Hsuan Yu Liu 823327369

Part 1

Question 1: Show what happens to the variable domains when forward-checking is employed after each of the first two decisions:

a. CA is assigned B.

b. AZ is assigned O.

	CA	NV	UT	AZ	CO	NM	TX
Initial Domains	R B Y O	R B Y O	R B Y O	R B Y O	R B Y O	R B Y O	R B Y O
CA is assigned B	B	R Y O	R B Y O	R Y O	R B Y O	R B Y O	R B Y O
AZ is assigned O	B	R Y	R B Y	O	R B Y	R B Y	R B Y O

a. NV and AZ remove 'B' in their domains.

b. NV, UT, and NM remove 'O' in their domains

Question 2: Using maintaining arc consistency instead of forward checking.

	CA	NV	UT	AZ	CO	NM	TX
Initial Domains	R B Y O	R B Y O	R B Y O	R B Y O	R B Y O	R B Y O	R B Y O
CA is assigned B	B	R Y O	R B Y O	R Y O	R B Y O	R B Y O	R B Y O
AZ is assigned O	B	R Y	R B Y	O	R B Y	R B Y	R B Y O

a. Check (CA, NV)→(CA, AZ)→(AZ, NM) → (AZ, CO) → (AZ, UT) → (AZ, NV) → (NV, UT) → (NV, AZ): NV and AZ remove 'B' in their domains.

b. Check (AZ, NV)→(AZ, UT) →(AZ, CO)→(AZ, NM)→ (NV, UT) → (UT, CO) →(CO, NM) → (NM, CO)→(NM,TX): NV, UT, CO, and NM remove 'O' in their domains

---

Question 3: Show how the conflict sets would be updated for these two moves if we were using conflict-directed backjumping.

{}← Conflict set ; ◯← Assigned

Conflict set	CA	NV	UT	AZ	CO	NM	TX
Initial set							
CA is assigned B	B	{ CA= B }		{ CA= B }			
AZ is assigned O	B	{ CA= B, AZ=O }	{ AZ=O }	O	{ AZ=O }	{ AZ=O }	

Question 4: Suppose students have the following set of simplified constraints:

i. Students cannot be enrolled in overlapping courses.
ii. Students cannot enroll in classes on days that they work.
iii. A student may not enroll in a course that does not have space.

a. Write a specification for this constraint satisfaction problem.

Variables:
Divide a student's timetable into four parts.

1. MW 14:00~15:15 → S_iMW1 (A period, 14:00 to 15:15, of Student i on Monday and Wednesday)

2. MW 15:00~16:15 → S_iMW2 (A period, 15:00 to 16:15, of Student i on Monday and Wednesday)

3. TTh 14:00~15:30 → S_iTTh2 (A period, 14:00 to 15:30, of Student i on Tuesday and Thursday)

4. TTh 18:00~19:15 → S_iTTh2 (A period, 18:00 to 19:15, of Student i on Tuesday and Thursday)

One student has 4 periods, so we have i*4 variables.

For example,
Variables= { S_1MW1, S_1MW2, S_1TTh1, S_1TTh2, S_2MW1, S_2MW2, S_2TTh1, S_2TTh2, …}

Domain:
Four Courses, C_1, C_2, C_3 and C_4. One Empty, E, and a Work, W.

C_1: A course whose Tutorial is 1

C_2: A course whose Tutorial is 2

E: The period is free.

W: Students work either days.

C_3: A course whose Tutorial is 3

C_4: A course whose Tutorial is 4

For example,

S_1MW1 = W means S_1 works either Monday or Wednesday.

S_2MW1 = C_1 means S_2 take a course C_1.

Constraints:

1. MW1 and MW2 cannot take course simultaneously. If MW1 = C_1, the other must be E.

2. TTh1 and TTh2 cannot take course simultaneously. If TTh1 = C_1, the other must be E

3. If MW_x is W then the other must be W as well. → MW1 = W ↔MW2 = W

4. If TTh_x is W then the other must be W as well. → TTh1 = W ↔ TTh2 = W

5. The number of C_x cannot be more than 30. E.g. Cannot be more than 30 students taking C_1.

b. Show how encapsulation could be used to create a binarize the constraint between multiple students the enrollment limit.

i. Students cannot be enrolled in overlapping courses. & ii. Students cannot enroll in classes on days that they work.
Combine MW1 and WM2 into a set and TTh1 and TTh2 into a set. The elements in a set cannot be two C_i or more. Also, if an element is W, the other must be W too.
E.g. $U_{i,ii}$ = [{ S_1MW1= C1, S_1MW2= C2 }, { S_1MW1= C1, S_1MW2= E }, { S_1TTh1= C3, S_1TTh2= W }, { S_1TTh1= E, S_1TTh2= C4 }, …]
{ S_1MW1= C1, S_1MW2= C2 } = False
{ S_1MW1= C1, S_1MW2= E } = True
{ S_1TTh1= C3, S_1TTh2= W } = False
{ S_1TTh1= W, S_1TTh2= W } = True
{ S_1TTh1= E, S_1TTh2= W } = False
{ S_1TTh1= E, S_1TTh2= C4 } = True

iii. A student may not enroll in a course that does not have space:

We can combine all S_xMW1 to be a set and count how many C1. If the number of C1 is larger than 30 then it conflicts. Following this rule, we can combine different MW and TTh to be sets respectively.

E.g. U_{iii} = [{S_1 = C1, S_2 =C1, S_3 = C1, S_4 = E, S_5 =E …, S_i =W…, S_j =C1…}, {S_1 = C2, S_2 =C2, S_3 = C2, S_4 = E, S_5 =E …, S_i =W…, S_j =C2…}, {S_1 = C3, S_2 =C3, S_3 = C3, S_4 = E, S_5 =E …, S_i =W…, S_j =C3…}, {S_1 = C4, S_2 =C4, S_3 = C4, S_4 = E, S_5 =E …, S_i =W…, S_j =C4…}].

If {S_1 = C1, S_2 =C1, S_3 = C1, S_4 = E, S_5 =E …, S_i =W…, S_j =C1…} contains more than 30 C1, then it is failure.

If {S_1 = C2, S_2 =C2, S_3 = C2, S_4 = E, S_5 =E …, S_i =W…, S_j =C2…} contains more than 30 C2, then it is failure.

If {S_1 = C3, S_2 =C3, S_3 = C3, S_4 = E, S_5 =E …, S_i =W…, S_j =C3…} contains more than 30 C3, then it is failure.