer(x1) | Rich(x1)
-Rich(x1) | Happy(x1)
--- negated query ---
-Intelligent(BillBelichek)

-bash-4.3$ ./fol-prove-reference < output
1: -Rich(x1) | Happy(x1)
2: -FootballPlayer(x1) | Rich(x1)
3: -Coach(x1) | CanRead(x1)
4: Coach(BILLBELICHEK)
5: QuarterBack(ANDYDALTON)
6: -Quarterback(x1) | FootballPlayer(x1)
7: FootballPlayer(TOMBRADY)
8: -FootballPlayer(C1) | -CanRead(C1)
9: -CanRead(x1) | Intelligent(x1)
10: -Intelligent(BILLBELICHEK)
10 and 9 give 11: -CanRead(BILLBELICHEK)
11 and 3 give 12: -Coach(BILLBELICHEK)
12 and 4 give 13: <empty>
3 total resolutions

Andy Dalton is a football player

-bash-4.3$ ./wffs-to-cnf < football-kb.fol > output
…
--- negated query ---
-FootballPlayer(AndyDalton)

-bash-4.3$ ./fol-prove-reference < output
1: -Rich(x1) | Happy(x1)
2: -FootballPlayer(x1) | Rich(x1)
3: -Coach(x1) | CanRead(x1)
4: Coach(BILLBELICHEK)
5: Quarterback(ANDYDALTON)
6: -Quarterback(x1) | FootballPlayer(x1)
7: FootballPlayer(TOMBRADY)
8: -FootballPlayer(C1) | -CanRead(C1)
9: -CanRead(x1) | Intelligent(x1)
10: -FootballPlayer(ANDYDALTON)
10 and 6 give 11: -Quarterback(ANDYDALTON)
11 and 5 give 12: <empty>
2 total resolutions

Andy Dalton can read (can't be proven)

```
-bash-4.3$ ./wffs-to-cnf < football-kb.fol > output
…
--- negated query ---
-CanRead(AndyDalton)

-bash-4.3$ ./fol-prove-reference < output
1: -Rich(x1) | Happy(x1)
2: -FootballPlayer(x1) | Rich(x1)
3: -Coach(x1) | CanRead(x1)
4: Coach(BILLBELICHEK)
5: Quarterback(ANDYDALTON)
6: -Quarterback(x1) | FootballPlayer(x1)
7: FootballPlayer(TOMBRADY)
8: -FootballPlayer(C1) | -CanRead(C1)
9: -CanRead(x1) | Intelligent(x1)
10: -CanRead(ANDYDALTON)
No proof exists.
2 total resolutions
```

Tom Brady is Happy

```
-bash-4.3$ ./wffs-to-cnf < football-kb.fol > output
…
--- negated query ---
-Happy(TomBrady)

-bash-4.3$ ./fol-prove-reference < output
1: -Rich(x1) | Happy(x1)
2: -FootballPlayer(x1) | Rich(x1)
3: -Coach(x1) | CanRead(x1)
4: Coach(BILLBELICHEK)
5: Quarterback(ANDYDALTON)
6: -Quarterback(x1) | FootballPlayer(x1)
7: FootballPlayer(TOMBRADY)
8: -FootballPlayer(C1) | -CanRead(C1)
9: -CanRead(x1) | Intelligent(x1)
10: -Happy(TOMBRADY)
10 and 1 give 11: -Rich(TOMBRADY)
11 and 2 give 12: -FootballPlayer(TOMBRADY)
12 and 7 give 13: <empty>
3 total resolutions
```

Validator running on sample problems:

```
********************************************************************************
./unify-validator ./run.sh < cnfs/e1.cnf >> a6-transcript
P(F(renamec1x), G(G(B)))
-P(F(renamec1x), G(G(B)))

All match!
********************************************************************************

********************************************************************************
./unify-validator ./run.sh < cnfs/e2.cnf >> a6-transcript
G(F(T), R(T), T)
-G(F(T), R(T), T)

All match!
********************************************************************************

********************************************************************************
./unify-validator ./run.sh < cnfs/e3.cnf >> a6-transcript
Bar(Val(renamec1x, BB), Val(renamec1x, BB))
-Bar(Val(renamec1x, BB), Val(renamec1x, BB))

All match!
********************************************************************************

********************************************************************************
./unify-validator ./run.sh < cnfs/e4.cnf >> a6-transcript
P(renamec2e, C, A, B, A)
-P(renamec2e, C, A, B, A)

All match!
********************************************************************************

********************************************************************************
./unify-validator ./run.sh < cnfs/e5.cnf >> a6-transcript
P(renamec2f, renamec2f, renamec2f)
-P(renamec2f, renamec2f, renamec2f)

All match!
********************************************************************************

********************************************************************************
./unify-validator ./run.sh < cnfs/e6.cnf >> a6-transcript
All match!
********************************************************************************

********************************************************************************
./unify-validator ./run.sh < cnfs/e7.cnf >> a6-transcript
P(C(A, renamec2t1), C(B, C(renamec1b0, renamec1l0)), C(A, C(B, C(B, C(A, Emp)))))
-P(C(A, renamec2t1), C(B, C(renamec1b0, renamec1l0)), C(A, C(B, C(B, C(A, Emp)))))

All match!
********************************************************************************
```

```
********************************************************************************
./unify-validator ./run.sh < cnfs/p1.cnf >> a6-transcript
-Human(Socrates) | Mortal(Socrates)
Human(Socrates)

All match!
********************************************************************************

********************************************************************************
./unify-validator ./run.sh < cnfs/p2.cnf >> a6-transcript
-Human(Socrates) | Mortal(Socrates)
Human(Socrates) | -Mortal(F(renamec2y))

-Human(F(renamec2y)) | Mortal(F(renamec2y))
Human(Socrates) | -Mortal(F(renamec2y))

All match!
********************************************************************************

********************************************************************************
./unify-validator ./run.sh < cnfs/p3.cnf >> a6-transcript
-Human(Socrates, Socrates) | Mortal(Socrates)
Human(Socrates, Socrates) | -Mortal(Socrates)

-Human(renamec2y, Socrates) | Mortal(renamec2y)
Human(renamec2y, renamec2y) | -Mortal(renamec2y)

All match!
********************************************************************************

********************************************************************************
./unify-validator ./run.sh < cnfs/p4.cnf >> a6-transcript
A(F(B), F(B))
-A(F(B), F(B))

All match!
********************************************************************************

********************************************************************************
./unify-validator ./run.sh < cnfs/p5.cnf >> a6-transcript
All match!
********************************************************************************

********************************************************************************
./unify-validator ./run.sh < cnfs/s1.cnf >> a6-transcript
-A(C, C) | B(C)
A(C, C) | -B(C)

-A(renamec2y, C) | B(renamec2y)
A(renamec2y, renamec2y) | -B(renamec2y)

All match!
********************************************************************************
```

```
**********************************************************************************
./unify-validator ./run.sh < cnfs/s2.cnf >> a6-transcript
-A(F(renamec2x), F(F(renamec2x))) | B(F(renamec2x), F(renamec2x))
A(F(renamec2x), renamec2x) | -B(F(renamec2x), F(renamec2x))

All match!
**********************************************************************************

**********************************************************************************
./unify-validator ./run.sh < cnfs/s3.cnf >> a6-transcript
A(A, A, renamec2b) | B(C(A, A, renamec2b)) | C(renamec2b)
-A(A, A, renamec2b) | -C(D)

A(renamec1a, renamec1b, D) | B(C(renamec1a, renamec1b, D)) | C(D)
-A(A, A, renamec2b) | -C(D)

All match!
**********************************************************************************

**********************************************************************************
./unify-validator ./run.sh < cnfs/test.cnf >> a6-transcript
A(A, A, renamec2b) | B(C(A, A, renamec2b)) | C(renamec2b)
-A(A, A, renamec2b) | -C(D)

A(renamec1a, renamec1b, D) | B(C(renamec1a, renamec1b, D)) | C(D)
-A(A, A, renamec2b) | -C(D)

All match!
```