ECE 322
Assignment 2

Arun Woosaree
XXXXXXX

October 27, 2019

1 Credit Union

Conditions				
city dweller	1	x	0	x
male	1	1	0	1
female	0	0	1	0
age $<$ 25	x	1	0	0
25 $<$ age $<$ 65	x	0	1	0
age $>$ 65	x	0	0	1

Actions				
Show Product A	1	x	x	x
Show Product B	x	1	x	x
Show Product C	x	x	1	x
Do Not Show Product D	0	0	0	1

Note: Do Not Show Product D = 0 means that Product D will be shown.

a) Maximal number of rules

Given there are 2 possibilities for gender (in this problem), 2 possibilities for city dweller, and 3 possibilities for age, the maximal number of rules is $2 \times 2 \times 3 = 12$.

b) Simplified table

The table above is already simplified, so here are the resulting test cases: (Please refer to the table above, which is the simplified table).

Test	city dweller	male	age	Expected
1	1	male	24	Show Product A
2	1	male	24	Show Product B
3	0	female	26	Show Product C
4	1	male	66	Do Not Show Product D

2

For the given subdomain, the following lines form the boundaries:

- $y = 5, 0 \leq x \leq 7$
- $x = 0, 0 \leq y \leq 5$
- $y = -x, 0 \leq x \leq 1$
- $y = x - 2, 1 \leq x \leq 7$

a) EPC Strategy

From the boundary lines, we see that the maximum value that x can have is 7, its minimum is -1, and that the maximum value that y can have is 5 while its minimum value is 0. Using the EPC testing strategy, $4^2 + 1 = 17$ test cases are expected. The extreme points chosen are $(7, 7.1, 0, -0.1)$ for x, and $(5, 5.1, 0, -0.1)$ for y. For the additional test case within the boundary, $(x = 1, y = 1)$ is chosen. The full list of suggested test cases is found below:

test id	x	y	Expected
1	7	5	In domain
2	7	5.1	Out of domain
3	7	-1	Out of domain
4	7	-0.1	Out of domain
5	7.1	5	Out of domain
6	7.1	5.1	Out of domain
7	7.1	-1	Out of domain
8	7.1	-0.1	Out of domain
9	0	5	In domain
10	0	5.1	Out of domain
11	0	-1	Out of domain
12	0	-0.1	Out of domain
13	-0.1	5	Out of domain
14	-0.1	5.1	Out of domain
15	-0.1	-1	Out of domain
16	-0.1	-0.1	Out of domain
17	1	1	In domain

b) Weak n x 1 Strategy

Given that there are 4 boundaries, we expect $4(2 + 1) + 1 = 13$ test cases. The dimensionality is 2, so 2 points are chosen on each boundary, as well as one additional point just outside of each boundary. The last test case is one point inside the boundaries. The full list of suggested test cases is found below:

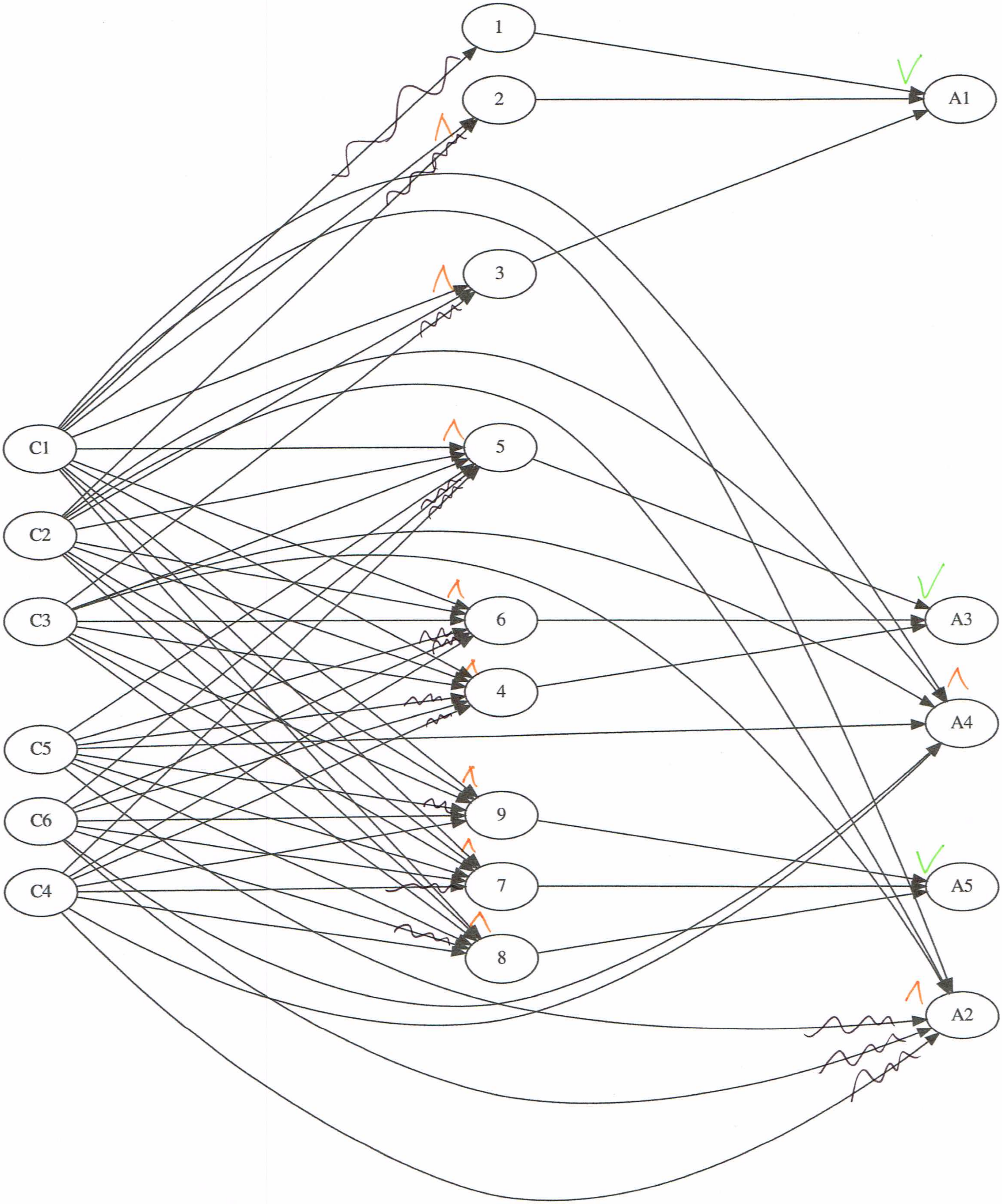
test id	description	x	y	Expected
1	on $y = 5, 0 \leq x \leq 7$ boundary	2	5	In domain
2	on $y = 5, 0 \leq x \leq 7$ boundary	4	5	In domain
3	outside $y = 5, 0 \leq x \leq 7$ boundary	3	5.1	Outside of domain
4	on $x = 0, 0 \leq y \leq 5$ boundary	0	2	In domain
5	on $x = 0, 0 \leq y \leq 5$ boundary	0	4	In domain
6	outside $x = 0, 0 \leq y \leq 5$ boundary	-0.1	3	Outside of domain
7	on $y = -x, 0 \leq x \leq 1$ boundary	0.3	-0.3	In domain
8	on $y = -x, 0 \leq x \leq 1$ boundary	0.7	-0.7	In domain
9	outside $y = -x, 0 \leq x \leq 1$ boundary	0.5	-0.6	Outside of domain
10	on $y = x - 2, 1 \leq x \leq 7$ boundary	3	1	In domain
11	on $y = x - 2, 1 \leq x \leq 7$ boundary	5	3	In domain
12	outside $y = x - 2, 1 \leq x \leq 7$ boundary	4	1.9	Outside of domain
13	Inside the boundaries	1	1	In domain

3 Cause-Effect Graph

From the following decision table, the cause effect graph below is generated: The cause effect graph can be simplified by choosing the intermediate nodes differently.

Conditions											
C1: $a < b + c$?	0	1	1	1	1	1	1	1	1	1	1
C2: $b < a + c$?	x	0	1	1	1	1	1	1	1	1	1
C3: $c < a + b$?	x	x	0	1	1	1	1	1	1	1	1
C4: $a = b$?	x	x	x	1	1	1	1	0	0	0	0
C5: $a = c$?	x	x	x	1	1	0	0	1	1	0	0
C6: $b = c$?	x	x	x	1	0	1	0	1	0	1	0

Actions											
A1: Not a Triangle	1	1	1	x	x	x	x	x	x	x	x
A2: Scalene	x	x	x	x	x	x	x	x	x	x	1
A3: Isosceles	x	x	x	x	x	x	1	x	1	1	x
A3: Equilateral	x	x	x	1	x	x	x	x	x	x	x
A4: Impossible	x	x	x	x	1	1	x	1	x	x	x


1
2
3
5
6
4
9
7
8
A1
A3
A4
A5
A2
C1
C2
C3
C5
C6
C4

4 Test Cases

The following description table is derived from the cause effect graph (Because of the requires, $(C3 = 1, C6 = 0)$ will never happen)

Conditions					
C3	1	0	0	x	0
C4	x	x	1	x	0
C5	1	x	1	0	x
C6	1	0	1	x	x
Effects					
E	1	0	1	0	0

From the decision table above, the following test cases are generated:

Test	C3	C4	C5	C6	Expected
1	1	0	1	1	1
2	0	1	1	0	0
3	0	1	1	1	1
4	1	1	0	1	0
5	0	0	1	1	0

5 Combinatorial Testing

There are $2 \times 3 \times 3 \times 3 \times 3 \times 2 \times 3 \times 3 = 2916$ total possible combinations to test. Ideally, the orthogonal array should be of size $2^2 3^6$. Using some code, the following mapping is created:

	PRINTERS	PLUGINS	BROWSERS	OPERATING SYSTEMS	SERVERS	MONITORS	EMAIL SYSTEMS	SOFTWARE PACKAGES
1	printer2	plugin2	browser3	os1	server2	monitor2	email1	software2
2	printer1	plugin1	browser1	os3	server1	monitor1	email2	software2
3	printer1	plugin2	browser1	os2	server3	monitor2	email3	software1
4	printer2	plugin1	browser2	os2	server2	monitor1	email1	software3
5	printer2	plugin2	browser2	os3	server3	monitor2	email2	software3
6	printer1	plugin1	browser3	os1	server1	monitor1	email3	software3
7	printer2	plugin2	browser3	os2	server1	monitor1	email2	software1
8	printer1	plugin1	browser2	os3	server2	monitor2	email3	software1
9	printer1	plugin1	browser3	os3	server3	monitor1	email1	software2
10	printer2	plugin1	browser1	os1	server1	monitor2	email1	software1
11	printer2	plugin1	browser2	os2	server1	monitor2	email3	software2
12	printer1	plugin1	browser2	os1	server3	monitor1	email2	software3
13	printer1	plugin2	browser1	os2	server2	monitor2	email2	software3

resulting in 13 test cases, as opposed to 2916 if we were to test all possible combinations, a huge improvement ($\frac{2916-13}{2916} \times 100 \approx 99.55\%$)

Alternatively, using a standard orthogonal array, the closest fit is $L_{18}(2^1 3^7)$, which would result in 18 test cases and still a huge improvement over testing all possible combinations ($\approx 99.38\%$)