Problem 3.1

Apply the unification algorithm and show the most general unifier of the following formulas:

1. $p(f(x), f(y), y)$ and $p(y, z, f(x))$;
2. $p(g(x, y), g(y, g(z, b)))$ and $p(z, g(a, x))$.

Note: x, y, z denote variables, f, g are function symbols, p is a predicate symbol and a, b are constants.

Sub-Problem 3.1.1 Starting with the set of equations $\{p(f(x), f(y), y) = p(y, z, f(x))\}$

- The set of equations is:

$$\{\ p(f(x), f(y), y) = p(y, z, f(x))\ \}$$

There exists a non-isolated equation:

$$p(f(x), f(y), y) = p(y, z, f(x))$$

Apply: $(f(s_1, \ldots, s_n), f(t_1, \ldots, t_n)) \Rightarrow s_1 = t_1, \ldots, s_n = t_n$

- The set of equations is:

$$\{\ f(x) = y,\ f(y) = z,\ y = f(x)\ \}$$

There exists a non-isolated equation:

$$f(x) = y$$

Apply: $(t, x) \Rightarrow (x, t)$ and subsequently replace x by t in all other equations of E.

- The set of equations is:

$$\{\ y = f(x),\ f(f(x)) = z,\ f(x) = f(x)\ \}$$

There exists a non-isolated equation:

$$f(f(x)) = z$$

Apply:$(t, x) \Rightarrow (x, t)$ and subsequently replace x by t in all other equations of E.

- The set of equations is:

$$\{\ y = f(x),\ z = f(f(x)),\ f(x) = f(x)\ \}$$

  There exists a non-isolated equation:

$$f(x) = f(x))$$

  Apply: $(t,t) \Rightarrow$ remove this equations from E.

- The set of equations is:

$$\{\ y = f(x),\ z = f(f(x))\ \}$$

  All equations are isolated equation. Terminate and return

$$\Theta := \{\ y \mapsto f(x),\ z \mapsto f(f(x))\ \}$$

Finally, to be sure whether the substitution is actually a unifying substitution

$$E\Theta = \{p(f(x), f(f(x)), f(x)) = p(f(x), f(f(x)), f(x))\}$$

Sub-Problem 3.1.2 Starting with the set of equations $\{p(g(x,y), g(y, g(z,b))) = p(z, g(a,x))\}$

- The set of equations is:

$$\{\ p(g(x,y), g(y, g(z,b))) = p(z, g(a,x))\ \}$$

  There exists a non-isolated equation:

$$p(g(x,y), g(y, g(z,b))) = p(z, g(a,x))$$

  Apply: $(f(s_1,\ldots,s_n), f(t_1,\ldots,t_n)) \Rightarrow s_1 = t_1, \ldots, s_n = t_n$

- The set of equations is:

$$\{\ g(x,y) = z,\ g(y, g(z,b)) = g(a,x)\ \}$$

  There exists a non-isolated equation:

$$g(x,y) = z$$

  Apply: $(t,x) \Rightarrow (x,t)$ and subsequently replace x by t in all other equations of E.

- The set of equations is:

$$\{ \ z = g(x, y), \ g(y, g(g(x, y), b)) = g(a, x) \ \}$$

There exists a non-isolated equation:

$$g(y, g(g(x, y), b)) = g(a, x)$$

Apply: $(f(s_1, \ldots, s_n), f(t_1, \ldots, t_n)) \Rightarrow s_1 = t_1, \ldots, s_n = t_n$

- The set of equations is:

$$\{ \ z = g(x, y), \ y = a, \ g(g(x, y), b) = x \ \}$$

There exists a non-isolated equation:

$$y = a$$

Apply: $(x, t) \Rightarrow$ replace x by t in all other equations of E.

- The set of equations is:

$$\{ \ z = g(x, y), \ y = a, \ g(g(x, a)), b) = x \ \}$$

There exists a non-isolated equation:

$$g(g(x, a)), b) = x$$

Apply: $(t, x) \Rightarrow (x, t)$ and observe that x occurs in t.

Halt with failure.

Problem 3.2

Consider an ordering $\succ$ on ground non-equality atoms that is total and well- founded. We denote the literal ordering induced by $\succ$ also by $\succ$. Let C and D be ground clauses without equality literals. Let A and B respectively denote the maximal atoms of C and D wrt $\succ$. Assume that A and B are syntactically the same atoms. Assume also that A occurs negatively in C but only positively in D. Show that $C \succ_{bag} D$.

Firstly, we know that for a strict ordering $>$ over X, a bag extension $>_{bag}$ is defined as the smallest transitive relation satisfying

$$\{x, y_1, \ldots, y_n\} >_{bag} \{x_1, \ldots, x_m, y_1, \ldots, y_n\} \quad if \ \forall i \in \{1, \ldots, m\} : x > x_i$$

Secondly, we know that $>$ is total and well-founded over ground non-equality atoms. Moreover, we know that the extension of such an ordering is also total and well-founded, and satisfies $\neg p > p$ as well as if $p > q$ then $\neg p > p > \neg q > q$. Thirdly, we know that a clause can be considered as a bag of literals, i.e.

$$C = \{\neg A, \overline{x}, \overline{y}\} \qquad\qquad D = \{A, \overline{z}, \overline{y}\}$$

(Note: $\overline{a} = (a_i)_{i \in I}$ for some $I \subset \mathbb{N}$) Since, A is the maximal atom in the bag, any atom must be smaller than or syntactically equal to A. Given the extension of $>$ to literals, it follows that $\neg A$ is a maximal literal. As for D, since A is a maximal atom and only occurs positively in D. It follows, by the extension of the ordering to literals, that $\forall l \in \overline{z} \cup \overline{y}$ either $l = A$ (syntactically) or $A > l$.
Fourthly, given the fact that $\forall l \in \overline{x} \cup \overline{y}\ \neg A > l$, it follows that

$$\{\neg A, \overline{y}\} >_{bag} \{\overline{x}, \overline{y}\}$$

and especially

$$\{\neg A, \overline{y}\} >_{bag} \{A, \overline{x}, \overline{y}\}$$

Moreover, it is clearly the case that the ordering of the bags $\{\neg A, \overline{y}\}$ and $\{\neg A, \overline{x}, \overline{y}\}$ is based on the structure of $\overline{x}$. That is, if $\overline{x}$ is empty, they are the same bags. Otherwise, it must be the case that

$$\{\neg A, \overline{x}, \overline{y}\} >_{bag} \{\neg A, \overline{y}\}$$

That is, apart from the elements in $\overline{x}$ both bags share the same literals. Now, since $\forall l \in D\ \neg A > l$, this holds especially for $\neg A > A$ and $\forall l \in \overline{z}\ \neg A > l$. Hence,

$$\{\neg A, \overline{y}\} >_{bag} \{A, \overline{z}, \overline{y}\}$$

with due to transitivity leads to

$$\{\neg A, \overline{x}, \overline{y}\} >_{bag} \{A, \overline{z}, \overline{y}\}$$

Problem 3.3

> Let Σ be a signature containing only function symbols such that Σ contains at least one constant. Let $\gg$ be a precedence relation on Σ and $w : \Sigma \to \mathbb{N}$ be a weight function compatible with $\gg$. Consider the (ground) Knuth-Bendix order $>_{KB}$ induced by $\gg$ and w on the set of ground terms of Σ. Describe the set of ground terms that have the minimal weight wrt $>_{KB}$.

Some notational clarifications. Let $\overline{a} = (a_i)_{i \in \{1, \ldots n\}}$ if n it is clear from the context, e.g. as input for a function of arity n.

Firstly, we know that there exists at least one constant c. Hence, we know that there exists a minimal constant. Assume without loss of generality that one of those minimal constant is c_{min}. Since it is possible for constant symbols to have the same weight let

$$\Gamma_C \coloneqq \{c \mid c \in FS_0\ w(c) = w(c_{min})\}$$

Secondly, we need to show the statement, for any $k > 1$ and an arbitrary tuple of terms $\overline{a}$ it follows that $\forall f \in FS_k \forall a_i \in \overline{a}\ |f(\overline{a})| > |a_i|$. This follows from the fact that the input to f must contain at least two constant symbols, and the fact that w is compatible with $\gg$, requiring $\forall c \in FS_0\ w(c) > 0$. That is, $|f(\overline{a})| = w(f) + \sum_{a_i \in \overline{a}} |a_i| > w(c_1) + w(c_2)$ where c_1, c_2 are constants in $a_1, a_2 \in \overline{a}$. If one wants to be precise this would require induction.

Thirdly, we need to show the statement, for an arbitrary term a it follows that $\forall f \in FS_1\ w(f) > 0 \Leftrightarrow |f(a)| > |a|$. If $w(f) > 0$ then $|f(a)| = w(f) + |a| > |a|$. If $|f(a)| > |a|$ then $|f(a)| = w(f) + w(a) > w(a)$ requires $w(f) > 0$.

Fourthly, we need to show the statement, for an arbitrary term a and for an $n > 0$ it follows that $\forall f \in FS_1\ w(f) = 0 \Leftrightarrow |f^n(a)| = |a|$. Clearly, $|f(a)| = n \cdot w(f) + |a| = |a|$.

From the second statement, it is clearly the case that all non-unary functions are excluded from the set of ground terms with minimal weight. From the third statement, one can conclude that the set of minimal ground terms, can not contain terms with a function symbol of a weight greater 0. Furthermore, since w is consistent with $\gg$ we know just one single function symbol can satisfy this condition. Henceforth, let f_{min} be this unique function symbol with $w(f_{min}) = 0$. Moreover, from the fourth statement if follows that one can have arbitrary iterations of this function symbol f_{min}, while at the same time retaining the same weight as the term to which this sequence of functions symbols is applied to. Moreover, with the weight of the $f_{min}^k(a)$ being the same as the weight of a, it clearly follows that any term of the form $f_{min}^k(a)$ can only be minimal if and only if a has minimal weight to begin with. Those, insights are sufficient enough to define the set

$$\Gamma_F \coloneqq \{f_{min}^k(c) \mid c \in \Gamma_C \land k > 0\}$$

In the final step of the characterisation, a case distinction is required. If f_{min} does not exists, given the reasoning above, there can not be a term with a function symbol of arity greater that 0 in the set of weight minimal ground terms, i.e. only the weight minimal constant symbols are permitted. Otherwise, arbitrary applications of f_{min} to constant symbols of minimal weight are permitted as well. That is,

$$\begin{array}{ll} \Gamma := \Gamma_C \cup \Gamma_F & \textit{if } \exists f_{min} \in FS_1 f_{min} = 0 \\ \Gamma := \Gamma_C & otw.0 \end{array}$$

Problem 3.4

Consider the following set S of clauses:

$$\begin{array}{l} \neg p(z,a) \vee \neg p(z,x) \vee \neg p(x,z), \\ p(y,a) \vee p(y,f(y)), \\ p(w,a) \vee p(f(w),w) \end{array}$$

where p is a predicate symbol, f is a function symbol, x, y, z, w are variables and a is a constant. Give a refutation proof of S by using the non-ground binary resolution inference system $\mathbb{BR}$. For each newly derived clause, label the clauses from which it was derived by which inference rule and indicate most general unifiers.

The required numbering is given in the summary below. The reasoning behind those inferences, however, is presented beforehand. First, we need to specify a precedence ordering

$$f \gg p \gg a \gg w \gg x \gg y \gg z$$

and a weight function w that is compatible with the precedence ordering. In this case let w be the constant function 1.
Consider the clause $\neg p(z,a) \vee \neg p(z,x) \vee \neg p(x,z)$ and the clause $p(y,a) \vee p(y,f(y))$. In the prior, only negative literals exist. Hence, given the fact that our selection function is well-behaved we can choose an arbitrary negative literal or have to choose all maximal positive literals. In this case $\neg p(x,z)$ is chosen at random. As for the latter, given the fact that $\#(y, p(y,f(y))) = 2 \geq 1 = \#(y, p(y,a))$ and $|p(y,f(y))| = 1+1+1+1 = 4 \geq 3 = 1+1+1 = |p(y,a)|$ it follows that $p(y,f(y)) \succ_{KB} p(y,a)$. Hence,

$$\cfrac{\cfrac{\neg p(z,a) \vee \neg p(z,x) \vee \underline{\neg p(x,z)} \qquad p(y,a) \vee \underline{p(y,f(y))}}{(\neg p(z,a) \vee \neg p(z,x) \vee p(y,a))\{z \mapsto f(y), x \mapsto y\}}\text{BR}}{\neg p(f(y),a) \vee \neg p(f(y),y) \vee p(y,a)}\,\textit{sub.}$$

(Note: The last step is not part of the calculus, it sole purpose is to make the substitution with the mgu more explicit.)
Consider the clause $\neg p(f(y),a) \vee \neg p(f(y),y) \vee p(y,a)$ and the clause $p(w,a) \vee p(f(w),w)$. In the prior, some negative literals exists. Now, given the fact that our selection function is well-behaved we can choose an arbitrary

negative literal or have to choose all maximal positive literals, the literal $\neg p(f(y), y)$ is chosen at random. As for the latter, given the fact that $\#(w, p(f(w), w)) = 2 \geq 1 = \#(w, p(w, a))$ and $|p(f(w), w)| = 1 + 1 + 1 + 1 = 4 \geq 3 = 1 + 1 + 1 = |p(w, a)|$ it follows that $p(f(w), w) \succ_{KB} p(w, a)$. Hence,

$$\cfrac{\cfrac{\neg p(f(y), a) \vee \underline{\neg p(f(y), y)} \vee p(y, a) \qquad p(w, a) \vee \underline{p(f(w), w)}}{(\neg p(f(y), a) \vee p(y, a) \vee p(w, a))\{w \mapsto y\}}\ \text{BR}}{\neg p(f(y), a) \vee p(y, a) \vee p(y, a)}\ \textit{sub.}$$

The resulting clause can be reduced, by positive factoring. That is, in this case the selection function chooses all maximal positive literals. Since, both literals are syntactically identical they are incomparable with respect to the KBO. Moreover, being the only two positive literals, it follows that they are maximal.

$$\frac{\neg p(f(y), a) \vee \underline{p(y, a)} \vee \underline{p(y, a)}}{\neg p(f(y), a) \vee p(y, a)}\ \text{Fact}$$

Consider the clause $\neg p(f(y), a) \vee p(y, a)$ and the clause $p(w, a) \vee p(f(w), w)$. In the prior, some negative literals exists. Now, given the fact that our selection function is well-behaved we can choose an arbitrary negative literal or have to choose all maximal positive literals, the literal $\neg p(f(y), a)$ is chosen at random. As for the latter, given the fact that $\#(w, p(f(w), w)) = 2 \geq 1 = \#(w, p(w, a))$ and $|p(f(w), w)| = 1+1+1+1 = 4 \geq 3 = 1+1+1 = |p(w, a)|$ it follows that $p(f(w), w) \succ_{KB} p(w, a)$. Hence,

$$\cfrac{\cfrac{\underline{\neg p(f(y), a)} \vee p(y, a) \qquad p(w, a) \vee \underline{p(f(w), w)}}{(p(y, a) \vee p(w, a))\{w \mapsto a, y \mapsto a\}}\ \text{BR}}{p(a, a) \vee p(a, a)}\ \textit{sub.}$$

The resulting clause can be reduced, by positive factoring. That is, in this case the selection function chooses all maximal positive literals. Since, both literals are syntactically identical they are incomparable with respect to the KBO. Moreover, being the only two positive literals, it follows that they are maximal.

$$\frac{\underline{p(a, a)} \vee \underline{p(a, a)}}{p(a, a)}\ \text{Fact}$$

The clause $\neg p(z, a) \vee \neg p(z, x) \vee \neg p(x, z)$ can be reduced, by negative factoring. That is, in this case the selection function chooses two negative literals.

$$\cfrac{\cfrac{\neg p(z, a) \vee \underline{\neg p(z, x)} \vee \underline{\neg p(x, z)}}{(\neg p(z, a) \vee \neg p(z, x))\{z \mapsto x\}}\ \text{Fact}}{\neg p(x, a) \vee \neg p(x, x)}\ \textit{sub.}$$

The clause $\neg p(x,a) \vee \neg p(x,x)$ can be reduced, by negative factoring. That is, in this case the selection function chooses two negative literals.

$$\dfrac{\dfrac{\underline{\neg p(x,a)} \vee \underline{\neg p(x,x)}}{(\neg p(x,a))\{x \mapsto a\}}\text{ Fact}}{\neg p(a,a)}\textit{ sub.}$$

Consider the clause $\neg p(a,a)$ and the clause $p(a,a)$. Since they are the only literals in the clause no other literals could be selected.

$$\dfrac{\dfrac{\underline{\neg p(a,a)} \qquad \underline{p(a,a)}}{(\square)\{\,\}}\text{ BR}}{\square}\textit{ sub.}$$

To summarise.

(1)	$\neg p(z,a) \vee \neg p(z,x) \vee \neg p(x,z)$	
(2)	$p(y,a) \vee p(y,f(y))$	
(3)	$p(w,a) \vee p(f(w),w)$	
$(1),(2) \overset{BR}{\Longrightarrow} (4)$	$\neg p(f(y),a) \vee \neg p(f(y),y) \vee p(y,a)$	$\{z \mapsto f(y), x \mapsto y\}$
$(4),(3) \overset{BR}{\Longrightarrow} (5)$	$\neg p(f(y),a) \vee p(y,a) \vee p(y,a)$	$\{w \mapsto y\}$
$(5) \overset{Fact.}{\Longrightarrow} (6)$	$\neg p(f(y),a) \vee p(y,a)$	$\{\}$
$(6),(3) \overset{BR}{\Longrightarrow} (7)$	$p(a,a) \vee p(a,a)$	$\{w \mapsto a, y \mapsto a\}$
$(7) \overset{Fact.}{\Longrightarrow} (8)$	$p(a,a)$	$\{\}$
$(1) \overset{Fact.}{\Longrightarrow} (9)$	$\neg p(x,a) \vee \neg p(x,x)$	$\{z \mapsto x\}$
$(9) \overset{Fact.}{\Longrightarrow} (10)$	$\neg p(a,a)$	$\{x \mapsto a\}$
$(10),(7) \overset{BR}{\Longrightarrow} (11)$	$\square$	$\{\}$

Problem 3.5

Consider the KBO ordering $\succ$ generated by the precedence $f \gg a \gg b \gg c$ and the weight function that assigns weight 1 to each symbol from $\{f, a, b, c\}$. Let σ be a well-behaved selection function w.r.t. $\succ$. Consider the set S of ground formulas:

$$a = b \vee a = c, \quad f(a) \neq f(b), \quad b = c$$

Apply saturation on S using an inference process based on the ground superposition calculus $\mathbb{SUP}_{\succ,\sigma}$ (including the inference rules of ground binary resolution with selection). Show that S is unsatisfiable by finding a refutation of S such that during saturation only 4 new clauses are generated. Give details on what literals are selected and which terms are maximal.

Consider the clause $a = b \vee a = c$. We have $|a = b| = 1+1 = 2 = 1+1 = |a = c|$, but since $b \gg c$ it follows that $b \succ_{KB} c$ and therefore $a = b \succ_{KB} a = c$. Moreover, the orientation of the equalities is already correct, due to the fact that $b \gg c$. Hence,

$$\frac{\underline{a = b} \vee a = c}{a = b \vee b \neq c} \text{ (EF)}$$

Consider the clause $a = b \vee b \neq c$ and the clause. $f(a) \neq f(b)$. We have $|a = b| = 1 + 1 = 2 = 1 + 1 = |b \neq c|$, but since $a \gg b$ it follows that $a \succ_{KB} b$ and therefore $a = b \succ_{KB} b \neq c$. Above all, since $a = b$ is the only positive literal it is automatically the maximal literal. Moreover, the orientation of the equality follows directly from $a \succ_{KB} b$. Similarly, we have $|f(a)| = 1+1 = 2 = 1 + 1 = |f(b)|$, but since $a \gg b$ it follows that $a \succ_{KB} b$ and therefore $f(a) \succ_{KB} f(b)$. Hence,

$$\frac{\underline{a = b} \vee b \neq c \qquad \underline{f(a) \neq f(b)}}{f(b) \neq f(b) \vee b \neq c} \text{ (Sup)}$$

In the resulting clause $f(b) \neq f(b) \vee b \neq c$ we have $|f(b) \neq f(b)| > |b \neq c|$. With $f(b)$ and $f(b)$ being incomparable and $b \succ_{KB} c$. Hence,

$$\frac{\underline{f(b) \neq f(b)} \vee b \neq c}{b \neq c} \text{ (ER)}$$

Lastly, consider the clauses $b \neq c$ and $b = c$. The direction of the equality and inequality already satisfies the required condition, due to $b \gg c$. Moreover, being the only literals in their respective clauses, it follows that only those can be selected.

$$\frac{\underline{b \neq c} \qquad \underline{b = c}}{\square} \text{ (BR)}$$

Problem 3.6

Consider the group theory axiomatisation used in the lecture. Prove that the group's left identity element e is also a right identity.

- Formalize the problem in `TPTP` and solve it using `Vampire`, by running `Vampire` with the additional option specification `--avatar off` . Provide your `TPTP` encoding and `Vampire` output.

- Explain the superposition reasoning part of the `Vampire` proof by detailing the superposition inferences, generated clauses and mgu (if any) in the `Vampire` poof.

The program encoding is

```
fof(left_identity,axiom, !  [X] : mult(e,X) = X).
fof(left_inverse,axiom,!  [X] : mult(inverse(X),X) = e).
fof(associativity,axiom,!  [X,Y,Z] : mult(mult(X,Y),Z) = mult(X,mult(Y,Z))).
fof(right_identity,conjecture,!  [X] : mult(X,e) = X).
```

and the output of the vampire solver is

```
% Refutation found.  Thanks to Tanya!
% SZS status Theorem for group
% SZS output start Proof for group
1.  !  [X0] :  mult(e,X0) = X0 [input]
2.  !  [X0] :  e = mult(inverse(X0),X0) [input]
3.  !  [X0,X1,X2] :  mult(X0,mult(X1,X2)) = mult(mult(X0,X1),X2) [input]
4.  !  [X0] :  mult(X0,e) = X0 [input]
5.   !  [X0] :  mult(X0,e) = X0 [negated conjecture 4]
6.  ?  [X0] :  mult(X0,e) != X0 [ennf transformation 5]
7.  ?  [X0] :  mult(X0,e) != X0 => mult(sK0,e) != sK0 [choice axiom]
8.  mult(sK0,e) != sK0 [skolemisation 6,7]
9.  mult(e,X0) = X0 [cnf transformation 1]
10.  e = mult(inverse(X0),X0) [cnf transformation 2]
11.  mult(X0,mult(X1,X2)) = mult(mult(X0,X1),X2) [cnf transformation 3]
12.  mult(sK0,e) != sK0 [cnf transformation 8]
14.  mult(e,X3) = mult(inverse(X2),mult(X2,X3)) [superposition 11,10]
16.  mult(inverse(X2),mult(X2,X3)) = X3 [forward demodulation 14,9]
20.  mult(inverse(inverse(X1)),e) = X1 [superposition 16,10]
22.  mult(X5,X6) = mult(inverse(inverse(X5)),X6) [superposition 16,16]
33.  mult(X3,e) = X3 [superposition 22,20]
53.  sK0 != sK0 [superposition 12,33]
54.  $false [trivial inequality removal 53]

% SZS output end Proof for group
% ------------------------------
% Version:  Vampire 4.2.2 (commit e1949dd on 2017-12-14 18:39:21 +0000)
% Termination reason:  Refutation
% Memory used [KB]: 383
% Time elapsed:  0.019 s
% ------------------------------
% ------------------------------
```

with the following superposition steps

```
14.  mult(e,X3) = mult(inverse(X2),mult(X2,X3)) [superposition 11,10]
20.  mult(inverse(inverse(X1)),e) = X1 [superposition 16,10]
22.  mult(X5,X6) = mult(inverse(inverse(X5)),X6) [superposition 16,16]
33.  mult(X3,e) = X3 [superposition 22,20]
53.  sK0 != sK0 [superposition 12,33]
```

which will be investigated in the subsequent list. However, first a small discussion, regarding resolvent of variable disjointed variants. As far as I am aware, was it not specified that the calculus allows for or requires the creation of variable disjointed variants before unification. However, after some time I came to the conclusion that this is necessary, if one considers the solver output to be correct. As an example of the necessity of considering variable disjointed variants consider the first and third superposition. Additionally, it is the lifting lemma which requires clauses with no shared variables, which served as a weak justification of my actions.

- `mult(e,X3) = mult(inverse(X2),mult(X2,X3)) [superposition 11,10]`
  The formula $\cdot(e, x_3) = \cdot(\iota(x_2), \cdot(x_2, x_3))$ is obtained by applying the superposition rule to formula (10), i.e $e = \cdot(\iota(x_0), x_0)$ and formula (11), i.e. $\cdot(x_0, \cdot(x_1, x_2)) = \cdot(\cdot(x_0, x_1), x_2)$, both of which are the axioms. That is, after creating a variable disjointed variant one obtains the inference

$$\dfrac{\dfrac{\cdot(\iota(x_0), x_0) = e \qquad \cdot(\cdot(x_1, x_2), x_3) = \cdot(x_1, \cdot(x_2, x_3))}{(\cdot(e, x_3) = \cdot(x_1, \cdot(x_2, x_3)))\{x_0 \mapsto x_2, x_1 \mapsto \iota(x_2)\}}\,(\text{Sup})}{\cdot(e, x_3) = \cdot(\iota(x_2), \cdot(x_2, x_3))}\,\textit{sub.}$$

  Where the mgu of $\cdot(\iota(x_0), x_0)$ and $\cdot(x_1, x_2)$ is $\Theta := \{x_0 \mapsto x_2, x_1 \mapsto \iota(x_2)\}$. That is, $(\cdot(\iota(x_0), x_0))\Theta = (\cdot(\iota(x_2), x_2))$ and $(\cdot(x_1, x_2))\Theta = (\cdot(\iota(x_2), x_2))$ leading to $(\cdot(e, x_3) = \cdot(x_1, \cdot(x_2, x_3)))\Theta = (\cdot(e, x_3) = \cdot(\iota(x_2), \cdot(x_2, x_3)))$.

  (Note: Here my reasoning why it was necessary to create variable disjointed clauses. Firstly, with (e, x_3) being part of the result it is required that $\cdot(\iota(x_0), x_0)$ has to be unified with some term in $\cdot(x_0, \cdot(x_1, x_2)) = \cdot(\cdot(x_0, x_1), x_2)$. Now given the structure of $\cdot(e, x_3) = \cdot(\iota(x_2), \cdot(x_2, x_3))$ the only possible case for unification is to unify $\cdot(\iota(x_0), x_0)$ and $\cdot(x_0, x_1)$, which without variable renaming is an impossibility.)

- `mult(inverse(inverse(X1)),e) = X1 [superposition 16,10]`
  The formula $\cdot(\iota(\iota(x_1)), e) = x_1$ is obtained by applying the superposition rule to formula (10), i.e $e = \cdot(\iota(x_0), x_0)$ and formula (16), i.e. $\cdot(\iota(x_2), \cdot(x_2, x_3)) = x_3$. That is,

$$\cfrac{\cfrac{\cdot(\iota(x_0),x_0)=e \qquad \cdot(\iota(x_2),\cdot(x_2,x_3))=x_3}{(\cdot(\iota(x_2),e)=x_3)\{x_0\mapsto x_1,x_2\mapsto\iota(x_1),x_3\mapsto x_1\}}\text{(Sup)}}{\cdot(\iota(\iota(x_1)),e)=x_1}\mathit{sub.}$$

Where the mgu of $\cdot(\iota(x_0),x_0)$ and $\cdot(x_2,x_3)$ is $\Theta := \{x_0 \mapsto x_1, x_2 \mapsto \iota(x_1), x_3 \mapsto x_1\}$. That is, $(\cdot(\iota(x_0),x_0))\Theta = (\cdot(\iota(x_1),x_1))$ and $(\cdot(x_2,x_3)))\Theta = (\cdot(\iota(x_1),x_1)) = x_1)$ leading to $(\cdot(\iota(x_2),e) = x_3)\Theta = (\cdot(\iota(\iota(x_1)),e) = x_1)$.

- `mult(X5,X6) = mult(inverse(inverse(X5)),X6) [superposition 16,16]`
The formula $\cdot(x_5,x_6) = \cdot(\iota(\iota(x_5)),x_6)$ is obtained by applying the superposition rule to formula (16), i.e $\cdot(\iota(x_2),\cdot(x_2,x_3)) = x_3$ and formula (16), i.e. $\cdot(\iota(x_2),\cdot(x_2,x_3)) = x_3$. That is, after creating a variable disjointed variant one obtains the inference

$$\cfrac{\cfrac{\cdot(\iota(x_0),\cdot(x_0,x_1))=x_1 \qquad \cdot(\iota(x_2),\cdot(x_2,x_3))=x_3}{(\cdot(\iota(x_2),x_1)=x_3)\{x_0\mapsto x_5,x_1\mapsto x_6,x_2\mapsto\iota(x_5),x_3\mapsto\cdot(x_5,x_6)\}}\text{(Sup)}}{\cdot(\iota(\iota(x_5)),x_6)=\cdot(x_5,x_6)}\mathit{sub.}$$

Where the mgu of $\cdot(\iota(x_0),\cdot(x_0,x_1))$ and $\cdot(x_2,x_3)$ is $\Theta := \{x_0 \mapsto x_5, x_1 \mapsto x_6, x_2 \mapsto \iota(x_5), x_3 \mapsto \cdot(x_5,x_6)\}$. That is, $(\cdot(\iota(x_0),\cdot(x_0,x_1)))\Theta = (\cdot(\iota(x_5),\cdot(x_5,x_6)))$ and $(\cdot(x_2,x_3))\Theta = (\cdot(\iota(x_5),(x_5,x_6)))$, leading to $(\cdot(\iota(x_2),x_1) = x_3)\Theta = (\cdot(\iota(\iota(x_5)),x_6) = \cdot(x_5,x_6))$.

(Note: Again unfortunately I had to make the variables in those two clauses disjointed. Since both terms are equal, any substitution would result again in two equal terms. Preventing the inference as provided by the solver.)

- `mult(X3,e) = X3 [superposition 22,20]`
The formula $\cdot(x_3,e) = x_3$ is obtained by applying the superposition rule to formula (20), i.e $\cdot(\iota(\iota(x_1)),e) = x_1$ and formula (22), i.e. $\cdot(x_5,x_6) = \cdot(\iota(\iota(x_5)),x_6)$. That is,

$$\cfrac{\cfrac{\cdot(\iota(\iota(x_1)),e)=x_1 \qquad \cdot(\iota(\iota(x_5)),x_6)=\cdot(x_5,x_6)}{(x_1=\cdot(x_5,x_6))\{x_1\mapsto x_3,x_5\mapsto x_3,x_6\mapsto e\}}\text{(Sup)}}{x_3=\cdot(x_3,e)}\mathit{sub.}$$

Where the mgu of $\cdot(\iota(\iota(x_1)),e)$ and $\cdot(\iota(\iota(x_5)),x_6)$ is $\Theta := \{x_1 \mapsto x_3, x_5 \mapsto x_3, x_6 \mapsto e\}$. That is, $(\cdot(\iota(\iota(x_1)),e))\Theta = (\cdot(\iota(\iota(x_3)),e))$ and $(\cdot(\iota(\iota(x_5)),x_6))\Theta = (\cdot(\iota(\iota(x_3)),e))$, leading to $(x_1 = \cdot(x_5,x_6))\Theta = (x_3 = \cdot(x_3,e))$.

- `sK0 != sK0 [superposition 12,33]`
  The formula $s_{k_0} \neq s_{k_0}$ is obtained by applying the superposition rule to formula (12), i.e $\cdot(s_{k_0}, e) \neq s_{k_0}$ and formula (33), i.e. $\cdot(x_3, e) = x_3$. That is,

$$\dfrac{\dfrac{\cdot(x_3, e) = x_3 \qquad \cdot(s_{k_0}, e) \neq s_{k_0}}{(x_3 \neq s_{k_0})\{x_3 \mapsto s_{k_0}\}}\ \text{(Sup)}}{s_{k_0} \neq s_{k_0}}\ \textit{sub.}$$

  Where the mgu of $\cdot(x_3, e)$ and $\cdot(s_{k_0}, e)$ is $\Theta := \{x_3 \mapsto s_{k_0}\}$. That is, $(\cdot(x_3, e))\Theta = (\cdot(s_{k_0}, e))$ and $(\cdot(s_{k_0}, e))\Theta = (\cdot(s_{k_0}, e))$, leading to $(x_3 \neq s_{k_0})\Theta = (s_{k_0} \neq s_{k_0})$.

In the following, two inferences seem to be part of the inference system as introduced in the lecture.

- `mult(inverse(X2),mult(X2,X3)) = X3 [forward demodulation 14,9]`
  The formula $\cdot(\iota(x_2), \cdot(x_2, x_3)) = x_3$ is obtained by applying the forward demodulation rule to formula (9), i.e $\cdot(e, x_0) = X_0$ and formula (14), i.e. $\cdot(e, x_3) = \cdot(\iota(x_2), \cdot(x_2, x_3))$. That is,

$$\dfrac{\dfrac{\cdot(e, x_0) = X_0 \qquad \cdot(\iota(x_2), \cdot(x_2, x_3)) = \cdot(e, x_3)}{(\cdot(\iota(x_2), \cdot(x_2, x_3)) = x_0)\{x_0 \mapsto x_3\}}\ \text{(Sup)}}{\cdot(\iota(x_2), \cdot(x_2, x_3)) = x_3}\ \textit{sub.}$$

  Where the mgu of $\cdot(e, x_0)$ and $\cdot(e, x_3)$ is $\Theta := \{x_0 \mapsto x_3\}$. Leading to $(\cdot(\iota(x_2), \cdot(x_2, x_3)) = x_0)\Theta = (\cdot(\iota(x_2), \cdot(x_2, x_3)) = x_3)$ after the substitution.

- `$false [trivial inequality removal 53]`
  Given the fact that "trivial inequality removal" is not an inference that was formally introduced in the lecture. The only possible method known capable of explaining this inference is "Equality Resolution". That is,

$$\dfrac{s_{k_0} \neq s_{k_0}}{\square}\ \text{(ER)}$$

Apart from that, most other generated clauses are either the result of skolemisation (8) , some normal form transformations (6,9,10,11,12) and the negation of the input conjecture (5). The only, remaining step is the one resulting in clause (7). In fact this is not an inference, but rather an axiom expressing that since there exists an element satisfying the statement, one can introduce a fresh skolem constant to remove the existential quantification.