CPSC-406 Report

Everett Prussak
Chapman University

February 23, 2023

Abstract

Consisting of CPSC 406 Material at Chapman University with Professor Alexander Kurz. This report will include an Introduction, Weekly Homework, and a Paper on the group project, which is done throughout the semester.

Contents

1 Introduction

This report...

2 Homework

This section contains solutions to homework.

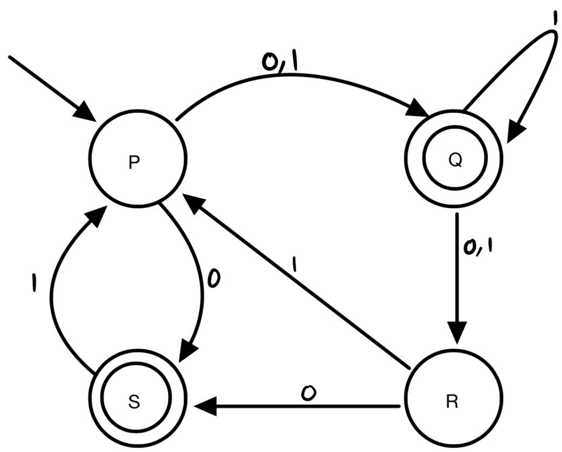
2.1 Week 2 (Homework 1)

This week's homework was to solve the following NFA:

Exercise 2.3.2: Convert to a DFA the following NFA:

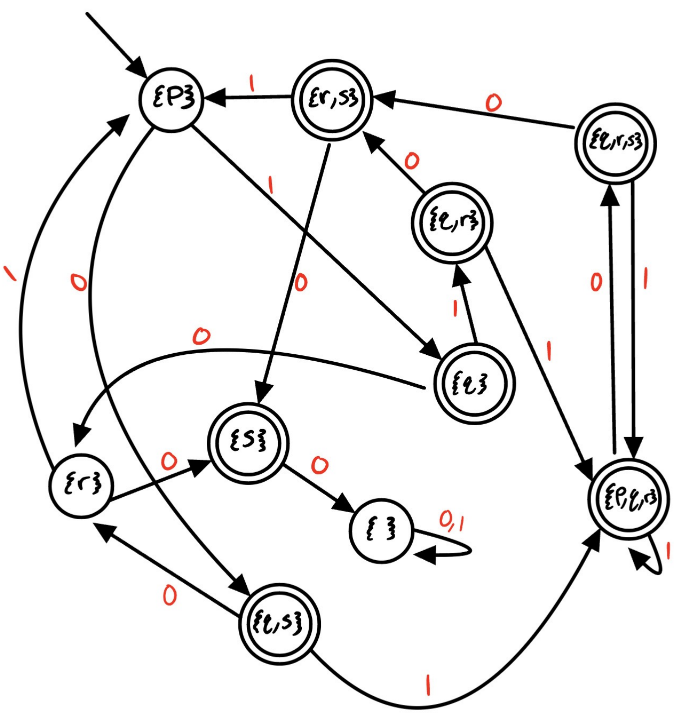
	0	1
$\rightarrow p$	$\{q, s\}$	$\{q\}$
$*q$	$\{r\}$	$\{q, r\}$
r	$\{s\}$	$\{p\}$
$*s$	$\emptyset$	$\{p\}$

This is the following NFA but drawn out:



From this NFA, the following DFA table can be made:

	0	1
→ {p}	{q, s}	{q}
*{p, s}	{r}	{p, q, r}
*{q}	{r}	{q, r}
{r}	{s}	{p}
*{p, q, r}	{q, r, s}	{p, q, r}
*{q, r}	{r, s}	{p, q, r}
*{s}	Ø	{p}
*{q, r, s}	{r, s}	{p, q, r}
*{r, s}	{s}	{p}
Ø	Ø	Ø



This DFA diagram will allow for the correct initial state, final states, and correct path for each input.

2.2 Week 3 (Homework 2)

Week 3 consisted of 2 Questions. They are in their respective sections.

2.2.1 Question 1

For Week 3 Question the object was to write the steps of the unification algorithm for each pair.

1. $f(X, f(X,Y)) \stackrel{?}{=} f(f(Y,a), f(U,b))$
2. $f(g(U), f(X,Y)) \stackrel{?}{=} f(X, f(Y,U))$
3. $h(U, f(g(V),W), g(W)) \stackrel{?}{=} h(f(X,b), U, Z)$

For number 1 of question 1 I got the following answer:

```
1. f(X,f(X,Y)) = f(f(Y,a),f(U,b))

  1. X = f(Y,a)
```

```
    o1 = [f(Y,a)/X]

2. f(X,Y) = f(U,b)
    3. X = U
        o3 = [U/X]
    4. Y = b
        o4 = [b/Y]

5. X(o3 * o4) = U, f(o3 * o4)(Y,a) = f(b,a)
    U = f(b, a)
    o5 = [f(b,a)/U]


o = o3 * o4 * o5 = [U/X, b/Y, f(b,a)/U]
X = U, Y = b, U = f(b,a)
```

Note: Sigma Symbol was not working in Verbatim, thus a simple lowercase o was subsitituted.

For number 2 of question 1 I got the following answer:

```
2. f(g(U),f(X,Y)) = f(X,f(Y,U))

  1. g(U) = X
       o1 = [g(U)/X]

  2. f(X,Y) = f(Y,U)
       3. X = Y
          o3 = [Y/X]
       4. Y = U
          o4 = [U/Y]

o = o1 * o2 * o3 = [X/U, Y/X, g(Y)/Y]
```

For number 3 of question 1 I got the following answer:

```
3. h(U,f(g(V),W),g(W)) = h(f(X,b),U,Z)

  1. U = f(X,b)
     o1 = [f(X,b)/U]

  2. f(g(V),W) = U
     o2 = [f(g(V),W)/U]

  3. g(W) = Z
     o3 = [g(W)/Z]

  4. U(o1 * o2) --> f(X,b) = f(g(V),W)

     5. X = g(V)
        o5 = [g(V)/X]
```

```
    6. b = W
       o6 = [b/W]

 7. Z(o3 * o6) = g(W), W = b
    o7 = [g(b)/Z]

 8. U(o1 * o2 * o6) --> f(X,b) = f(g(V), b)
    o8 [f(g(V),b)/U]


 o = o5 * o6 * o7 * o8 = [f(g(V),b)/U, g(V)/X, b/W, g(b)/Z]


 U = f(g(V),b), W = b, X = g(V), Z = g(b)
```

2.2.2 Question 2

For question 2, the task was to draw a SLD Recursion Tree for the following:

Question 2. Consider the following variant of the network connections problem.

```
% addr(X,Y) = X holds the address of Y
% serv(X) = X is an address server
% conn(X,Y) = X can initiate a connection to Y
% twoway(X,Y) = either end can initiate a connection

addr(a,d).
addr(a,b).
addr(b,c).
addr(c,a).

serv(b).

conn(X,Y):- addr(X,Y).
conn(X,Y):- addr(X,Z), serv(Z), addr(Z,Y).

twoway(X,Y):- conn(X,Y), conn(Y,X).
```

Draw the complete SLD-tree for this program together with the goal

```
?- twoway(W,a).
```

I got the following SLD Tree:

```
       twoway(W,a)
      /            \
     /              \
  conn(W,a)        conn(a,W)
  /        \       /        \
W=b        W=d   W=c        no
             |             / | \
            no            b  d  c
```

twoway(W,a) becomes conn(W,a) and conn(a,W) because of the rule twoway(X,Y):- conn(X,Y), conn(Y,X). This becomes the first part of the tree. Then for the conn(X,Y), it becomes conn(W,a). This side of the tree will split into W=b and W=d. W=d eventually fails because there is no serv(d). However, W=b is successful because we have addr(a,b) and a a serv(b), which allows for the conn(b,a) to be true. On the right side of the tree, we have conn(a,W). This will split into w=c and no. Since c holds the address of a, and b holds the address of c, and serv(b) exists, we can create a conn(a,c) because of these factors. We see in the equation addr(X,Z), serv(Z), addr(Z,Y) can be applied here. In our case it would look similar to addr(a,b) serv(b) addr(b,c), which allows conn(b,c) to be true.

...

3 Paper

...

4 Conclusions

(approx 400 words) A critical reflection on the content of the course. Step back from the technical details. How does the course fit into the wider world of software engineering? What did you find most interesting or useful? What improvements would you suggest?

References

[ALG] Algorithm Analysis, Chapman University, 2023.