COMP2620/6262 (Logic) Tutorial

Week 9

Semester 1, 2025

Tutorial Quiz

In each tutorial, apart from week 2, there is a short quiz on skills practised in the previous tutorial. Your top 7 quiz attempts, out of the 9 available, will collectively count for 50% of your final mark.

This week's quiz is on **tableaux** for first order logic, with branching quantifier rules. Your tutor will hand out blank paper, on which you should clearly write your university ID and name. Your tutor will also hand out paper with all tableaux rules for first order logic, with branching quantifier rules. They will then write a signed proposition on the whiteboard. You should then use the tableaux method to extract a finite satisfying model. Be explicit about what your model (universe of discourse and interpretation of quantifiers) is. You will have **twenty minutes** to do this.

You should not attempt to construct a completed tableaux, because some branches will be infinite. You do not need to pursue branches that you think will close, or multiple open branches. If your tableau becomes repetitive due to multiple applications of certain quantifier rules, you may skip some repetitive steps so long as you justify your lines by explaining which line you got started with, which line this part of your tableau resembles, and which substitutions for bound variables you used to get there, e.g. 'from (3), as for (6), with $[b/x]$ and $[c/y]$'. Do not skip any steps the first time you apply these rules.

You are not permitted to have any other resource on the table during this quiz, including any electronic device. If you finish your quiz before time elapses you may put your hand up and your tutor will collect your sheet. Once you have done this, you may get a device out and start work silently on this week's questions. If you are still working when time elapses you must stop writing immediately and let your tutor collect your paper.

This Week's Exercises

This tutorial involves the tableaux rules for linear temporal logic:

$$\frac{\mathbf{T}:\bot}{\times} \qquad \frac{\mathbf{T}:\neg\varphi}{\mathbf{F}:\varphi} \qquad \frac{\mathbf{F}:\neg\varphi}{\mathbf{T}:\varphi} \qquad \frac{\mathbf{T}:\varphi\vee\psi}{\mathbf{T}:\varphi \qquad \mathbf{T}:\psi} \qquad \frac{\mathbf{F}:\varphi\vee\psi}{\begin{array}{c}\mathbf{F}:\varphi\\ \mathbf{F}:\psi\end{array}}$$

$$\frac{\mathbf{T}:\varphi\wedge\psi}{\begin{array}{c}\mathbf{T}:\varphi\\ \mathbf{T}:\psi\end{array}} \qquad \frac{\mathbf{F}:\varphi\wedge\psi}{\mathbf{F}:\varphi \qquad \mathbf{F}:\psi} \qquad \frac{\mathbf{T}:\varphi\to\psi}{\mathbf{F}:\varphi \qquad \mathbf{T}:\psi} \qquad \frac{\mathbf{F}:\varphi\to\psi}{\begin{array}{c}\mathbf{T}:\varphi\\ \mathbf{F}:\psi\end{array}}$$

$$\frac{\mathbf{T}:G\varphi}{\begin{array}{c}\mathbf{T}:\varphi\\ \mathbf{T}:XG\varphi\end{array}} \qquad \frac{\mathbf{F}:G\varphi}{\mathbf{F}:\varphi \qquad \mathbf{F}:XG\varphi} \qquad \frac{\mathbf{T}:F\varphi}{\mathbf{T}:\varphi \qquad \mathbf{T}:XF\varphi} \qquad \frac{\mathbf{F}:F\varphi}{\begin{array}{c}\mathbf{F}:\varphi\\ \mathbf{F}:XF\varphi\end{array}}$$

$$\frac{\mathbf{T}:\varphi\,U\,\psi}{\mathbf{T}:\psi \qquad \begin{array}{c}\mathbf{T}:\varphi\\ \mathbf{T}:X(\varphi\,U\,\psi)\end{array}} \qquad \frac{\mathbf{F}:\varphi\,U\,\psi}{\begin{array}{c}\mathbf{F}:\psi\\ \mathbf{F}:\varphi \qquad \mathbf{F}:X(\varphi\,U\,\psi)\end{array}}$$

If the node is poised, you may step:

$$\frac{\mathbf{T}:X\varphi_1 \quad \cdots \quad \mathbf{T}:X\varphi_m \quad \mathbf{F}:X\psi_1 \quad \cdots \quad \mathbf{T}:X\psi_n}{\begin{array}{c}⇟\\ \mathbf{T}:\varphi_1,\cdots,\mathbf{T}:\varphi_m,\mathbf{F}:\psi_1,\cdots,\mathbf{F}:\psi_n\end{array}}$$

- **Loop rule:** If we call our current node n, and it is poised, and

- there is a node $l < n$ such that all base case and X-formulas of n were already in l;
- and for every X-eventuality - respectively $\mathbf{T} : XF\varphi$, $\mathbf{T} : X(\varphi\, U\, \psi)$, or $\mathbf{F} : G\varphi$ - in l we have, respectively, $\mathbf{T} : \varphi$, $\mathbf{T} : \psi$, or $\mathbf{F} : \varphi$, in some node m such that $l < m \leq n$;

then the current branch terminates open (satisfiable).

- **Simple repetition rule:** If we call our current node n, and it is poised, and
  - there is a node $l < n$ with the same base case and X-formulas as n, and this includes at least one X-eventuality;
  - and there is no X-eventuality - respectively $\mathbf{T} : XF\varphi$, $\mathbf{T} : X(\varphi\, U\, \psi)$, or $\mathbf{F} : G\varphi$ - such that, respectively, $\mathbf{T} : \varphi$, $\mathbf{T} : \psi$, or $\mathbf{F} : \varphi$, is in a node m such that $l < m \leq n$;

  then the current branch should be closed (unsatisfiable) with a cross.

1. For each of the following sequents, use tableaux to either show that they are valid by crossing every branch, or show that are they invalid by finding an open terminated branch. You do not need to explore the tableau further if you have found a terminated open branch. If the sequent is not valid, draw a diagram presenting the model that you extract from your tableau.

   For these questions, you will not need the loop and repetition rules.

   - $XFp \vdash Fp$
   - $\bot\, U\, p \vdash p$
   - $p \vdash \bot\, U\, p$
   - $p \to q, Xp \vdash Xq$
   - $p, Xp \vdash Gp$
   - $p\, U\, q, q\, U\, r \vdash p\, U\, r$

2. **The test at the start of the next tutorial will resemble this question, except that you will not be asked to explicitly extract a satisfying model where one exists.**

   Perform the same task for each of the following sequents, which will require the loop or simple repetition rules. If you find parts of your tableau are repeating parts you have already constructed, you explain in English what you conclude from that. rather than repeating yourself.

   - $Fp \vdash FXp$
   - $FXp \vdash Fp$
   - $Fp \vdash XFp$
   - $Fp, G(Xp \to p) \vdash p$
   - $\vdash Fp\, U\, \neg p$
   - $Gp \vdash GGp$
   - $p, Xp\, U\, q \vdash F(p \wedge q)$
   - $Gp \vee Gq \vdash G(p \vee q)$
   - $\neg q\, U\, p, \neg r\, U\, q \vdash \neg r\, U\, p$
   - $FGp \vdash GFp$ (Tricky)