Assignment 2

// DB5.java Nitish Kumar

// Closure of X under F

// Usage: java FDS F X

// F is a file that has the first line all the attributes and

// then an FD a line with a space between the left-hand side and the right-hand side

// X is a string of characters represent a set of attributes

**import** java.io.\*;

**import** java.util.\*;

**public** **class** DB5{

**class** FD{

HashSet<Character> lhs; **char** rhs;

**public** FD(HashSet<Character> l, **char** r){

lhs = l; rhs = r;

}

**public** **boolean** equals(Object obj){

FD fd2 = (FD)obj;

**return** lhs.equals(fd2.lhs) && rhs == fd2.rhs;

}

};

HashSet<Character> R = **new** HashSet<Character>(); // all attributes

HashSet<FD> F = **new** HashSet<FD>(); // the set of FDs

HashSet<Character> X = **null**; // X used in Algorithm 3.7

**public** DB5(String filename){ // 1. split FDs so each FD has a single attribute on the right

Scanner in = **null**;

**try** {

in = **new** Scanner(**new** File(filename));

} **catch** (FileNotFoundException e){

System.***err***.println(filename + " not found");

System.*exit*(1);

}

String line = in.nextLine();

**for** (**int** i = 0; i < line.length(); i++) R.add(line.charAt(i));

**while** (in.hasNextLine()){

HashSet<Character> l = **new** HashSet<Character>();

String[] terms = in.nextLine().split(" ");

**for** (**int** i = 0; i < terms[0].length(); i++) l.add(terms[0].charAt(i));

**for** (**int** i = 0; i < terms[1].length(); i++) F.add(**new** FD(l, terms[1].charAt(i)));

}

in.close();

}

HashSet<Character> string2set(String X){

HashSet<Character> Y = **new** HashSet<Character>();

**for** (**int** i = 0; i < X.length(); i++) Y.add(X.charAt(i));

**return** Y;

}

**void** printSet(Set<Character> X){

**for** (**char** c: X) System.***out***.print(c);

}

HashSet<Character> closure(HashSet<Character> Xinit){ // Algorithm 3.7

X = **new** HashSet<Character>(Xinit); // 2. initialize

**int** len = 0;

**do** { // 3. push out

len = X.size();

F.forEach(fd -> {

// your code for step 3 of Algorithm 3.7

**if** (X.containsAll(fd.lhs) && !X.contains(fd.rhs)) {

X.add(fd.rhs);

}

});

} **while** (X.size() > len);

**return** X; // 4. found closure of X

}

**boolean** followedBy(FD fd){ // fd follows from FDS

**boolean** status;

status = closure(fd.lhs).contains(fd.rhs);

**return** status;

}

**void** example39(){

System.***out***.println();

System.***out***.println(followedBy(**new** FD(string2set("AB"), 'D')));

System.***out***.println(followedBy(**new** FD(string2set("D"), 'A')));

}

**public** **static** **void** main(String[] args){

DB5 db5 = **new** DB5(args[0]);

HashSet<Character> X = db5.string2set(args[1]);

db5.printSet(db5.closure(X));

db5.example39();

}

}