CS3234: Logic and Formal Systems
Assignment 4, due 11:00 AM, October 7.

1. (10 marks) **Note updated problem statement**

   Consider the encoding of tournament scheduling in SAT presented in Lecture 7 (B/W slide 11). Encode the following constraints as propositional formulas. You may introduce propositional atoms other than $p_{x,y,z}$. In this case, formulate constraints between the new variables and $p_{x,y,z}$, also as propositional formulas. For example, if you introduce a variable $b_{1,4}$, which encodes whether Team 1 has a bye in Date 4, you need the following formula that expresses the connection between $b_{1,4}$ and the p variables:

$$
\begin{aligned}
(b_{1,4} \quad \rightarrow \quad & \neg p_{1,2,4} \wedge \neg p_{1,3,4} \wedge \cdots \wedge \neg p_{1,9,4} \wedge \\
& \neg p_{2,1,4} \wedge \neg p_{3,1,4} \wedge \cdots \wedge \neg p_{9,1,4}) \wedge \\
(\neg p_{1,2,4} \wedge \neg p_{1,3,4} \wedge \cdots \wedge \neg p_{1,9,4} \wedge & \\
\neg p_{2,1,4} \wedge \neg p_{3,1,4} \wedge \cdots \wedge \neg p_{9,1,4} \quad \rightarrow \quad & b_{1,4})
\end{aligned}
$$

   As in the previous formula, you may use the $\cdots$ notation, if the meaning is clear.

   - UNC (Team 1) plays Duke (Team 2) in the last date and in Date 11.

     Solution. The slides mention that $p_{x,y,z}$ encodes that Team x has a home game against Team y in Date z. Thus, the requirement that UNC plays Duke in Date 11 needs to consider both possibilities for the venue: UNC plays home or Duke plays home. This is encoded by:

$$p_{1,2,11} \vee p_{2,1,11}$$

     Similarly, the two options need to be considered for the last date. Overall, the following formula captures the requirement:

$$(p_{1,2,11} \vee p_{2,1,11}) \wedge (p_{1,2,18} \vee p_{2,1,18})$$

   - The following pairings must occur at least once in Dates 11 to 18: Duke (Team 2) – GT (Team 3), Duke (Team 2) – Wake (Team 4), GT (Team 3) – UNC (Team 1), UNC (Team 1) – Wake (Team 4).

     Solution. The fact that a pairing occurs at least once can be represented by a disjunction of all possibilities, again considering both options for the venue. Overall, the following conjunction results:

$$
\begin{aligned}
&(p_{2,3,11} \vee p_{3,2,11} \vee p_{2,3,12} \vee p_{3,2,12} \vee \cdots \vee p_{2,3,18} \vee p_{3,2,18}) \wedge \\
&(p_{2,4,11} \vee p_{4,2,11} \vee p_{2,4,12} \vee p_{4,2,12} \vee \cdots \vee p_{2,4,18} \vee p_{4,2,18}) \wedge \\
&(p_{3,1,11} \vee p_{1,3,11} \vee p_{3,1,12} \vee p_{1,3,12} \vee \cdots \vee p_{3,1,18} \vee p_{1,3,18}) \wedge \\
&(p_{1,4,11} \vee p_{4,1,11} \vee p_{1,4,12} \vee p_{4,1,12} \vee \cdots \vee p_{1,4,18} \vee p_{4,1,18})
\end{aligned}
$$

   - No team can play away on both last dates.

Solution. A team x plays away on a date, if there is some other team that plays home against x on that date. Thus, one way of expressing the constraint is:

$$
\begin{array}{l}
\neg((p_{2,1,17} \vee p_{3,1,17} \vee p_{4,1,17} \vee \cdots \vee p_{9,1,17}) \\
\quad \wedge \\
\quad (p_{2,1,18} \vee p_{3,1,18} \vee p_{4,1,18} \vee \cdots \vee p_{9,1,18}) \\
\ \ ) \\
\wedge \\
\neg((p_{1,2,17} \vee p_{3,2,17} \vee p_{4,2,17} \vee \cdots \vee p_{9,2,17}) \\
\quad \wedge \\
\quad (p_{1,2,18} \vee p_{3,2,18} \vee p_{4,2,18} \vee \cdots \vee p_{9,2,18}) \\
\ \ ) \\
\wedge \\
\vdots \\
\wedge \\
\neg((p_{1,9,17} \vee p_{2,9,17} \vee p_{3,9,17} \vee \cdots \vee p_{8,9,17}) \\
\quad \wedge \\
\quad (p_{1,9,18} \vee p_{2,9,18} \vee p_{3,9,18} \vee \cdots \vee p_{8,9,18}) \\
\ \ )
\end{array}
$$

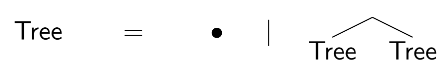
- Dates 1 and 8 are mirrored. This means two teams play each other in Date 1, iff they play each other in Date 8.

Solution. Mirroring can be enforced by stating the double-implication of the corresponding p variables:

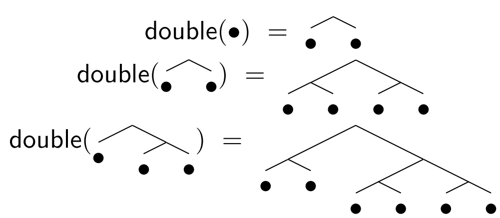
$$
\begin{aligned}
&((p_{1,2,1} \to p_{2,1,8}) \wedge (p_{2,1,8} \to p_{1,2,1}) \wedge \\
&\ (p_{1,3,1} \to p_{3,1,8}) \wedge (p_{3,1,8} \to p_{1,3,1}) \wedge \\
&\ \vdots \\
&\ (p_{1,9,1} \to p_{9,1,8}) \wedge (p_{9,1,8} \to p_{1,9,1})) \\
&\wedge \\
&((p_{2,1,1} \to p_{1,2,8}) \wedge (p_{1,2,8} \to p_{2,1,1}) \wedge \\
&\ (p_{2,3,1} \to p_{3,2,8}) \wedge (p_{3,2,8} \to p_{2,3,1}) \wedge \\
&\ \vdots \\
&\ (p_{2,9,1} \to p_{9,2,8}) \wedge (p_{9,2,8} \to p_{2,9,1})) \\
&\wedge \\
&\vdots \\
&\wedge \\
&((p_{9,1,1} \to p_{1,9,8}) \wedge (p_{1,9,8} \to p_{9,1,1}) \wedge \\
&\ (p_{9,2,1} \to p_{2,9,8}) \wedge (p_{2,9,8} \to p_{9,2,1}) \wedge \\
&\ \vdots \\
&\ (p_{8,9,1} \to p_{9,8,8}) \wedge (p_{9,8,8} \to p_{8,9,1}))
\end{aligned}
$$

2. (10 marks) (a.k.a. exercise 8 in the lecture notes)

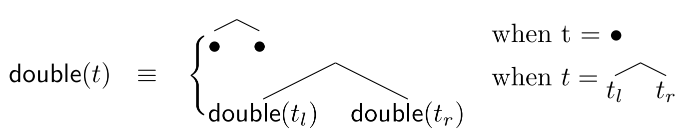
   We inductively define the set of binary trees as follows:



   Define (paper and Coq) a function that doubles all of the leaves in a tree. For example, this function should behave as follows:



Solution. Here is the double function:



And here is the Coq implementation:

```
Fixpoint double (t : Tree) : Tree :=
  match t with
   | Leaf => Node Leaf Leaf
   | Node tl tr => Node (double tl) (double tr)
  end.
```

3. (10 marks) (a.k.a. exercise 10 in the lecture notes)

   Using the double function you defined for the previous problem, and the following definition for leaves:

$$\mathsf{leaves}(t) \quad \equiv \quad \begin{cases} 1 & \text{when t} = \bullet \\ \mathsf{leaves}(t_l) + \mathsf{leaves}(t_r) & \text{when } t = \widehat{t_l \quad t_r} \end{cases}$$

   Please prove:

$$\forall t \ \big(\mathsf{leaves}(\mathsf{double}(t)) \ = \ \mathsf{leaves}(t) + \mathsf{leaves}(t)\big)$$

   Solution. By induction on the structure of t using induction hypothesis

$$\mathsf{IH}(t) \quad \equiv \quad \mathsf{leaves}(\mathsf{double}(t)) \ = \ \mathsf{leaves}(t) + \mathsf{leaves}(t)$$

   - Case 1: $(t = \bullet)$. We have

$$\begin{aligned} \mathsf{leaves}(\mathsf{double}(\bullet)) \quad &= \quad \mathsf{leaves}(\widehat{\bullet \quad \bullet}) \\ &= \quad \mathsf{leaves}(\bullet) + \mathsf{leaves}(\bullet) \end{aligned}$$

     This is enough to prove case 1.

   - Case 2: $(t = \widehat{t_l \quad t_r})$. We can assume the induction hypotheses:

$$\begin{aligned} \mathsf{leaves}(\mathsf{double}(t_l)) \ &= \ \mathsf{leaves}(t_l) + \mathsf{leaves}(t_l) \\ \mathsf{leaves}(\mathsf{double}(t_r)) \ &= \ \mathsf{leaves}(t_r) + \mathsf{leaves}(t_r) \end{aligned}$$

     We therefore have:

$$\begin{aligned} &\mathsf{leaves}(\mathsf{double}(\widehat{t_l \quad t_r})) \\ &= \ \mathsf{leaves}\Big(\widehat{\mathsf{double}(t_l) \quad \mathsf{double}(t_r)}\Big) \\ &= \ \mathsf{leaves}(\mathsf{double}(t_l)) + \mathsf{leaves}(\mathsf{double}(t_r)) \\ &= \ \mathsf{leaves}(t_l) + \mathsf{leaves}(t_l) + \mathsf{leaves}(t_r) + \mathsf{leaves}(t_r) \\ &= \ \big(\mathsf{leaves}(t_l) + \mathsf{leaves}(t_r)\big) + \big(\mathsf{leaves}(t_l) + \mathsf{leaves}(t_r)\big) \\ &= \ \mathsf{leaves}(\widehat{t_l \quad t_r}) + \mathsf{leaves}(\widehat{t_l \quad t_r}) \end{aligned}$$

     This is enough to prove case 2.

Thus by structural induction we have proved that

$$\forall t \ \big(\mathsf{leaves}(\mathsf{double}(t)) \ = \ \mathsf{leaves}(t) + \mathsf{leaves}(t)\big)$$

4. (10 marks)

Now we define the function nodes as follows:

$$\mathsf{nodes}(t) \quad \equiv \quad \begin{cases} 0 & \text{when t} = \bullet \\ 1 + \mathsf{nodes}(t_l) + \mathsf{nodes}(t_r) & \text{when } t = \widehat{t_l \quad t_r} \end{cases}$$

Please prove:

$$\forall t \ \big(\mathsf{nodes}(\mathsf{double}(t)) \ = \ \mathsf{nodes}(t) + \mathsf{leaves}(t)\big)$$

Solution. By induction on the structure of t using induction hypothesis

$$\mathsf{IH}(t) \quad \equiv \quad \mathsf{nodes}(\mathsf{double}(t)) \ = \ \mathsf{nodes}(t) + \mathsf{leaves}(t)$$

- Case 1: $(t = \bullet)$. We have

$$\begin{aligned} \mathsf{nodes}(\mathsf{double}(\bullet)) \quad &= \quad \mathsf{nodes}(\widehat{\bullet \quad \bullet}) \\ &= \quad 1 + \mathsf{nodes}(\bullet) + \mathsf{nodes}(\bullet) \\ &= \quad 1 + 0 + 0 \\ &= \quad 0 + 1 \\ &= \quad \mathsf{nodes}(\bullet) + \mathsf{leaves}(\bullet) \end{aligned}$$

  This is enough to prove case 1.

- Case 2: $(t = \widehat{t_l \quad t_r})$. We can assume the induction hypotheses:

$$\begin{aligned} \mathsf{nodes}(\mathsf{double}(t_l)) \ &= \ \mathsf{nodes}(t_l) + \mathsf{leaves}(t_l) \\ \mathsf{nodes}(\mathsf{double}(t_r)) \ &= \ \mathsf{nodes}(t_r) + \mathsf{leaves}(t_r) \end{aligned}$$

  We then have:

$$\begin{aligned} &\mathsf{nodes}(\mathsf{double}(\widehat{t_l \quad t_r})) \\ &= \ \mathsf{nodes}\Big(\widehat{\mathsf{double}(t_l) \qquad \mathsf{double}(t_r)}\Big) \\ &= \ 1 + \mathsf{nodes}(\mathsf{double}(t_l)) + \mathsf{nodes}(\mathsf{double}(t_r)) \\ &= \ 1 + \mathsf{nodes}(t_l) + \mathsf{leaves}(t_l) + \mathsf{nodes}(t_r) + \mathsf{leaves}(t_r) \\ &= \ \big(1 + \mathsf{nodes}(t_l) + \mathsf{nodes}(t_r)\big) + \big(\mathsf{leaves}(t_l) + \mathsf{leaves}(t_r)\big) \\ &= \ \mathsf{nodes}(\widehat{t_l \quad t_r}) + \mathsf{leaves}(\widehat{t_l \quad t_r}) \end{aligned}$$

Thus by structural induction we have proved that

$$\forall t \ \big(\mathsf{nodes}(\mathsf{double}(t)) \ = \ \mathsf{nodes}(t) + \mathsf{leaves}(t)\big)$$

5. (15 marks) (a.k.a. exercise 11 in the lecture notes)

   Suppose we attempt to define streams inductively via the rule

$$\frac{n : \mathsf{nat} \qquad s : \mathsf{stream}}{n \mathbin{@} s \ : \ \mathsf{stream}}\ \mathsf{Strm}$$

   Prove that in that case, stream is empty; that is,

$$\neg \exists s : \mathsf{stream}(\top)$$

   Note that this is (deMorgan-) equivalent to:

$$\forall s : \mathsf{stream}(\bot)$$

   If you set this up correctly, the proof should be very short.

   Solution. By induction on the structure of s. We use $\mathsf{IH}(s) = \bot$.

   - Case 1: $(s = n@s')$. We can assume $\mathsf{IH}(s')$, that is, $\bot$. Since we are trying to prove $\mathsf{IH}(s)$, that is, $\bot$, we are done.

   Thus by we have proved $\forall s : \mathsf{stream}(\bot)$ by induction.

6. (15 marks) (a.k.a. exercise 5 in the lecture notes)

   Give a series of rules and a complete and invertible set of objects satisfying those rules that is **neither** the least (inductive) nor greatest (coinductive).

   Solution. For the rules, use the following:

$$\frac{t : \tau}{\mathsf{Y}(t) : \tau}\ \mathsf{Y}\tau \qquad\qquad \frac{t : \tau}{\mathsf{Z}(t) : \tau}\ \mathsf{Z}\tau$$

   Define the set $\mathbb{S}$ as follows:

$$\begin{array}{lllll}
\mathbb{S} \ \equiv \ \Big\{ & \mathsf{Y}(\mathsf{Y}(\mathsf{Y}(\mathsf{Y}(\ldots)))), \text{ (call this element } \mathsf{Y}_\omega) & & & \\
& \mathsf{Z}(\mathsf{Y}_\omega), \quad \mathsf{Z}(\mathsf{Z}(\mathsf{Y}_\omega)), & \mathsf{Z}(\mathsf{Z}(\mathsf{Z}(\mathsf{Y}_\omega))), & \mathsf{Z}(\mathsf{Z}(\mathsf{Z}(\mathsf{Z}(\mathsf{Y}_\omega)))), & \ldots \\
& \qquad\qquad \mathsf{Y}(\mathsf{Z}(\mathsf{Y}_\omega)), & \mathsf{Y}(\mathsf{Z}(\mathsf{Z}(\mathsf{Y}_\omega))), & \mathsf{Y}(\mathsf{Z}(\mathsf{Z}(\mathsf{Z}(\mathsf{Y}_\omega)))), & \ldots \\
& & \mathsf{Z}(\mathsf{Y}(\mathsf{Z}(\mathsf{Y}_\omega))), & \mathsf{Z}(\mathsf{Y}(\mathsf{Z}(\mathsf{Z}(\mathsf{Y}_\omega)))), & \ldots \\
& & \mathsf{Y}(\mathsf{Y}(\mathsf{Z}(\mathsf{Y}_\omega))), & \mathsf{Z}(\mathsf{Z}(\mathsf{Y}(\mathsf{Z}(\mathsf{Y}_\omega)))), & \ldots \\
& & & \mathsf{Y}(\mathsf{Y}(\mathsf{Z}(\mathsf{Z}(\mathsf{Y}_\omega)))), & \ldots \\
& & & \mathsf{Y}(\mathsf{Z}(\mathsf{Y}(\mathsf{Z}(\mathsf{Y}_\omega)))), & \ldots \\
& & & \mathsf{Z}(\mathsf{Y}(\mathsf{Y}(\mathsf{Z}(\mathsf{Y}_\omega)))), & \ldots \\
& & & \mathsf{Y}(\mathsf{Y}(\mathsf{Y}(\mathsf{Z}(\mathsf{Y}_\omega)))), & \ldots \\
& \ldots \Big\} & & &
\end{array}$$

   In other words, $\mathbb{S}$ contains the infinite-Y element Y_ω, plus all **finite** sequences of Z and Y (including the empty sequence), followed by a single

Z, followed by Y_ω. The set $\mathbb{S}$ is complete since given an element $s \in \mathbb{S}$, applying the constructors Y or Z one time still leaves you in $\mathbb{S}$ (we just made the finite sequence at the beginning one element longer, except in the case where we added an extra Y to the front of Y_ω, which just gives us Y_ω again). The set $\mathbb{S}$ is invertible since every element is made from the rules $\mathsf{Y}\tau$ and $\mathsf{Z}\tau$. The set $\mathbb{S}$ is not the least set since the least complete and invertible set satisfying the rules $\mathsf{Y}\tau$ and $\mathsf{Z}\tau$ is empty (see problem 5). The set $\mathbb{S}$ is not the greatest complete and invertible set since it does not contain, among other things, the infinite-Z element:

$$\mathsf{Z}_\omega \;\equiv\; \mathsf{Z}(\mathsf{Z}(\mathsf{Z}(\mathsf{Z}(\ldots)))) \;\notin \mathbb{S}$$

7. (20 marks)

Please prove

$$\forall t\ \big((t = \overset{\frown}{\;t \quad t\;}) \;\Rightarrow\; \bot\big)$$

You can assume that generators are injective; that is, from

$$\overset{\frown}{\;t_{11} \quad t_{12}\;} \;=\; \overset{\frown}{\;t_{21} \quad t_{22}\;}$$

you may conclude

$$t_{11} \;=\; t_{21} \qquad \text{and} \qquad t_{12} \;=\; t_{22}$$

You may also assume that

$$\forall t_1 \forall t_2 \big((\bullet \;=\; \overset{\frown}{\;t_1 \quad t_2\;}) \;\Rightarrow\; \bot\big)$$

Solution. We will first prove the related fact:

$$\forall t \forall t'\ (t = \overset{\frown}{\;t \quad t'\;} \;\Rightarrow\; \bot)$$

By induction on the structure of t using the induction hypothesis:

$$\mathsf{IH}'(t) \;\equiv\; \forall t'\ (t = \overset{\frown}{\;t \quad t'\;} \;\Rightarrow\; \bot)$$

- Case 1: $(t = \bullet)$. We want to prove $\mathsf{IH}'(\bullet)$, that is,

  $$\forall t'\ (\bullet = \overset{\frown}{\;\bullet \quad t'\;} \;\Rightarrow\; \bot)$$

  This is automatic from the given assumption

  $$\forall t_1 \forall t_2 \big((\bullet \;=\; \overset{\frown}{\;t_1 \quad t_2\;}) \;\Rightarrow\; \bot\big)$$

  So we are done with case 1.

- Case 2: ($t = \widehat{t_l \quad t_r}$). We assume $\mathsf{IH}'(t_l)$ and $\mathsf{IH}'(t_r)$. We will only need $\mathsf{IH}'(t_l)$, *i.e.*:

  $$\forall t' \ (t_l = \widehat{t_l \quad t'} \quad \Rightarrow \quad \bot)$$

  Now assume $t = \widehat{t \quad t}$; that is,

  $$\widehat{t_l \quad t_r} \quad = \quad \widehat{\widehat{t_l \quad t_r} \quad \widehat{t_l \quad t_r}}$$

  Since generators are injective we have

  $$t_l = \widehat{t_l \quad t_r} \qquad\qquad \text{and} \qquad\qquad t_r = \widehat{t_l \quad t_r}$$

  By $\mathsf{IH}'(t_l)$, we can conclude $\bot$ by setting $t' = t_r$. Thus we are done with case 2.

We have therefore proved

$$\forall t \forall t' \ (t = \widehat{t \quad t'} \quad \Rightarrow \quad \bot)$$

by structural induction.

It is clear that this is strictly stronger than the original goal of

$$\forall t \ (t = \widehat{t \quad t} \quad \Rightarrow \quad \bot)$$

By simply setting $t' = t$.