CS3342 – Assignment 4
due Apr. 6, 2023
2-day no-penalty extension until: Apr. 8, 11:55pm

1. (20pt) Consider the following statement: If X is smart and hard working, then X should do graduate studies, but, if X is smart but not hard working, then X should not do graduate studies.

   (a) (10pt) Express the above statement as a conjunction of Horn clauses (see slide 8, ch.12.1): Recall that Horn clauses have a single non-negated term.

   (b) (10pt) Write the above Horn clauses in Prolog.

2. (30pt) Consider the following Prolog facts and rules:

```
insert(X, L, [X|L]).
insert(X, [H|L], [H|L1]) :- insert(X, L, L1).
permute([], []).
permute([H|T], P) :- permute(T, P1), insert(H, P1, P).
```

   Draw the Prolog trees corresponding to the following queries; show the rule and substitution used at each step:

   (a) (10pt)

```
?- permute([a,b], Y).
Y = [a, b] ;
Y = [b, a] ;
false.
```

   (b) (10pt) Show the part of the tree with solutions and explain why the tree is infinite:

```
?- permute(Y, [a,b]).
Y = [a, b] ;
Y = [b, a] ;
...
```

   (c) (10pt) Explain the effect of the cut in the tree.

```
?- not(permute(Y, [])).
false.
```

3. (50pt) Write a Prolog program `magic_square.pl` which computes all 3×3 magic squares. (An $n \times n$ magic square contains the numbers $1, 2, \ldots, n^2$ such that the sum on all rows, columns and diagonals is the same.) The program uses brute force search (but gives instantaneous answers), behaving as follows (the user inputs ';' repeatedly in order to force the search for more solutions):

```
?- magic_square(S).
S = [2, 7, 6, 9, 5, 1, 4, 3, 8] ;
S = [2, 9, 4, 7, 5, 3, 6, 1, 8] ;
S = [4, 3, 8, 9, 5, 1, 2, 7, 6] ;
S = [4, 9, 2, 3, 5, 7, 8, 1, 6] ;
S = [6, 1, 8, 7, 5, 3, 2, 9, 4] ;
S = [6, 7, 2, 1, 5, 9, 8, 3, 4] ;
S = [8, 1, 6, 3, 5, 7, 4, 9, 2] ;
S = [8, 3, 4, 1, 5, 9, 6, 7, 2] ;
false.
```

READ ME! Submit source code, if required, as separate file(s).

Submit your answers as a single pdf file in OWL. Solutions should be typed; readable (by others!) hand-written solutions are also acceptable.

External tools: You are allowed to use any external tools, such as JFLAP, ChatGPT, lambda expression calculators, etc., to help you solve the assignments. Make sure you understand the solutions as no tools will be available during the exams!

LaTeX: For those interested, the best (the only!) program for scientific writing is LaTeX. It is free and you can start using it in minutes: `https://tobi.oetiker.ch/lshort/lshort.pdf`

1. (20pt) Consider the following statement: If X is smart and hard working, then X should do graduate studies, but, if X is smart but not hard working, then X should not do graduate studies.

   (a) (10pt) Express the above statement as a conjunction of Horn clauses (see slide 8, ch.12.1): Recall that Horn clauses have a single non-negated term.

   (b) (10pt) Write the above Horn clauses in Prolog.

(a) $S(x)$: x is smart

$H(x)$: x is hard working

$G(x)$: x should do graduate study

$(S(x) \wedge H(x) \rightarrow G(x)) \wedge ((S(x) \rightarrow H(x)) \vee (S(x) \rightarrow G(x)))$

(b) G :- S, H

H :- S

G :- S

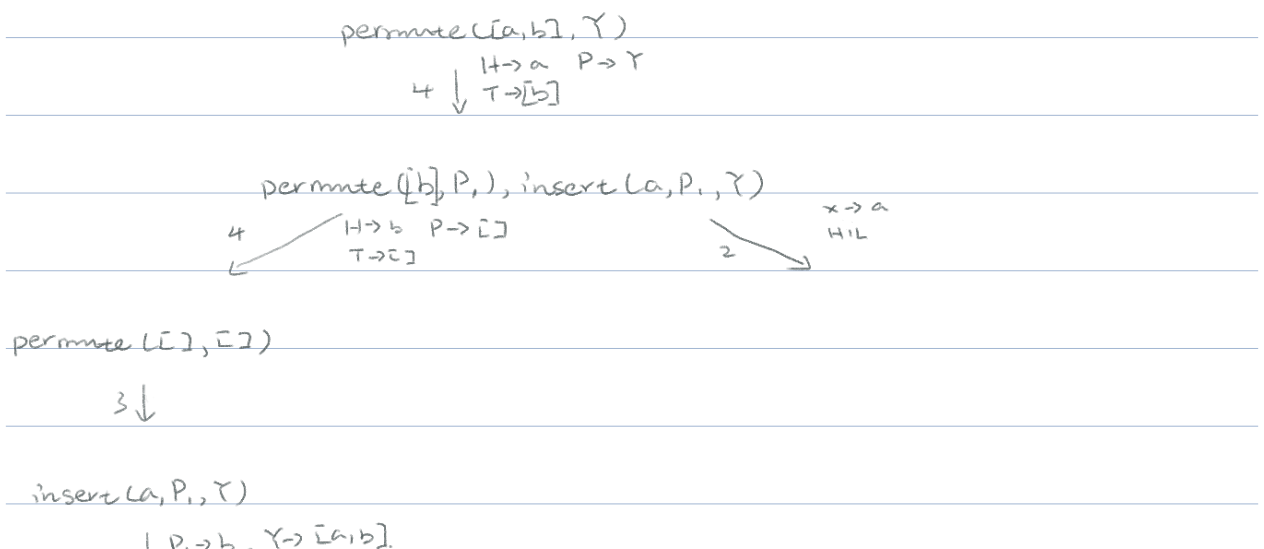
2. (30pt) Consider the following Prolog facts and rules:

```
insert(X, L, [X|L]).
insert(X, [H|L], [H|L1]) :- insert(X, L, L1).
permute([], []).
permute([H|T], P) :- permute(T, P1), insert(H, P1, P).
```

Draw the Prolog trees corresponding to the following queries; show the rule and substitution used at each step:

(a) (10pt)

```
?- permute([a,b], Y).
Y = [a, b] ;
Y = [b, a] ;
false.
```



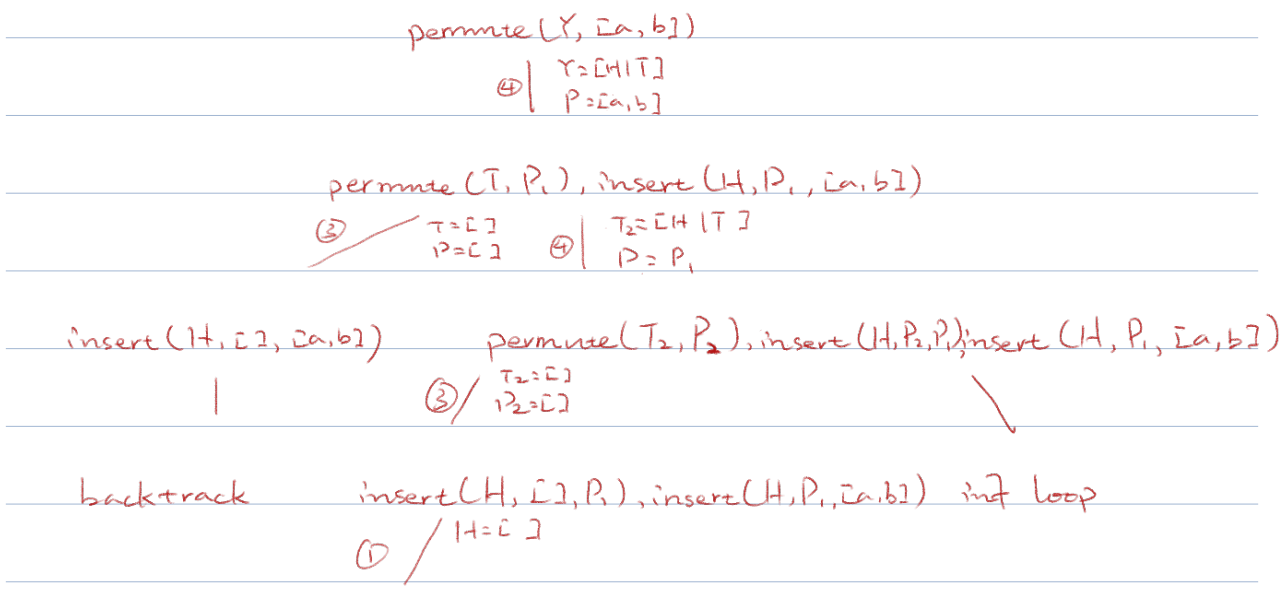
2. (30pt) Consider the following Prolog facts and rules:

```
1) insert(X, L, [X|L]).
2) insert(X, [H|L], [H|L1]) :- insert(X, L, L1).
3) permute([], []).
4) permute([H|T], P) :- permute(T, P1), insert(H, P1, P).
```

Draw the Prolog trees corresponding to the following queries; show the rule and substitution used at each step:

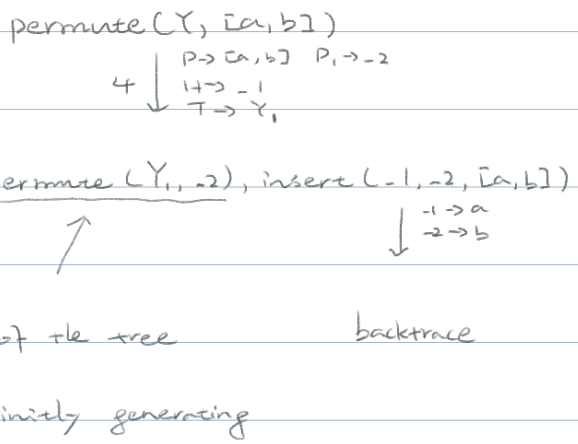
(b) (10pt) Show the part of the tree with solutions and explain why the tree is infinite:

```
?- permute(Y, [a,b]).
Y = [a, b] ;
Y = [b, a] ;
```



(b) (10pt) Show the part of the tree with solutions and explain why the tree is infinite:

```
?- permute(Y, [a,b]).
Y = [a, b] ;
Y = [b, a] ;
```



(c) (10pt) Explain the effect of the cut in the tree.

```
?- not(permute(Y, [])).
false.
```

not(permute(Y, [])) would be permute(Y, []), !, fail and if the permutation success, it would cut off and prevent backtracking