Ricky Ma 82943424
Bruce Cui 13412151
Late days claimed: 0

Question 1: Inference in Markov Networks

a. Computing P(A|b1) using variable elimination

i. Factors are f0(D,A), f1(A,B), f2(B,C), f3(C,D)

f0		
d0	a0	100
d0	a1	1
d1	a0	1
d1	a1	100
f1		
a0	b0	30
a0	b1	5
a1	b0	1
a1	b1	10
f2		
b0	c0	100
b0	c1	1
b1	c0	1
b1	c1	100
f3		
c0	d0	1
c0	d1	100
c1	d0	100
c1	d1	1

ii. Then we update the observed value B=b1, so f1 and f2 changes to f4 and f5 as follows:

f4	
a0	5
a1	10
f5	
c0	1
c1	100

iii. Now, specify elimination ordering D,C:

$$\sum_D f_0(D,A) f_3(C,D) \sum_C f_4(A) f_5(C)$$

iv. Combine f0 and f3 to form f6(D,A,C):

f6			
d0	a0	c0	100
d0	a0	c1	10000
d0	a1	c0	1
d0	a1	c1	100
d1	a0	c0	100
d1	a0	c1	1
d1	a1	c0	10000
d1	a1	c1	100

v. Sum out D to form f7(A,C):

f7		
a0	c0	200
a0	c1	10001
a1	c0	10001
a1	c1	200

vi. Eliminate C by combining f5(C) and f7(A,C), then summing out on C:

f8		
a0	c0	200
a0	c1	1000100
a1	c0	10001
a1	c1	20000

f9	
a0	1000300
a1	30001

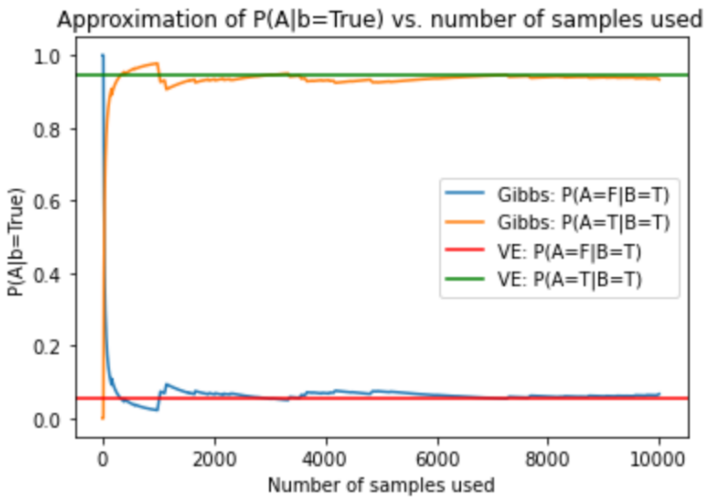
vii. Combine f4(A) and f9(A) to form f10(A):

f10	
a0	5001500
a1	300010

viii. Normalize f10(A) to get P(A|b1):

a0	0.94341
a1	0.05659

b.




```python
def prob_ab(A, B):
    if not A and not B:
        return 30
    elif not A and B:
        return 5
    elif A and not B:
        return 1
    else:
        return 10

def prob_bc(B, C):
    if not B and not C:
        return 100
    elif not B and C:
        return 1
    elif B and not C:
        return 1
    else:
        return 100

def prob_cd(C, D):
    if not C and not D:
        return 1
    elif not C and D:
        return 100
    elif C and not D:
        return 100
    else:
        return 1

def prob_da(D, A):
    if not D and not A:
        return 100
    elif not D and A:
        return 1
    elif D and not A:
        return 1
    else:
        return 100
```

```python
def random_bool(v1, v2):
    p = v1 / (v1 + v2)
    return random.uniform(0, 1) <= p

def next_variable(curr_variable):
    variables = ['A', 'C', 'D']
    return random.choice(variables)

def sample_variable(curr_variable, A, B, C, D):
    if curr_variable == 'A':
        A0 = prob_ab(0, B) * prob_da(D, 0)
        A1 = prob_ab(1, B) * prob_da(D, 1)
        A = random_bool(A1, A0)

    elif curr_variable == 'D':
        D0 = prob_cd(C, 0) * prob_da(0, A)
        D1 = prob_cd(C, 1) * prob_da(1, A)
        D = random_bool(D1, D0)

    elif curr_variable == 'C':
        C0 = prob_cd(0, D) * prob_bc(B, 0)
        C1 = prob_cd(1, D) * prob_bc(B, 1)
        C = random_bool(C1, C0)

    return A, B, C, D
```

```python
A1_count = 0
A0_count = 0
num_samples = 10000
curr_variable = 'A'
(A, B, C, D) = (True, True, True, True)

pa0_givenb1 = []
pa1_givenb1 = []
for i in range(1,num_samples+1):
    (A, B, C, D) = sample_variable(curr_variable, A, B, C, D)
    if A:
        A1_count += 1
    else:
        A0_count += 1
    curr_variable = next_variable(curr_variable)
    pa0_givenb1.append(A0_count / i)
    pa1_givenb1.append(A1_count / i)
```

Question 2: Resolution in Propositional Logics

<u>KB</u>

$(B \lor C) \land \neg A$

$\neg(B \rightarrow A)$

$A \rightarrow (\neg B \lor C)$

$A \lor E$

C

a. $KB \Vdash \neg(A \leftrightarrow C)$
   i. Convert second and third clauses to CNF
      $\neg(B \rightarrow A) \equiv \neg(\neg B \lor A) \equiv \neg A \land B$
      $A \rightarrow (\neg B \lor C) \equiv \neg A \lor \neg B \lor C$
   ii. Simplify KB to CNF
      - Conjunction of second and first clause:
         - $[\neg A \land B] \land [(B \lor C) \land \neg A] \equiv \neg A \land B$
      - Conjunction of result from above with fourth clause:
         - $[\neg A \land B] \land [A \lor E] \equiv \neg A \land B \land E$
      - Conjunction of result from above with fifth clause:
         - $[\neg A \land B \land E] \land C \equiv \neg A \land B \land E \land C$
      - Conjunction of result from above with third clause:
         - $[\neg A \land B \land E \land C] \land [\neg A \lor \neg B \lor C] \equiv \neg A \land B \land E \land C$
      - Conjunction of all clauses in $KB \equiv \neg A \land B \land E \land C$
   iii. Apply resolution algorithm, where $KB \equiv \neg A \land B \land E \land C$, $\alpha \equiv \neg(A \leftrightarrow C)$
      - Simplify alpha: $(A \leftrightarrow C) \equiv (A \rightarrow C) \land (C \rightarrow A) \equiv (\neg A \lor C) \land (\neg C \lor A)$
      - New = [ ]
      - Clauses = $CNF(KB \land \alpha) \equiv \neg A \land B \land E \land C \land (\neg A \lor C) \land (\neg C \lor A)$
         - Resolvents = $\neg A \land (\neg C \lor A) \equiv \neg A \land \neg C$
            i. New = new ∪ resolvents = $\neg A \land \neg C$
         - Resolvents = $C \land (\neg C \lor A) \equiv A \land C$
            i. New = new ∪ resolvents = $\neg A \land \neg C \land A \land C$
         - Remaining resolvents are redundant, and are left out for brevity
      - New clauses (A and C) were created, so starting the second loop, clauses = clauses ∪ new
      - Clauses = $\neg A \land B \land E \land C \land (\neg A \lor C) \land (\neg C \lor A) \land \neg C \land A$
         - Resolvents = $\neg A \land A \equiv \emptyset$, return true
         - It can also be shown that: resolvents = $\neg C \land C \equiv \emptyset$, in which we also return true
   iv. The algorithm returns true. Therefore the query is entailed, i.e. $KB \Vdash \neg(A \leftrightarrow C)$.

b. Since KB entails $\neg(A \leftrightarrow C)$, no interpretation exists in which KB is true and $\neg(A \leftrightarrow C)$ is false.

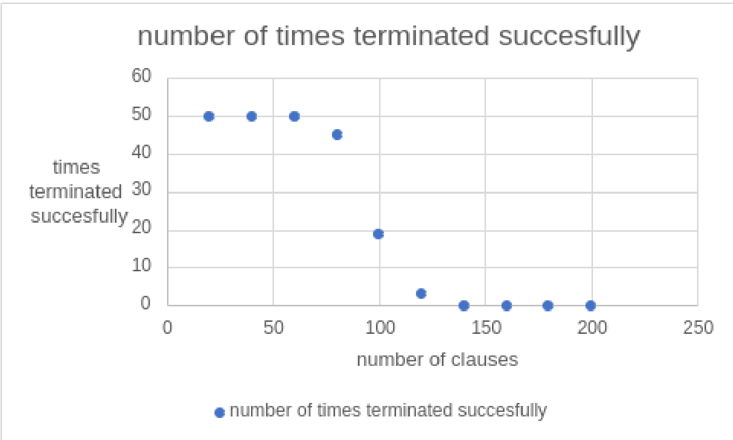
Question 3: WalkSAT

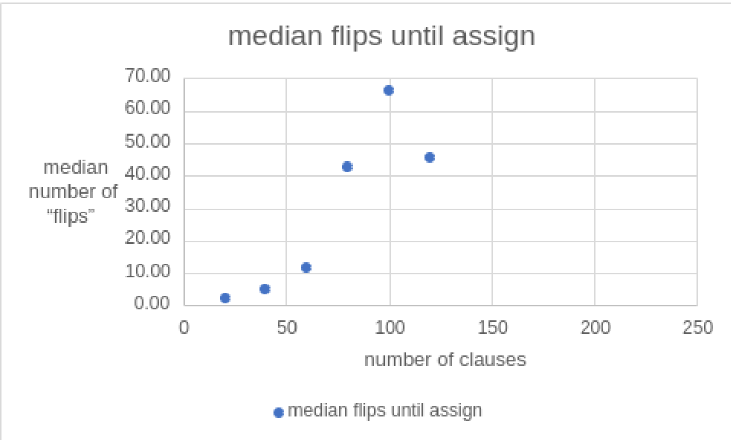
Used: https://www.cs.rochester.edu/u/kautz/walksat/

How many times WalkSAT terminates successfully for each of these sets of 50 problems
The median number of "flips" (i.e. steps) for each of these sets of 50 problems, considering only the cases when WalkSAT terminates successfully

The table and figures below are determined by fixing N (number of variables) at 20 and generating 50 random 3SAT problems for each value of C/N from 1 to 10 (so, with number of clauses ranging from 20, 40, …, 200).

Number of clauses	Median number of flips until assigned	Number of times terminated successfully
20	2.25	50
40	5.15	50
60	11.60	50
80	42.80	45
100	66.40	19
120	45.30	3
140		0
160		0
180		0
200		0





Source code: https://github.com/brucecui97/cpsc422A3Q3

Question 4: Semantics of First-Order Logic

Interpretations for a first-order logic specification that includes:
10 constant symbols, 2 ternary predicates, 2 binary predicates, 10 unary predicates
Assume the world contains 10 objects and they are all assigned unique names.

10 objects and 10 symbols:
- Assuming unique names, each object is assigned 1 symbol, so there are 10! interpretations

10 unary predicates:
- 10 objects, each can be 0 or 1, so there are 2^10 interpretations for 1 predicate
- With 10 predicates, we have (2^10)^10 interpretations

2 binary predicates:
- 10*10 = 100 possibilities, each one can be 0 or 1, so there are 2^100 interpretations for 1 predicate
- With 2 predicates, we have (2^100)^2 interpretations

2 ternary predicates:
- 10*10*10 = 1000 possibilities, each one can be 0 or 1, so there are 2^1000 interpretations for 1 predicate
- With 2 predicates, we have (2^1000)^2 interpretations

TOTAL:
$10! * 2^{10^{10}} * 2^{100^{2}} * 2^{1000^{2}}$ interpretations