Optional questions are marked with a \(*\). The whole Section 3 is optional, but helps prepare for the next course. All sections are independent.

1 **Sequent calculus**

We consider the sequent calculus as seen in the course slides, that is the cut-free sequent calculus. Its rules and the axiom rule and all right and left rules (including right and left contractions, denoted rc and lc). We do not (yet) consider the cut rule presented in the course notes.

**Exercise 1: Sequent calculus proofs**

Give proofs in sequent calculus of the following formulas:

1. $A \lor (A \Rightarrow B)$

$$
\begin{array}{c}
\text{ax} \quad A \vdash A; B \\
\vdash A; A \Rightarrow B \\
\Rightarrow_{right} \\
\vdash A \lor (A \Rightarrow B) \\
\wedge_{right}
\end{array}
$$

2. $A \lor (B \land C) \Leftrightarrow (A \lor B) \land (A \lor C)$

$$
\begin{array}{c}
\text{ax} \quad A \vdash A; C \\
\wedge_{left} \\
\text{ax} \quad B; C \vdash A; C \\
\text{ax} \quad A \vdash A; B \\
\lor_{left} \\
\text{ax} \quad B; C \vdash A; B \\
\lor_{left} \\
\wedge_{left} \\
\wedge_{left} \\
\wedge_{right} \\
\leftrightarrow
\end{array}
$$

The proof of the other implication is similar.
3. \(((A \Rightarrow B) \Rightarrow A) \Rightarrow A\)

\[\begin{align*}
A & \vdash B; A \quad \text{ax} \\
\vdash A \Rightarrow B; A \quad \Rightarrow \text{right} \\
A & \vdash A \\
(\Rightarrow) & \vdash \Rightarrow \text{left} \\
\vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A \\
\Rightarrow \text{right}
\end{align*}\]

4. \(\neg \neg A \Rightarrow A\)

\[\begin{align*}
A & \vdash A \quad \text{ax} \\
\vdash A; \neg A \quad \neg \text{right} \\
\neg A & \vdash A \quad \neg \text{left} \\
\vdash \neg \neg A \Rightarrow A \\
\Rightarrow \text{right}
\end{align*}\]

5. \(\neg (A \lor B) \iff \neg A \land \neg B\)

\[\begin{align*}
A & \vdash B; A \quad \text{ax} \\
\vdash A \Rightarrow B; A \quad \Rightarrow \text{right} \\
A & \vdash A \quad \text{ax} \\
(\Rightarrow) & \vdash \Rightarrow \text{left} \\
\vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A \\
\Rightarrow \text{right}
\end{align*}\]

6. \(\neg \forall x. R(x) \Rightarrow \exists x. \neg R(x)\)

\[\begin{align*}
\neg R(x) & \vdash R(x) \quad \text{ax} \\
\vdash \neg R(x); \neg R(x) \quad \neg \text{right} \\
\vdash R(x); \exists x. \neg R(x) \quad \exists \text{right} \\
\vdash \forall x. R(x); \exists x. \neg R(x) \quad \forall \text{right} \\
\neg \forall x. R(x) & \vdash \exists x. \neg R(x) \quad \neg \text{left} \\
\vdash \neg \forall x. R(x) \Rightarrow \exists x. \neg R(x) \\
\Rightarrow \text{right}
\end{align*}\]

7. \(\forall x. (Q \lor R(x)) \Rightarrow (Q \lor \forall x. R(x))\)

\[\begin{align*}
Q & \vdash Q; R(x) \quad \text{ax} \\
\vdash R(x); Q; R(x) \quad \text{ax} \\
R(x) & \vdash Q; R(x) \\
Q & \lor R(x) \vdash Q; R(x) \quad \lor \text{left} \\
\forall x. (Q \lor R(x)) & \vdash Q; R(x) \quad \forall \text{left} \\
\forall x. (Q \lor R(x)) & \vdash \forall x. R(x) \quad \forall \text{right} \\
\forall x. (Q \lor R(x)) & \vdash Q \lor \forall x. R(x) \quad \forall \text{right} \\
\vdash \forall x. (Q \lor R(x)) \Rightarrow (Q \lor \forall x. R(x)) \\
\Rightarrow \text{right}
\end{align*}\]
8. \( \exists x. [(R(a) \lor R(b)) \Rightarrow R(x)] \)

\[
\begin{array}{c}
R(a); R(a) \lor R(b) \vdash R(a); R(b) \\
R(a) \lor R(b); R(a) \lor R(b) \vdash R(a); R(b) \\
R(a) \lor R(b) \vdash R(a); (R(a) \lor R(b)) \Rightarrow R(b) \\
\vdash (R(a) \lor R(b) \Rightarrow R(a); (R(a) \lor R(b)) \Rightarrow R(b) \\
\vdash \exists x. [(R(a) \lor R(b)) \Rightarrow R(x)]; \exists x. [(R(a) \lor R(b)) \Rightarrow R(x)] \\
\vdash \exists x. [(R(a) \lor R(b)) \Rightarrow R(x)]
\end{array}
\]

9. Show that if \( a \neq b \), there is no proof of 8. that does not use contractions.

Let us assume that there exists a contraction-free proof of 8. As the context is empty, the first rule to be applied has to be a right rule. The only right rule applicable is \( \exists right \). Let us call \( t \) the term used in this rule. We have a contraction-free proof of the sequent \( \Gamma \vdash (R(a) \lor R(b)) \Rightarrow R(t) \). Again, the only applicable rule is \( \Rightarrow right \), and we have a contraction-free proof of the sequent \( R(a) \lor R(b) \vdash R(t) \). Now, the only applicable rule is a left rule, that is \( \forall left \), giving contraction-free proofs of sequents \( R(a) \vdash R(t) \) and \( R(b) \vdash R(t) \). The only applicable rule is the axiom rule, so we deduce that \( t = a \) and \( t = b \), which is a contradiction.

Actually, there is no proof of 8. using no right contraction (but potentially left contractions). The syntactical proof is a bit heavier. When we will have proven the equivalence of sequent calculus with NK, we will be able to do an easier semantical proof: if \( a \neq b \), \( R(a) \lor R(b) \vdash R(t) \) is not valid.

**Exercise 2: Interpolation theorem**

If \( \phi \) is a formula, we call \( L(\phi) \) the set of free variables and function and predicate symbols appearing in \( \phi \). By extension, if \( \Gamma \) is a multiset of formulas, we write \( L(\Gamma) = \bigcup_{\phi \in \Gamma} L(\phi) \). We want to show that if \( L(\Gamma_1 \cup \Gamma_2 \cup \Delta_1 \cup \Delta_2) \) does not contain a function symbol and if \( \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2 \) is provable, then there exists a formula \( \xi \) such that:

- \( \Gamma_1 \vdash \xi, \Delta_1 \) and \( \Gamma_2, \xi \vdash \Delta_2 \) are provable;
- \( L(\xi) \subseteq L(\Gamma_1 \cup \Delta_1) \cap L(\Gamma_2 \cup \Delta_2) \)

1. Prove this result. You will consider the following cases in detail: \( ax \); \( \Rightarrow right \); \( \Rightarrow left \); \( \forall left \); \( \forall left \).

We proceed by induction on the proof of the sequent \( \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2 \).

**ax:** There exists \( \phi \in (\Gamma_1 \cup \Gamma_2) \cap (\Delta_1 \cup \Delta_2) \). There are four cases: if

- \( \phi \in \Gamma_1 \cap \Delta_1 \), we take \( \xi = \bot \),
- \( \phi \in \Gamma_2 \cap \Delta_2 \), we take \( \xi = T \),
- \( \phi \in \Gamma_2 \cap \Delta_1 \), we take \( \xi = \neg \phi \),
- \( \phi \in \Gamma_1 \cap \Delta_2 \), we take \( \xi = \phi \).
⇒_{right}: There exists \( \phi = \phi_1 \Rightarrow \phi'_1 \in \Delta_1 \cup \Delta_2 \). Let us assume that \( \Delta_1 = \phi, \Delta'_1 \). We have a proof of the sequent \((\Gamma_1, \phi_1), \Gamma_2 \vdash (\Delta'_1, \phi'_1), \Delta_2 \). By induction hypothesis, we have \( \xi \) such that

- \( \Gamma_1, \phi_1 \vdash \xi, \phi'_1, \Delta'_1 \) and \( \Gamma_2, \xi \vdash \Delta_2 \) are provable;
- \( L(\xi) \subseteq L(\Gamma_1 \cup \Delta'_1 \cup \{ \phi_1, \phi'_1 \}) \cap L(\Gamma_2 \cup \Delta_2) \)
  \[= L(\Gamma_1 \cup \Delta_1) \cap L(\Gamma_2 \cup \Delta_2) \]

We take formula \( \xi \). The case \( \Delta_2 = \phi, \Delta'_2 \) is symmetric.

⇒_{left}: There exists \( \phi = \phi_1 \Rightarrow \phi'_1 \in \Gamma_1 \cup \Gamma_2 \). Let us assume that \( \Gamma_1 = \phi, \Gamma'_1 \). We have proofs of the sequents \( \Gamma_1, \Gamma_2 \vdash (\phi_1, \Delta_1), \Delta_2 \) and \( (\Gamma'_1, \phi'_1), \Gamma_2 \vdash \Delta_1, \Delta_2 \). By induction hypothesis, we have \( \xi_1 \) and \( \xi'_1 \) such that

- \( \Gamma'_1 \vdash \xi_1, \phi_1, \Delta_1 \) and \( \Gamma_2, \xi_1 \vdash \Delta_2 \) are provable;
- \( \Gamma'_1, \phi'_1 \vdash \xi'_1, \Delta_1 \) and \( \Gamma_2, \xi'_1 \vdash \Delta_2 \) are provable;
- \( L(\xi_1) \subseteq L(\Gamma_1 \cup \Delta'_1 \cup \{ \phi_1 \}) \cap L(\Gamma_2 \cup \Delta_2) \)
  \[\subseteq L(\Gamma_1 \cup \Delta_1) \cap L(\Gamma_2 \cup \Delta_2) \]
- \( L(\xi'_1) \subseteq L(\Gamma_1 \cup \Delta'_1 \cup \{ \phi'_1 \}) \cap L(\Gamma_2 \cup \Delta_2) \)
  \[\subseteq L(\Gamma_1 \cup \Delta_1) \cap L(\Gamma_2 \cup \Delta_2) \]

We define \( \xi = \xi_1 \lor \xi'_1 \). The case \( \Gamma_2 = \phi, \Gamma'_2 \) is dual: after applying the induction hypothesis over sequents \( \Gamma_1, \Gamma'_1 \vdash (\phi_1, \Delta_1) \) and \( \Gamma_1, (\phi'_1, \Gamma'_2) \vdash \Delta_1, \Delta_2 \), yielding two formulas \( \xi_2 \) and \( \xi'_2 \), we define \( \xi = \xi_2 \lor \xi'_2 \).

∀_{right}: There exists \( \phi = \forall x. \phi' \in \Delta_1 \cup \Delta_2 \), and \( x \notin fv(\Gamma_1, \Gamma_2, \Delta'_1, \Delta_2) \).

Wlog, \( \Delta_1 = \phi, \Delta'_1 \). We have a proof of \( \Gamma_1, \Gamma_2 \vdash (\phi', \Delta'_1), \Delta_2 \). By induction hypothesis, we have \( \xi \) such that

- \( \Gamma_1 \vdash \xi, \phi', \Delta_1 \) and \( \Gamma_2, \xi \vdash \Delta_2 \) are provable;
- \( L(\xi) \subseteq L(\Gamma_1 \cup \Delta_1 \cup \{ \phi' \}) \cap L(\Gamma_2 \cup \Delta_2) \)
  \[\subseteq [L(\Gamma_1 \cup \Delta_1) \cup \{ x \}] \cap L(\Gamma_2 \cup \Delta_2) \]
  \[= L(\Gamma_1 \cup \Delta_1) \cap L(\Gamma_2 \cup \Delta_2) \]

We choose directly formula \( \xi \).

∀_{left}: There exists \( \phi = \forall x. \phi' \in \Gamma_1 \cup \Gamma_2 \). Let us assume that \( \Gamma_1 = \phi, \Gamma'_1 \). We have a proof of the sequent \( (\Gamma'_1, \phi'\{x \rightarrow t\}), \Gamma_2 \vdash \Delta_1, \Delta_2 \) for a term \( t \). As there is no function symbol, \( t = y \in X \). By induction hypothesis, we have \( \xi' \) such that

- \( \Gamma'_1, \phi'\{x \rightarrow y\} \vdash \xi', \Delta_1 \) and \( \Gamma_2, \xi' \vdash \Delta_2 \) are provable;
- \( L(\xi_1) \subseteq [L(\Gamma'_1 \cup \Delta_1 \cup \{ \phi', y \}) \cup \{ x \}] \cap L(\Gamma_2 \cup \Delta_2) \)
  \[\subseteq L(\Gamma_1 \cup \Delta_1 \cup \{ y \}) \cap L(\Gamma_2 \cup \Delta_2) \]

There are three cases:
• \( y \in L(\Gamma_1, \Delta_1) \cap L(\Gamma_2, \Delta_2) \): we take \( \xi = \xi' \).
• \( y \notin L(\Gamma_2, \Delta_2) \): then \( y \notin fv(\xi) \), so we can take \( \xi = \xi' \).
• \( y \notin L(\Gamma_1, \Delta_1) \): then \( y \) can appear free in \( \xi \) even though it is not in \( L(\Gamma_1 \cup \Delta_1) \cap L(\Gamma_2 \cup \Delta_2) \) so we take \( \xi = \forall y.\xi' \).

Dually, if \( \Gamma_2 = \phi, \Gamma'_2 \), we apply the induction hypothesis to \( \Gamma'_1, (\phi'\{x \rightarrow t\}, \Gamma_2) \vdash \Delta_1, \Delta_2 \), and take either the obtained formula \( \xi' \) or \( \xi = \exists y.\xi' \).

2. Prove the interpolation theorem: if \( \phi \) and \( \psi \) are formulas that do not contain function symbols and if \( \vdash \phi \Rightarrow \psi \) is provable, then there exists a formula \( \xi \) such that:
   • \( \vdash \phi \Rightarrow \xi \) and \( \vdash \xi \Rightarrow \psi \) are provable;
   • \( L(\xi) \subseteq L(\phi) \cap L(\psi) \).

   If we have a proof of \( \vdash \phi \Rightarrow \psi \), we have a proof of \( \phi \vdash \psi \). We fix \( \Gamma_2 = \Delta_1 = \emptyset, \Gamma_1 = \phi, \) and \( \Delta_2 = \psi \). By the previous result, there exists \( \xi \) such that \( L(\xi) \subseteq L(\phi) \cap L(\psi) \), and \( \phi \vdash \xi \) and \( \xi \vdash \psi \) are provable. We can conclude, building the desired proofs with \( \Rightarrow_{\text{right}} \) and these last two proofs.

3. \((\star)\) We apply the interpolation theorem to prove the Beth theorem.

   Let \( P \) and \( P' \) be two unary predicates. Let \( \Gamma(P) \) be a set of closed formulas that do not contain the symbol \( P' \). We write \( \Gamma(P') \) the set of formulas generated by replacing the symbol \( P \) by the symbol \( P' \) in \( \Gamma(P) \).

   We say that \( \Gamma(P) \) implicitly defines \( P \) if \( \Gamma(P), \Gamma(P') \vdash \forall x. (P(x) \Leftrightarrow P'(x)) \) is provable; we say that \( \Gamma(P) \) explicitly defines \( P \) if there exists a formula \( \phi(x) \) using neither \( P \) nor \( P' \) such that \( \Gamma(P) \vdash \forall x. (A(x) \Leftrightarrow P'(x)) \).

   Prove that \( \Gamma(P) \) implicitly defines \( P \) iff \( \Gamma(P) \) explicitly defines \( P \).

   Hint: first show by induction that \( \Gamma(P) \vdash \Delta(P) \) implies \( \Gamma(P') \vdash \Delta(P') \).

   One implication is direct. For the other, use the interpolation theorem.

4. \((\star\star)\) In fact, the interpolation theorem holds even when \( \phi \) and \( \psi \) contain function symbols. Can you see how to use the previous result to treat this case?

   Hint: represent function symbols using predicate symbols.
2 Unification

A unification problem is a set $E$ of equations of the form $t \overset{?}{=} u$. A unifier (i.e. solution to the unification problem) of a set $E$ is a substitution $\sigma$ such that for every equation $t \overset{?}{=} u$ in $E$, $t\sigma = u\sigma$.

The unification procedure seen in class are reminded at the end of the exercise sheet (Algorithm 1).

Exercise 3: Some examples

Apply the procedure to the following unification problems (your answer should either be fail or the substitution returned by the procedure):

- $E_1 = f(x, g(a, y)) \overset{?}{=} f(h(y), g(y, a)); g(x, h(y)) \overset{?}{=} g(z, z)$

\[
\text{Unif}(E_1) = \begin{align*}
\text{Unif}(f(x, g(a, y)) \overset{?}{=} f(h(y), g(y, a)); g(x, h(y)) \overset{?}{=} g(z, z)) & = \\
\text{Unif}(x \overset{?}{=} h(y); g(a, y) \overset{?}{=} g(y, a); g(x, h(y)) \overset{?}{=} g(z, z)) & = \\
\text{Unif}(x \overset{?}{=} h(y); y \overset{?}{=} a; g(x, h(y)) \overset{?}{=} g(z, z)) & = \\
\text{Unif}(x \overset{?}{=} h(y); y \overset{?}{=} a; x \overset{?}{=} z; h(y) \overset{?}{=} z) & = \\
\text{Unif}(z \overset{?}{=} h(y); y \overset{?}{=} a)\{x \mapsto z\} & = \\
\text{Unif}(z \overset{?}{=} h(a); y \overset{?}{=} a)\{x \mapsto z; y \mapsto a\} & = \\
\text{Unif}(z \overset{?}{=} h(a))\{x \mapsto z; y \mapsto a\} & = \\
\text{Unif}(h(a) \overset{?}{=} h(a))\{x \mapsto h(a); y \mapsto a; z \mapsto h(a)\} & = \\
\text{Unif}(a \overset{?}{=} a)\{x \mapsto z; y \mapsto a; z \mapsto h(a)\} & = \\
\{x \mapsto h(a); y \mapsto a; z \mapsto h(a)\} & =
\end{align*}
\]

- $E_2 = f(x, x) \overset{?}{=} f(g(y), z); h(z) \overset{?}{=} h(y)$

\[
\text{Unif}(E_2) = \begin{align*}
\text{Unif}(f(x, x) \overset{?}{=} f(g(y), z); h(z) \overset{?}{=} h(y)) & = \\
\text{Unif}(x \overset{?}{=} g(y); x \overset{?}{=} z; h(z) \overset{?}{=} h(y)) & = \\
\text{Unif}(g(y) \overset{?}{=} z; h(z) \overset{?}{=} h(y))\{x \mapsto g(y)\} & = \\
\text{Unif}(h(g(y)) \overset{?}{=} h(y))\{x \mapsto g(y); z \mapsto g(y)\} & = \\
\text{Unif}(g(y) \overset{?}{=} y)\{x \mapsto g(y); z \mapsto g(y)\} & = \\
fail & =
\end{align*}
\]

- $E_3 = f(x, a) \overset{?}{=} f(b, y); f(x) \overset{?}{=} f(y)$

\[
\text{Unif}(E_3) = \begin{align*}
\text{Unif}(f(x, a) \overset{?}{=} f(b, y); f(x) \overset{?}{=} f(y)) & = \\
\end{align*}
\]
Exercise 4: Studying the unification algorithm

We study the properties of the unification algorithm $\text{Unif}$.

1. Is the procedure deterministic?

   No: if the problem contains multiple equations, the procedure has the choice of which equation to treat first.

2. Show that it always terminates (in failure or by returning a substitution).

   At each step, either the procedure fails, either the number of variables in the problem diminishes, either the number of variables stays the same but the sum of all sizes of terms in the problem diminishes.

3. We now want to prove that it indeed calculates a unifier of the set of equations $E$ given as entry. We will in fact prove a more general result.

   (a) Show that if $E$ contains $f(u_1,\ldots,u_n) \equiv g(t_1,\ldots,t_m)$, where $f \neq g$, then it is not unifiable.

      For every $\sigma$ we have $f(u_1,\ldots,u_n)\sigma \neq g(t_1,\ldots,t_m)\sigma$, so the unification problem $E$ has no answers.

   (b) Show that if $E$ contains $x \equiv t$, where $x \in \mathcal{X}$, $v \notin \mathcal{X}$ and $x \in \text{Var}(t)$, then it is not unifiable.

      For every $\sigma$ we have $|t\sigma| < |u\sigma|$. The problem $E$ is not unifiable.

   (c) Show that the unification problem $x \equiv f(y)$ has an infinity of unifiers (you can use a constant, i.e. 0-ary, function symbol $a$).

      Is there a most general one? When it exists, is there unicity of the mgu?
(d) Show that if \( E = E' \cup \{ x \equiv t \} \), then \( \sigma \) unifies \( E \) iff it unifies \( E' \).

Observe that this implies that \( \sigma \) is a mgu of \( E \) if it is a mgu of \( E' \).

For every \( \sigma \), \( x\sigma = x\sigma \), so it unifies \( E' \) iff it also unifies \( E \).

(e) Show that if \( E = E' \cup \{ f(u_1, \ldots, u_n) \equiv (t_1, \ldots, t_n) \} \), then \( \sigma \) unifies \( E \) iff it unifies \( E' \cup \{ u_1 \equiv t_1, \ldots, u_n \equiv t_n \} \).

Observe that this implies that \( \sigma \) is a mgu of \( E \) if it is a mgu of \( E' \).

For every \( \sigma, t_i\sigma = u_i\sigma \) for all \( i \in \{ 1, \ldots, n \} \) iff \( t\sigma = f(u_1\sigma, \ldots, u_n\sigma) = f(t_1\sigma, \ldots, t_n\sigma) = u\sigma \). \( \sigma \) is a solution to \( E \) iff it is a solution to this new unification problem.

(f) Show that for every substitution \( \sigma \), variable \( x \) and term \( t \), if \( x\sigma = t\sigma \) then \( \sigma = \sigma \circ \{ x \mapsto t \} \).

The proof is a straightforward induction on terms.

(g) Show that if \( E = E' \cup \{ x \equiv t \} \) where \( x \notin \text{var}(t) \), then \( \sigma \) unifies \( E'\{ x \mapsto t \} \) iff \( \sigma \circ \{ x \mapsto t \} \) unifies \( E \).

Prove using the previous question that: \( \sigma \) is a mgu of \( E'\{ x \mapsto t \} \) implies \( \sigma \circ \{ x \mapsto t \} \) is a mgu of \( E \).

First, by a straightforward induction on terms, we can prove that for every \( u \), \( (u\{ x \mapsto t \})\sigma = u(\sigma \circ \{ x \mapsto t \}) \). This result implies that \( \sigma \) unifies \( E'\{ x \mapsto t \} \) iff \( \sigma \circ \{ x \mapsto t \} \) unifies \( E' \). What’s more, \( \sigma \circ \{ x \mapsto t \} \) always unifies \( x \equiv t \), so the equivalence is proven.

Let us assume that a substitution \( \tau \) is a unifier for \( E \), and \( \sigma \) is a mgu of \( E'\{ x \mapsto t \} \). Observe that \( \tau \) also unifies \( E'\{ x \mapsto t \} \). By the previous equivalence and the fact that \( \sigma \) is a mgu, we have the existence of \( \eta \) such that \( \tau = \eta \circ \sigma \). By precomposing this equation by \( \{ x \mapsto t \} \), we have the equality \( \tau \circ \{ x \mapsto t \} = \eta \circ \sigma \circ \{ x \mapsto t \} \). As a consequence, there exists \( \eta \) such that \( \tau = \eta \circ (\sigma \circ \{ x \mapsto t \}) \).

(h) Show that if on input \( E \) the algorithm

- returns a substitution \( \sigma \), then \( \sigma \) is a mgu for \( E \);
- fails, then the unification problem \( E \) has no solution.

We proceed by induction on the size of the problem \( E \).

- If \( E = \emptyset \), then \( id \) is a unifier of \( E \).

- Else, \( E \) contains at least one equation \( t \equiv u \). We define \( E' = E \setminus \{ t \equiv u \} \).
– If \( t \equiv u \) makes the procedure fail, by questions b) and c), 
\( E \) is not unifiable, and the algorithm fails.
– In all other cases, we use the induction hypothesis on the 
recursive call of \texttt{Unif}. If it fails, \( E \) is not unifiable. Else, 
by questions e), f), and g), the returned substitution is a 
mgu for \( E \).

3 Additional exercises (⋆)

Exercise 5: A new rule
In this exercise we introduce the cut rule to the sequent calculus:

\[
\frac{\Gamma \vdash \psi \quad \Gamma, \psi \vdash \phi}{\Gamma \vdash \phi} \text{ cut}
\]

Give two proofs in sequent of \( A \Rightarrow B, A \Rightarrow C, B \land C \Rightarrow D \vdash A \Rightarrow D \): a first
with cuts, a second without cuts.

Exercise 6: Return to natural deduction
This exercise in an introduction to the proof a equivalence of natural deduction
and sequent calculus.

1. Show that a proof in natural deduction of a sequent \( \Gamma, \phi \land \psi \vdash \xi \) can be
transformed in a proof in natural deduction of the sequent \( \Gamma, \phi, \psi \vdash \xi \).
2. You can think of how to do the same transformation for other left rules.
The unification algorithm

The unification procedure is the following:

**Algorithm 1: Unif**

- **Input**: a unification problem $E$
- if $E = E' \cup \{ f(u_1, ..., u_n) \overset{?}{=} f(t_1, ..., t_n) \}$ then
  - $\text{Unif}(E' \cup \{ u_1 \overset{?}{=} t_1, ..., u_n \overset{?}{=} t_n \})$
- else if $E = E' \cup \{ f(u_1, ..., u_n) \overset{?}{=} g(t_1, ..., t_m) \}$ where $f \neq g$ then
  - fail
- else if $E = E' \cup \{ x \overset{?}{=} x \}$ then
  - $\text{Unif}(E')$
- else if $E = E' \cup \{ x \overset{?}{=} t \}$ where $x \in \mathcal{X}$ and $x \notin \text{Var}(t)$ then
  - $\text{Unif}(E' \{ x \mapsto t \}) \circ \{ x \mapsto t \}$
- else if $E = E' \cup \{ x \overset{?}{=} t \}$ where $x \in \mathcal{X}$, $t \notin \mathcal{X}$ and $x \in \text{Var}(t)$ then
  - fail
- else if $E = \emptyset$ then
  - $\text{id}$