Assignment 1

Qianlang Chen

Mon 01/13/2020

Section 1.2

Exercise 1.m (p21)

(todo)

Exercise 7.g (p21)

First five terms:

- $a_1 = 2^{2\times1-1} - 1 = 1$
- $a_2 = 2^{2\times2-1} - 1 = 7$
- $a_3 = 2^{2\times3-1} - 1 = 31$
- $a_4 = 2^{2\times4-1} - 1 = 127$
- $a_5 = 2^{2\times5-1} - 1 = 511$

Recursive formula:

$$a_1 = 1; a_n = 4 \times a_{n-1} + 3$$

Exercise 9.a (p21)

First five terms:

- $a_1 = 5$
- $a_2 = a_{2-1} + (2 + 4) = 11$
- $a_3 = a_{3-1} + (3 + 4) = 18$
- $a_4 = a_{4-1} + (4 + 4) = 26$
- $a_5 = a_{5-1} + (5 + 4) = 35$

Closed formula:

$$a_n = \frac{n^2 + n}{2} + 4n$$

Exercise 30 (p23)

Part (a)

- $a_1 = A$
- $a_2 = a_{2/2}B = a_1B = AB$

- $a_3 = a_{(3-1)/2}A = a_1A = AA$
- $a_4 = a_{4/2}B = a_2B = ABB$
- $a_5 = a_{(5-1)/2}A = a_2A = ABA$
- $a_6 = a_{6/2}B = a_3B = AAB$
- $a_7 = a_{(7-1)/2}A = a_3A = AAA$
- $a_8 = a_{8/2}B = a_4B = ABBB$
- $a_9 = a_{(9-1)/2}A = a_4A = ABBA$
- $a_{10} = a_{10/2}B = a_5B = ABAB$

Part (b)

- $a_{17} = a_{(17-1)/2}A = a_8A = ABBBA$
- $a_{21} = a_{(21-1)/2}A = a_{10}A = ABABA$

Part (c)

- $a_{630} = a_{630/2}B = a_{315}B = ABBAAABAAB$
- $a_{631} = a_{(631-1)/2}A = a_{315}A = ABBAAABAAA$

Section 1.3

Exercise 1.a (p37)

Let p represent "A is telling the truth" and q represent "B is telling the truth," according to the information given by the problem, we can generate the following truth table:

p	q	$p \wedge q$	$\neg p$
T	T	T	F
T	F	T	T
F	T	F	T
F	F	F	T

As seen from the truth table, only the third row provides a consistent result (where when $p = F$ and $q = T$, $p = p \wedge q$ and $q = \neg p$). Therefore, A is lying, and B is telling the truth.

Exercise 4.a (p37)

$$\neg p \wedge (q \vee r)$$

where p, q, and r represent A, B, and C being truthful, respectively.

Exercise 10.d (p38)

$$(x \leq 0) \wedge (y \leq 0)$$

Exercise 24.b (p40)

$$(p \vee t) \wedge (p \vee c) \equiv t \wedge (p \vee c) \quad \text{Universal bound} \equiv t \wedge p \quad \text{Identity} \equiv p \wedge t \quad \text{Commutative} \equiv p \quad \text{Identity} \tag{1}$$

Problem 7 (Java method)

Since `charIsNotLowerCaseLetter(c)` always produces exactly the opposite result of `charIsLowerCaseLetter(c)` and the definition of `charIsLowerCaseLetter(c)` is `'a' <= c && c <= 'z'`, according to DeMorgan's Law,

```
!('a' <= c && c <= 'z')
=== !('a' <= c) || !(c <= 'z')
=== 'a' > c || c > 'z'
```

Section 1.4

Exercise 14.c (p52)

Exercise 15.c (p52)

Exercise 17.c (p52)

Exercise 20.b (p52)

Section 1.5

Exercise 11.c (p66)

Exercise 11.e (p66)

Exercise 14.d (p67)

Exercise 17.a (p67)

Exercise 25.e (p67)

Exercise 26.e (p67)

Exercise 27.e (p67)