

Homework Assignment 4

October 3, 2024

Lucas Miguel Tassis

Exercise 1

We can write the requirements for the n -Queens problem as a propositional wff ψ_n as:

$$\psi_n = \psi_n^{\text{row}} \wedge \psi_n^{\text{col}} \wedge \psi_n^{\text{diag1}} \wedge \psi_n^{\text{diag2}}$$

where,

$$\psi_n^{\text{row}} = \bigwedge_{i=1}^n \bigvee_{j=1}^n \left(q_{i,j} \wedge \bigwedge_{k \neq j}^n \neg q_{i,k} \right)$$

In ψ_n^{row} , the $\bigwedge_{i=1}^n$ guarantees that every row will have a queen. The $\bigwedge_{k \neq j}^n$ add the negation of all other elements in the row, except $q_{i,j}$, therefore, only $q_{i,j}$ can be true for the expression to be true (and thus, eliminating the \vee). Finally, the $\bigvee_{j=1}^n$ guarantees that at least one of the expressions inside will have to be true. Therefore, if the expression is true, there is a queen, and only one, in each row.

$$\psi_n^{\text{col}} = \bigwedge_{j=1}^n \bigvee_{i=1}^n \left(q_{i,j} \wedge \bigwedge_{k \neq j}^n \neg q_{k,j} \right)$$

In ψ_n^{col} , the argument is analogous to the ψ_n^{row} . Only this time, we add the negation of all other elements in the column. Therefore, if the expression is true, there is a queen, and only one, in each column.

For the wffs ψ_n^{diag1} and ψ_n^{diag2} , I used the solution on the book [Discrete Mathematics and Its Applications](#) as base. I was not able to find a better solution than the one used in the book for the diagonal restrictions. The expressions were the following:

$$\psi_n^{\text{diag1}} = \bigwedge_{i=1}^{n-1} \bigwedge_{j=1}^{n-1} \bigwedge_{k=1}^{\min(n-i, n-j)} (\neg q_{i,j} \vee \neg q_{i+k, j+k})$$

The idea behind this expression is for each pair i, j defined by the $\bigwedge_{i=1}^{n-1}$ (covers all the rows) and

the $\bigwedge_{j=1}^{n-1}$ (covers all the columns), we check if there is a queen on the diagonal of q_{ij} . If there are two queens in a diagonal, the whole expression will be false. Notice that the upper limit is $n-1$ because the last element of each row/column do not have a diagonal, therefore, there is no need to check it. The way to cover the diagonal given a pair i, j is to use the fact that $i_1 - j_1 = i_2 - j_2$ is true for the diagonal values. Therefore, we have that $i - j = (i + k) - (j + k)$, and that we can walk along the diagonal of a pair i, j by adding k in each row/column index. To control how big this k can be considering n , the upper bound of k is defined as $\min(n-i, n-j)$. The upper bound avoids indices going beyond n on the columns and rows. Notice that with the expression ψ_n^{diag1} ,

for each pair i, j on the board, we check all the values in their diagonal to verify if there are two queens. Therefore, if the expression ψ_n^{diag1} is true, there is a queen, and only one, in each diagonal.

$$\psi_n^{\text{diag2}} = \bigwedge_{i=2}^n \bigwedge_{j=1}^{n-1} \bigwedge_{k=1}^{\min(i-1, n-j)} (\neg q_{i,j} \vee \neg q_{i-k, j+k})$$

The idea behind the antidiagonal expression is similar to the one for the diagonal expression. We only have to change the limits and also the way we calculate the antidiagonal of a pair i, j . First, the limit changes on $\bigwedge_{i=2}^n$ because at $i = 1$ there is no antidiagonal. For j the limit is similar. The way to cover the diagonal given a pair i, j , is to use the fact that $i_1 + j_1 = i_2 + j_2$ is true for the antidiagonal values. Therefore, we have that $i + j = (i - k) + (j + k)$, and that we can walk along the antidiagonal of a pair i, j by subtracting and adding k to the row/column index. To control how big this k can be considering n , the upper bound of k is defined as $\min(i - 1, n - j)$. The upper bound avoids indices going beyond n on the columns and rows. Notice that with the expression ψ_n^{diag2} , for each pair i, j on the board, we check all the values in their antidiagonal to verify if there are two queens. Therefore, if the expression ψ_n^{diag2} is true, there is a queen, and only one, in each antidiagonal.

Thus, we showed that we can write the requirements for the n -Queens problem as a propositional wff $\psi_n = \psi_n^{\text{row}} \wedge \psi_n^{\text{col}} \wedge \psi_n^{\text{diag1}} \wedge \psi_n^{\text{diag2}}$.

Exercise 2

The exercise asked us to simplify the formal definition of substitution by modifying the QPL BNF to guarantee that in every wff:

1. there is at most one binding occurrence for the same variable;
2. a variable cannot have both free and bound occurrences.

There are a few ways I thought about doing this exercise depending on if the new BNF should be as expressive as the original QPL BNF or not.

Case 1: Not as expressive as QPL

If we wanted to define a BNF for QPL following the two guarantees to generate wffs of the format $Q\mathbf{x}.\varphi$, *i.e.*, one quantifier on the outside of an expression φ , we could do the following:

$$\begin{aligned} \psi &::= \forall \mathbf{x}.\varphi \mid \exists \mathbf{x}.\varphi \\ \varphi &::= \perp \mid \top \mid \mathbf{x} \mid (\neg \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \end{aligned}$$

Considering that we create a condition that we cannot use a free variable as bound (I don't know how to express this in BNF), this BNF would ensure the two conditions, however, it would also not be as expressive as QPL. Notice that we would not be able to express $p \rightarrow \forall q.(\varphi \wedge \varphi)$ using this BNF.

Case 2: Trying to be similarly expressive as QPL

In the second try, I attempted to create a BNF that guarantee the two conditions and is as expressive as QPL:

$$\varphi ::= \perp \mid \top \mid \mathbf{FV} \mid (\neg \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid \forall_{\mathbf{BV}}.\psi \mid \exists_{\mathbf{BV}}.\psi$$

$$\psi ::= \perp \mid \top \mid \mathbf{BV} \mid (\neg\psi) \mid (\psi \wedge \psi) \mid (\psi \vee \psi) \mid (\psi \rightarrow \psi)$$

$$\mathbf{FV} ::= a \mid b \mid \dots \mid z$$

$$\mathbf{BV} ::= a' \mid b' \mid \dots \mid z'$$

Using this BNF, we guarantee that a variable cannot have both free and bound occurrences, because we split the variables in \mathbf{BV} and \mathbf{FV} . And if we explicitly say that we cannot repeat any variables already used, then, we would guarantee the first condition. I don't know if it would be possible to express this in BNF (in did some research about it and did not find any solution). Notice that this BNF still is not as expressive as QPL, because we cannot nest quantifiers $Q_1 \dots Q_n.(\varphi)$. However, all my attempts to create a BNF that allowed nested quantifiers broke the “at most one binding occurrence of the same variable”, unless we stated that we could not repeat bound variables.

To answer the remainder of the exercise: if a BNF that satisfy both condition exists, then the last two substitution rules would be simpler, because you would not have to worry about variable captures.

Exercise 3

The Lean template file with the solutions is available on [GitHub](#).

Exercise 4

The Lean template file with the solutions is available on [GitHub](#).

Problem 2

The Lean template file with the solutions is available on [GitHub](#).