COMP2620/6262 (Logic) Tutorial

Week 6

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 **natural deduction** for first order logic. 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 natural deduction rules They will then write a sequent on the board which you should attempt to prove with these rules. You should use the five part notation for natural deduction: which premises are being used; a line number; a formula; previous lines used; and the rule name. You will have **eighteen minutes** to attempt this proof.

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 natural deduction rules and the semantics for first order logic:

$$\frac{}{\varphi \vdash \varphi}\ A$$

$$\frac{\Gamma \vdash \varphi \quad \Gamma' \vdash \psi}{\Gamma, \Gamma' \vdash \varphi \wedge \psi}\ \wedge I \qquad \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi}\ \wedge E1 \qquad \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi}\ \wedge E2$$

$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}\ \to I \qquad \frac{\Gamma \vdash \varphi \to \psi \quad \Gamma' \vdash \varphi}{\Gamma, \Gamma' \vdash \psi}\ \to E$$

$$\frac{\Gamma, \varphi \vdash \bot}{\Gamma \vdash \neg\varphi}\ \neg I \qquad \frac{\Gamma \vdash \neg\varphi \quad \Gamma' \vdash \varphi}{\Gamma, \Gamma' \vdash \bot}\ \neg E \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash \varphi}\ \bot E$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi}\ \vee I1 \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi}\ \vee I2 \qquad \frac{\Gamma \vdash \varphi \vee \psi \quad \Gamma', \varphi \vdash \sigma \quad \Gamma'', \psi \vdash \sigma}{\Gamma, \Gamma', \Gamma'' \vdash \sigma}\ \vee E$$

$$\frac{\Gamma \vdash \neg\neg\varphi}{\Gamma \vdash \varphi}\ \neg\neg E$$

$$\frac{}{\vdash t = t}\ = I \qquad \frac{\Gamma \vdash t = u \quad \Gamma' \vdash \varphi[t/x]}{\Gamma, \Gamma' \vdash \varphi[u/x]}\ = E$$

$$\frac{\Gamma \vdash \varphi[t/x]}{\Gamma \vdash \exists x\varphi}\ \exists I \qquad a \notin FV(\Gamma, \varphi, \Gamma', \psi) : \ \frac{\Gamma \vdash \exists x\varphi \quad \Gamma', \varphi[a/x] \vdash \psi}{\Gamma, \Gamma' \vdash \psi}\ \exists E$$

$$a \notin FV(\Gamma, \varphi) : \ \frac{\Gamma \vdash \varphi[a/x]}{\Gamma \vdash \forall x\varphi}\ \forall I \qquad \frac{\Gamma \vdash \forall x\varphi}{\Gamma \vdash \varphi[t/x]}\ \forall E$$

We define $\vDash_{\mathcal{M},e} \varphi$, saying that φ is *satisfied* by $(\mathcal{M}, e)$, as:

- $\vDash_{\mathcal{M},e} P(t_1, \ldots, t_n)$ if $(t_1^{\mathcal{M},e}, \ldots, t_n^{\mathcal{M},e}) \in P^{\mathcal{M}}$
- $\vDash_{\mathcal{M},e} t = u$ if $t^{\mathcal{M},e} = u^{\mathcal{M},e}$ in the universe of discourse D

- $\vDash_{\mathcal{M},e} \bot$ never
- $\vDash_{\mathcal{M},e} \neg\varphi$ if it is not the case that $\vDash_{\mathcal{M},e} \varphi$
- $\vDash_{\mathcal{M},e} \varphi \wedge \psi$ if $\vDash_{\mathcal{M},e} \varphi$ and $\vDash_{\mathcal{M},e} \psi$
- $\vDash_{\mathcal{M},e} \varphi \vee \psi$ if $\vDash_{\mathcal{M},e} \varphi$ or $\vDash_{\mathcal{M},e} \psi$ (or both)
- $\vDash_{\mathcal{M},e} \varphi \to \psi$ if $\vDash_{\mathcal{M},e} \varphi$ implies $\vDash_{\mathcal{M},e} \psi$
- $\vDash_{\mathcal{M},e} \forall x\varphi$ if for all $d \in D$ we have $\vDash_{\mathcal{M},e[x\mapsto d]} \varphi$
- $\vDash_{\mathcal{M},e} \exists x\varphi$ if there exists $d \in D$ such that $\vDash_{\mathcal{M},e[x\mapsto d]} \varphi$

1. Let R be a binary predicate and f be a unary function. For each of the formulas below, ***(a)*** suggest a model and environment that satisfies it, and ***(b)*** suggest a model and environment that does not satisfy it.

   - $\forall x\, R(x, f\, x)$
     Solution. For all these questions there is a huge range of possible correct answers. For this one, let's define our universe of discourse as the integers, and R as equality. Then this formula is satisfied by the model in which we define f as the identity, and not satisfied if we define f as anything else, e.g. the successor function. Because there are no free variables, we do not need to specify an environment.
   - $R(x, x) \to \bot$
     Solution. Take our universe as the integers again, and say that $(i, j) \in R^{\mathcal{M}}$ when $i + j = 0$. Then the formula is not satisfied if our environment maps x to 0. For the 'is satisfied' case we cannot make the right hand side of the implication true, so we must make the left hand side false; let the environment map x to your favourite nonzero number.
   - $R(x, y) \wedge R(y, z) \to R(x, z)$
     Solution. Let our universe be the integers again, and say that $(i, j) \in R^{\mathcal{M}}$ exactly if $i \neq j$. Then the formula is satisfied e.g. by any environment that maps x, y, and z to different numbers (or any environment that maps x and y to the same number, or any environment that maps y and z to the same number, because these make the left side of the implication false, so the whole implication true). The formula is not satisfied e.g. by the environment mapping x to 42, y to 7, and z also to 42.
   - $\exists x\, f\, x = y$
     Solution. On the integers again, let f be the constant function mapping any input to 13. Then this formula is satisfied if we choose the environment that maps y to 13, and not satisfied by any other assignment to y.

2. The key lemma towards the proof of soundness is the substitution lemma: $\vDash_{\mathcal{M},e} \varphi[t/x]$ if and only if $\vDash_{e[x\mapsto t^{\mathcal{M},e}]} \varphi$. It is proved by induction on the structure of φ. A few cases were given lectures. Prove it for the following cases:

   - $\neg\varphi$
     Solution. $\vDash_{\mathcal{M},e} (\neg\varphi)[t/x]$ if and only if
     $\vDash_{\mathcal{M},e} \neg(\varphi[t/x])$ (definition of substitution), if and only if
     it is not the case that $\vDash_{\mathcal{M},e} \varphi[t/x]$, if and only if
     it is not the case that $\vDash_{e[x\mapsto t^{\mathcal{M},e}]} \varphi$ (by induction), if and only if
     $\vDash_{e[x\mapsto t^{\mathcal{M},e}]} \neg\varphi$.
   - $u = v$ for terms u and v. This will require the substitution lemma for terms, $(u[t/x])^{\mathcal{M},e} = u^{\mathcal{M},e[x\mapsto t^{\mathcal{M},e}]}$.
     Solution. $\vDash_{\mathcal{M},e} (u = v)[t/x]$ if and only if
     $\vDash_{\mathcal{M},e} (u[t/x]) = (v[t/x])$ (definition of substitution), if and only if
     $(u[t/x])^{\mathcal{M},e} = (v[t/x])^{\mathcal{M},e}$ in the universe of discourse, if and only if
     $u^{\mathcal{M},e[x\mapsto t^{\mathcal{M},e}]} = v^{\mathcal{M},e[x\mapsto t^{\mathcal{M},e}]}$ (substitution lemma for terms), if and only if
     $\vDash_{e[x\mapsto t^{\mathcal{M},e}]} u = v$.

- $\forall y\varphi$

  Solution. $\vDash_{\mathcal{M},e} (\forall y\varphi)[t/x]$ if and only if
  $\vDash_{\mathcal{M},e} \forall y(\varphi[t/x])$ (definition of substitution), if and only if
  for all d in the universe of discourse, $\vDash_{\mathcal{M},e[y\mapsto d]} \varphi[t/x]$, if and only if
  $\vDash_{e[y\mapsto d][x\mapsto t^{\mathcal{M},e}]} \varphi$ (by induction), if and only if
  $\vDash_{e[x\mapsto t^{\mathcal{M},e}][y\mapsto d]} \varphi$ because there is no clash between x and either y and t, so this reordering does not change the environment, if and only if
  $\vDash_{e[x\mapsto t^{\mathcal{M},e}]} \forall y\varphi$

3. The soundness of natural deduction for first order logic is proved by induction on the length of proofs. We assume that anything proved by a shorter natural deduction proof is semantically valid (satisfied by all models and environments), and then show for each natural deduction rule separately that the conclusion of that rule is also a sequent that is semantically valid. A number of cases were proved in lectures.

   Prove the soundness of the following natural deduction rules:

   - $\wedge E1$

     Solution. Assume for induction that any model and environment that satisfies all formula of Γ also satisfies $\varphi \wedge \psi$. This means by definition that any such model and environment satisfies φ and also satisfies ψ. In particular, it satisfies φ.

   - $\to E$

     Solution. Assume for induction that any model and environment that satisfies all formula of Γ also satisfies $\varphi \to \psi$, and similarly for Γ' and φ. This means that any model and environment that satisfies both Γ and Γ' must satisfy both $\varphi \to \psi$ and φ. The former by definition says that if φ is satisfied then so is ψ. But φ is satisfied, so then so is ψ.

   - $\to I$

     Solution. Assume for induction that any model and environment that satisfies all formula of Γ, and also φ, also satisfies ψ. Consider any model and environment that satisfies all formula of Γ. There are two cases to consider. The first is that this model and environment satisfies φ. Then it must satisfy ψ, so it is true that the satisfaction of φ implies the satisfaction of ψ, so $\varphi \to \psi$ is satisfied. The second case is that φ is not satisfied. But then the satisfaction of φ vacuously implies the satisfaction of ψ, so $\varphi \to \psi$ is satisfied

   - $\exists I$

     Solution. Assume for induction that any model $\mathcal{M}$ and environment e that satisfies all formula of Γ, also satisfies $\varphi[t/x]$. By the substitution lemma, $\vDash_{\mathcal{M},e[x\mapsto t^{\mathcal{M},e}]} \varphi$. But then $t^{\mathcal{M},e}$ is the witness required by the definition of satisfaction of $\exists x\varphi$.

   - $\exists E$ (Tricky)

     Solution. Assume for induction that any model and environment that satisfies all of Γ also satisfies $\exists x\varphi$, and that any model and environment that satisfies all of Γ' and also $\varphi[a/x]$ for an eigenvariable a, also satisfies ψ.

     Suppose $\mathcal{M}, e$ satisfies all of Γ and Γ'. By the first assumption and the substitution lemma, there exists d in the universe of discourse such that

     $$\vDash_{\mathcal{M},e[x\mapsto d]} \varphi$$

     Now because a does not appear free in $\Gamma, \Gamma', \varphi, \psi$, changing the value of $e(a)$ will not change anything about the satisfaction of these formulas (do you accept this claim? If it needs a separate proof, how would you go about it?). So $\mathcal{M}, e[a \mapsto d]$ satisfies all of Γ and Γ', and so by the first assumption again,

     $$\vDash_{\mathcal{M},e[a\mapsto d][x\mapsto d]} \varphi$$

     but d is the output of the function $e[a \mapsto d]$ on input a, so we could write this as

     $$\vDash_{\mathcal{M},e[a\mapsto d][x\mapsto a^{\mathcal{M},e[a\mapsto d]}]} \varphi$$

     By the substitution lemma, this gives $\vDash_{\mathcal{M},e[a\mapsto d]} \varphi[a/x]$. But this is exactly what is needed to invoke our second assumption, so $\vDash_{\mathcal{M},e[a\mapsto d]} \psi$. But, again, ψ does not contain a so we no longer care what it is mapped to, so $\vDash_{\mathcal{M},e} \psi$.

4. **The test at the start of the next tutorial will resemble these questions.**

   Attempt the following puzzles from the Logic4Fun website.

   - Pet Show

     Solution.

     Sorts:
     ```
     owner enum : Franz, George, Harriet, Isabelle, Joanne.
     pet enum : duck, echidna, frog, lizard, spider.
     suburb enum : Ainslie, Banks, Chapman, Downer, Evatt.
     ```
     Vocabulary:
     ```
     function {
     owns (owner) : pet {all_different}.
     from (owner) : suburb {all_different}.
     position (owner) : natnum {all_different}.
     }
     ```
     Constraints:
     ```
     ALL x (position x > 0 AND position x < 6).

     position Harriet = 1 AND SOME x (position x = 5 AND owns x = spider).
     SOME x (from x = Evatt AND owns x <> echidna AND position x <> 2).
     from Franz = Banks.
     SOME x (position x = 4 AND x <> Joanne AND x <> Isabelle AND from x = Downer).
     owns Joanne = lizard AND position Joanne <> 3.
     SOME x SOME y (owns x = duck AND from x = Chapman AND from y = Ainslie AND position
     x > position y).
     ```

   - Parking Bay

     Solution.

     Sorts:
     ```
     vehicle enum : bus, car, taxi, truck, van.
     colour enum : black, blue, brown, green, white.
     ```
     Vocabulary:
     ```
     function {
     coloured (vehicle) : colour {all_different}.
     position (colour) : natnum {all_different}.
     }
     ```
     Constraints:
     ```
     ALL x (position x > 0 AND position x < 6).

     position white = position (coloured taxi) + 1 AND position (coloured van) > position
     blue + 1.
     coloured car <> brown AND position brown <> 3.
     position (coloured car) > position green AND position (coloured truck) > position
     (coloured car) AND coloured car <> white AND coloured truck <> white.
     position black = position (coloured bus) + 2.
     position (coloured bus) < position (coloured taxi) AND position (coloured van)
     < position (coloured truck).
     ```

   - Transfer List

     Solution. This one is a bit sneaky, because the information that there are not multiple players who stayed at the same club is in the text above the grey box!

     Sorts:
     ```
     player enum : Aristotle, Boole, Curry, DeMorgan, Etchemendy.
     position enum : goalkeeper, defender, midfield, wing, striker.
     club enum : Wanderers, City, United, Victory, Stars.
     ```
     Vocabulary:
     ```
     function {
     pos (player) : position {all_different}.
     old (player) : club {all_different}.
     ```

```
new (player) : club {all_different}.
}
```
Constraints:
```
ALL x ALL y (old x = new x AND old y = new y IMP x = y).

old Etchemendy = City AND SOME x (pos Etchemendy > pos x AND new x = City) AND
SOME y (pos Etchemendy < pos y AND pos y = wing AND new y = Wanderers).
SOME x (old x = Stars AND new x = Stars).
new Boole = United AND SOME x (old x = United AND pos x < pos Boole).
SOME x (pos x = defender AND old x = Victory AND x <> DeMorgan).
pos Curry = goalkeeper AND new Curry <> Victory.
```

- Trading Post (seriously time consuming, so you might like to return to it in the teaching break)

  Solution. This is my solution. This question is complex enough that yours might look quite different. Do sink some time into this one before running to the solution below, because once you have done so, you will have a very good grip on how to work with Logic4Fun.

  Sorts:
```
child enum : Andy, Belinda, Clara, Daniel, Elizabeth, Frederic.
surname enum : Garrett, Hardy, Irwin, Jones, Kennedy.
toy enum : kite, football, skateboard, puzzle, costume, tricks.
```
  Vocabulary:
```
function {
age (child) : natnum.
named (child) : surname {surjective}.
arrived (child) : natnum {all_different}.
gave (child) : toy {all_different}.
took (child) : toy {all_different}.
}

predicate {
twin (child,child) {commutative}.
girl (child) {hidden}.
}
```
  Constraints:
```
% Basic constraints on functions and predicates
ALL x (age x > 5 AND age x < 11).
ALL x (x > 5 AND x < 11 IMP SOME y (age y = x)).

ALL x (arrived x > 0 AND arrived x < 7).

ALL x (gave x <> took x).

ALL x ALL y (x twin y IFF (x <> y AND age x = age y)).
ALL x ALL y (age x = age y IFF named x = named y).

NOT girl Andy AND girl Belinda AND girl Clara AND NOT girl Daniel AND girl Elizabeth
AND NOT girl Frederic.

% Numbered clues.  Clue 8 is split over two lines
age Andy <> 6 AND age Daniel > age Andy.
ALL x ALL y (age x = 6 AND age y = 10 IMP arrived x < arrived y).
named Frederic = Jones AND age Frederic <> 10 AND SOME x (age x = 10 AND took x
= tricks).
named Clara = Hardy AND SOME x (arrived x < arrived Clara AND gave Clara = took
x AND gave x = took Clara).
SOME x (gave x = skateboard AND age x = age Belinda + 3) AND SOME y SOME z (NOT
girl y AND took y = skateboard AND took z = football AND age y < age z).
SOME x SOME y SOME z (gave x = skateboard AND gave y = football AND gave z = kite
```

```
AND arrived x < arrived z AND arrived y < arrived z).
SOME x (age x = 7 AND named x <> Irwin AND arrived x = 1 AND NOT SOME y (x twin
y)) AND SOME x (age x = 9 AND arrived x = 6 AND NOT SOME y (x twin y)).
SOME x SOME y (x twin y AND named x <> Garrett AND gave x = puzzle AND gave y =
costume AND arrived x < arrived y AND ((took x = football AND took y = kite) OR
(took x = kite AND took y = football))).
SOME w SOME x SOME y SOME z (w twin z AND x <> y AND arrived w < arrived x AND
arrived w < arrived y AND arrived x < arrived z AND arrived y < arrived z).
SOME x (named x = Garrett AND girl x AND took x <> costume).

% Final clue, hidden in the final question.
SOME x (NOT girl x AND took x = puzzle).
```