Lab Exercises for Lecture 1

Problem 1.1. Consider a well-founded strict ordering $\succ$ on atoms. Prove that the induced ordering on literals, as defined in the lecture, is also well-founded.

Problem 1.2. 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 $C \succ_{bag} D$.

Solution:
Since B and A are syntactically the same atoms, we have that A is the maximal atom of D wrt $\succ$. We know that A occurs only positively in D. By using properties of the induced literal ordering $\succ$, we conclude that $\neg A \succ A \succ \neg D_j \succ D_j$ for every atom D_j of D different than A. Hence, by properties of the bag extension ordering of $\succ$, we have $\neg A \succ_{bag} D$.

By assumption, A is the maximal atom of C wrt $\succ$ and A occurs only negatively in C. As $\neg A \succ A$, we thus conclude that $\neg A \succ \neg C_i \succ C_i$, where C_i is an atom of C different than A. That is $\neg A$ is the maximal literal of C.
As $\neg A \succ_{bag} D$ and $\neg A$ is the maximal literal of C, by properties of the bag extension ordering of $\succ$, we finally conclude that $C \succ_{bag} D$.

Problem 1.3. Consider strict partial orderings $\succ_i$ over M_i, for $i = 1, 2$. Assume that $\succ_1$ and $\succ_2$ are well-founded. We define the ordering $\succ^*$ over $M_1 \times M_2$ as:

$$(a_1, a_2) \succ^* (b_1, b_2) \Leftrightarrow \Big(a_1 \succ_1 b_1 \quad \text{or} \quad (a_1 = b_1 \quad \text{and} \quad a_2 \succ_2 b_2)\Big)$$

Show that $\succ^*$ is well-founded. **Solution:**

If $\succ^*$ is not well-founded, there has to be a an infinite decreasing chain of pairs:

$$(x_0, y_0) \succ^* (x_1, y_1) \succ^* (x_2, y_2) \succ^* \ldots$$

By the definition of $\succ^*$ it follows that for each i, either $x_i \succ_1 x_{i+1}$ or $x_i = x_{i+1}$. However, since $\succ_1$ is well-founded, the sequence $\{x_i\}_{i \geq 0}$ has a minimal element: x_n for some n. Then, from the definition of $\succ^*$ we obtain $\forall i \geq n : y_n \succ_2 y_{n+1}$. However, that is a contradiction with $\succ_2$ being well-founded. Hence, $\succ^*$ must be well-founded as well.

Problem 1.4. Let $\mathbb{I}$ be a sound inference system on clauses and let S_0 be a non-empty set of clauses. Consider a fair $\mathbb{I}$-inference process $S_0 \to S_1 \to S_2 \to \ldots$, without redundancy elimination. Let I_∞ denote the limit of this fair $\mathbb{I}$-inference process. Show that I_∞ is the $\mathbb{I}$-closure of S_0.

Note: You need to prove that I_∞ is the smallest $\mathbb{I}$-saturated set containing S_0. Recall and use the property from the lecture on $\mathbb{I}$-inference processes $S_0 \to S_1 \to S_2 \to \ldots$, in particular that every S_i is a subset of the $\mathbb{I}$-closure of S_0.

Solution.

1. First, we show that I_∞ is saturated.

   Towards a contradiction, assume I_∞ is not saturated. This means there are clauses $C_1, \dots, C_n \in I_\infty$ and an inference
   $$\frac{C_1 \quad \cdots \quad C_n}{C}$$
   such that $C \notin I_\infty$.

   However, since $\mathbb{I}$ is fair, C is derived at some step S_i. Thus $C \in S_i \subseteq I_\infty$, a contradiction.

2. Next, we show that if X is an $\mathbb{I}$-saturated set of clauses with $S_0 \subseteq X$, then $I_\infty \subseteq X$.

   By induction over i, we see that $S_i \subseteq X$ for all i:

   - $S_0 \subseteq X$ holds by assumption.
   - Assume $S_i \subseteq X$. Let C be the clause derived in step $i+1$, i.e., $S_{i+1} \setminus S_i = \{C\}$. This means that we have clauses $C_1, \dots, C_n \in S_i$ and an inference
     $$\frac{C_1 \quad \cdots \quad C_n}{C} \quad .$$
     Because of $S_i \subseteq X$, we have $C_1, \dots, C_n \in X$ and thus, because X is saturated, also $C \in X$. Hence $S_{i+1} \subseteq X$.

   Recall the definition of $I_\infty := \bigcup_{i \in \mathbb{N}} S_i$. Since $S_i \subseteq X$ for all i, also $I_\infty \subseteq X$.

3. Since I_∞ is saturated with $S_0 \subseteq I_\infty$, and is a subset of *all* saturated sets with this property, I_∞ is the smallest such set.

Problem 1.5. Let S be the following set of clauses:

$$\{\, \neg p \lor \neg q, \quad \neg p \lor q, \quad p \lor \neg q, \quad p \lor q \,\}$$

Consider the binary resolution inference system BR (without ordering and selection function). Show that there exists an infinite number of different BR derivations of the empty clause from the clauses of S.

Solution.
First consider the following derivation of the empty clause.

$$\dfrac{\dfrac{\dfrac{\neg p \lor \neg q \qquad p \lor \neg q}{\neg q \lor \neg q} \qquad \dfrac{\dfrac{\neg p \lor q \qquad p \lor q}{q \lor q}}{q}}{\neg q} \qquad \dfrac{\dfrac{\neg p \lor q \qquad p \lor q}{q \lor q}}{q}\ (*)}{\Box}$$

Then consider the following other derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\cdots}{q} \qquad \neg p \lor \neg q}{\neg p} \qquad p \lor q}{q}}{\cdots}$$

The we can insert the latter derivation arbitrarily often at step (*) in the original derivation, hence there is an abritrary number of derications of the empty clause.

Lab Exercises for Lecture 2

Problem 2.1. Let $\succ$ be a total well-founded ordering on the ground atoms $p_1, \ldots, p_6$ such that $p_6 \succ p_5 \succ p_4 \succ p_3 \succ p_2 \succ p_1$. Consider the bag extension of $\succ$; for simplicity, denote the bag extension of $\succ$ also by $\succ$.
Using $\succ$, compare and order the following three clauses:

$$p_6 \vee \neg p_6, \qquad \neg p_2 \vee p_4 \vee p_5, \qquad p_2 \vee p_3.$$

Solution:

Since $\succ$ is total, $p_i \succ p_j$ and $p_i \succ \neg p_j$ for each $1 \leq j < i \leq 6$. Therefore $p_6 \succ \neg p_2$, $p_6 \succ p_4$, $p_6 \succ p_5$, and $p_5 \succ p_2$, $p_5 \succ p_3$. Further, since for any two bags B, B', such that $B \supset B'$, it holds that $B \succ B'$, we have $p_6 \vee \neg p_6 \ \succ \ p_6$ and $\neg p_2 \vee p_4 \vee p_5 \ \succ \ p_5$. By combining these observations, we get:

$$p_6 \vee \neg p_6 \ \succ \ p_6 \ \succ \ \neg p_2 \vee p_4 \vee p_5 \ \succ \ p_5 \ \succ \ p_2 \vee p_3$$

Therefore by transitivity of $\succ$:

$$p_6 \vee \neg p_6 \ \succ \ \neg p_2 \vee p_4 \vee p_5 \ \succ \ p_2 \vee p_3$$

Problem 2.2. Let p, q be boolean atoms and let S be the following set of ground formulas:

$$\{ \neg p \vee \neg q, \quad \neg p \vee q, \quad p \vee \neg q, \quad p \vee q \}$$

Take any ordering such that $p \succ q$ and any selection function σ over S such that

$$\{ \neg p \vee \underline{\neg q}, \quad \underline{\neg p} \vee q, \quad p \vee \underline{\neg q}, \quad \underline{p} \vee q \}.$$

(a) Is σ a well-behaved selection function over S? Justify your answer!

(b) How many inferences of $\mathbb{BR}_\sigma$ are applicable to S? Justify your answer!

Solution.

(a) Recall that a well-behaved selection function selects a negative literal, or all maximal literals in a clause.

In the first three clauses, a negative literal is selected. In the last clause, p is selected which is the (only) maximal literal due to $p \succ q$. Hence σ is well-behaved.

(b) No factoring inference is possible, because no positive literal appears more than once in any of the clauses. Binary resolution can only be performed on clauses where the resolved literal is selected. As such, there is only one possible inference (between the second and the clause):

$$\frac{\underline{\neg p} \vee q \quad \underline{p} \vee q}{q \vee q}$$

Problem 2.3. Give an example of a non-tautology ground clause with at least one selected literal so that this selection is not well-behaved for any ordering $\succ$. Justify your solution!

Solution.
Consider the following clause:

$$p \vee \underline{p}$$

Obviously p is maximal in this clause, for any ordering. There is one maximal literal which is not selected, hence the selection function is not well-behaved.

Problem 2.4. Let S be the set of clauses

$$\neg q \vee r, \quad \neg p \vee q, \quad \neg r \vee \neg q, \quad \neg q \vee \neg p, \quad \neg p \vee \neg r, \quad \neg r \vee p, \quad r \vee q \vee p$$

(a) Prove unsatisfiabiliy of S using BR.

(b) Formalize S in TPTP and prove its unsatisfiability using `Vampire`, by running `Vampire` with the additional option `-av off` .

Lab Exercises for Lecture 3

Problem 3.1. Consider a KBO ordering $\succ$ such that $inverse \gg times$ by precedence. Consider the literal:

$$inverse(times(x, y)) = times(inverse(y), inverse(x)).$$

Compare, w.r.t $\succ$, the left- and right-hand side terms of the equality when:

- $weight(inverse) = weigth(times) = 1$;

  Solution:

  As $weight(inverse) = weigth(times) = 1$, we have:

  $$weight(inverse(times(x, y))) = 2 + weight(x) + weight(y)$$

  and

  $$weight(times(inverse(y), inverse(x))) = 3 + weight(y) + weight(x).$$

  Thus,

  $$times(inverse(y), inverse(x)) \succ inverse(times(x, y)).$$

- $weight(inverse) = 0$ and $weight(times) = 1$.

  Solution:

  Using the given weights of $times$ and $inverse$, we have:

  $$weight(inverse(times(x, y)))) = weight(times(inverse(y), inverse(x))) = 1+weight(x)+weight(y).$$

  Then by precedence, given that $inverse \gg times$, we conclude that:

  $$inverse(times(x, y)) \succ times(inverse(y), inverse(x)).$$

Problem 3.2. 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 $\succ$ 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 $\succ$.

Solution:
Every ground term of Σ that has the minimal weight wrt $\succ_{KB}$ is either:
– a constant $c \in \Sigma$ such that c has the minimal weight among the constants of Σ,
– or a term $f^n(c)$, with $n \neq 0$, where $c \in \Sigma$ is a constant of the minimal weight among the constants of Σ and $w(f) = 0$.

Problem 3.3. Consider the set S of ground formulas:

$$\begin{aligned}\{\ & g(f(a)) = a \vee g(f(b)) = a,\\ & f(a) = a,\\ & f(b) \neq f(b) \vee f(b) = a,\\ & g(a) \neq a\ \}\end{aligned}$$

Show that S is unsatisfiable by applying saturation on S using an inference process based on the ground superposition calculus $\mathbb{S}\mathrm{up}_{\succ,\sigma}$ (including the inference rules of binary resolution $\mathbb{BR}_\sigma$), where σ is a well-behaved selection function wrt $\succ$ and:

(a) the ordering $\succ$ is the KBO ordering generated by the precedence $f \gg a \gg g \gg b$ and the weight function w with $w(f) = 0, w(b) = 1, w(a) = 2, w(g) = 3$;

(b) the ordering $\succ$ is the KBO ordering generated by the precedence $g \gg a \gg b \gg f$ and the weight function w with $w(g) = 0, w(b) = 1, w(f) = 1, w(a) = 3$.

Give details on what literals are selected and which terms are maximal.

Solution:
For clarity, we first number the clauses:

$$\begin{aligned}&(1)\ g(f(a)) = a \vee g(f(b)) = a\\ &(2)\ f(a) = a\\ &(3)\ f(b) \neq f(b) \vee f(b) = a\\ &(4)\ g(a) \neq a\end{aligned}$$

In the solution we mark selected literals by underlining them.

(a) The following literals are selected:

(1) $\underline{g(f(a)) = a} \vee g(f(b)) = a$ ($\{g(f(a)), a\} \succ \{g(f(b)), a\}$ since $g(f(a)) \succ g(f(b))$ by weight)

(2) $\underline{f(a) = a}$ (since it is the only literal in the clause)

(3) $\underline{f(b) \neq f(b)} \vee f(b) = a$ (since $f(b) \neq f(b)$ is negative)

(4) $\underline{g(a) \neq a}$ (since it is the only literal in the clause)

By equality resolution over (3), we get:

$$(5)\ \underline{f(b) = a}$$

Next, since $f(a) \succ a$ (by precedence) and $g(f(a)) \succ a$ (by weight), we apply superposition over (2) and (1), and get

$$(6)\ \underline{g(a) = a} \vee g(f(b)) = a,$$

where $g(a) = a$ is selected since $g(a) \succ g(f(b))$ (by weight) and thus $\{g(a), a\} \succ \{g(f(b)), a\}$. We then apply binary resolution over (6) and (4):

$$(7)\ \underline{g(f(b)) = a}$$

Next, since $a \succ f(b)$ (by weight) and $g(a) \succ a$ (by weight), we apply superposition over (4) and (5):

$$(8)\ \underline{g(f(b)) \neq a}$$

Finally, we apply binary resolution over (7) and (8):

$$(9)\ \square$$

Hence, S is unsatisfiable.

(b) All steps of the solution (a) depended on the following comparisons between (bags of) terms: $\{g(f(a)), a\} \succ \{g(f(b)), a\}$, $f(a) \succ a$, $g(f(a)) \succ a$, $\{g(a), a\} \succ \{g(f(b)), a\}$, $a \succ f(b)$, $g(a) \succ a$. However, all these comparisons hold also for the KBO generated by the precedence and the weight function from (b):

- from $g(f(a)) \succ g(f(b))$ by weight we get $\{g(f(a)), a\} \succ \{g(f(b)), a\}$,
- $f(a) \succ a$ by weight,
- $g(f(a)) \succ a$ by weight,
- from $g(a) \succ g(f(b))$ by weight it follows $\{g(a), a\} \succ \{g(f(b)), a\}$,
- $a \succ f(b)$ by weight,
- $g(a) \succ a$ by precedence.

Therefore the proof in (b) is the same as in (a).

Lab Exercises for Lecture 4

Problem 4.1. Apply the unification algorithm and show the most general unifier of the following atoms:

(a) $p(a, f(y), y)$ and $p(a, x, f(x))$;

Solution:

$$
\begin{array}{ll}
E = \{p(a, f(y), y) = p(a, x, f(x))\} & \Longrightarrow \\
E = \{a = a, \quad f(y) = x, \quad y = f(x)\} & \Longrightarrow \\
E = \{f(y) = x, \quad y = f(x)\} & \Longrightarrow \\
E = \{x = f(y), \quad y = f(x)\} & \Longrightarrow_{x \to f(y)} \\
E = \{x = f(y), \quad y = f(f(y))\} & \Longrightarrow_{Failure}
\end{array}
$$

(b) $p(f(x, a), f(f(b, a)))$ and $p(z, f(z))$;

Solution: Note that f occurs both as a unary and as a binary function. Hence, there are two possibilities to proceed:

- Report syntax error because one cannot use the same function name with different arities;
- Consider the unary and binary occurences of f as different functions. Use then f_1 for the unary occurence of f and f_2 for the binary function f. (Note: In Prolog, it is still possible to use the same function name with different arities.).

  In this case, we have:

$$
\begin{array}{ll}
E = \{p(f_2(x, a), f_1(f_2(b, a))) = p(z, f_1(z))\} & \Longrightarrow \\
E = \{f_2(x, a) = z, \quad f_1(f_2(b, a)) = f_1(z)\} & \Longrightarrow \\
E = \{z = f_2(x, a), \quad f_1(f_2(b, a)) = f_1(z)\} & \Longrightarrow_{z \to f_2(x,a)} \\
E = \{z = f_2(x, a), \quad f_1(f_2(b, a)) = f_1(f_2(x, a))\} & \Longrightarrow \\
E = \{z = f_2(x, a), \quad f_2(b, a) = f_2(x, a)\} & \Longrightarrow \\
E = \{z = f_2(x, a), \quad b = x, \quad a = a\} & \Longrightarrow \\
E = \{z = f_2(x, a), \quad x = b, \quad a = a\} & \Longrightarrow_{x \to b} \\
E = \{z = f_2(b, a), \quad x = b, \quad a = a\} & \Longrightarrow \\
E = \{z = f_2(b, a), \quad x = b\} & \Longrightarrow_{Success}
\end{array}
$$

  The mgu in this case is $\{z \to f_2(b, a), x \to b\}$.

(c) $p(f(x, y), f(y, z))$ and $p(z, f(w, f(y, w)))$.

Solution:

$$\begin{array}{lll}
E = \{p(f(x,y), f(y,z)) \ = \ p(z, f(w, f(y,w)))\} & \Longrightarrow & \\
E = \{f(x,y) = z, \quad f(y,z) = f(w, f(y,w))\} & \Longrightarrow & \\
E = \{z = f(x,y), \quad f(y,z) = f(w, f(y,w))\} & \Longrightarrow & {\scriptstyle z \to f(x,y)} \\
E = \{z = f(x,y), \quad f(y, f(x,y) = f(w, f(y,w))\}\} & \Longrightarrow & \\
E = \{z = f(x,y), \quad y = w, \quad f(x,y) = f(y,w)\} & \Longrightarrow & {\scriptstyle y \to w} \\
E = \{z = f(x,w), \quad y = w, \quad f(x,w), f(w,w)\}\} & \Longrightarrow & \\
E = \{z = f(x,w), \quad y = w, \quad x = w, \quad w = w\} & \Longrightarrow & {\scriptstyle x \to w} \\
E = \{z = f(w,w), \quad y = w, \quad x = w, \quad w = w\} & \Longrightarrow & \\
E = \{z = f(w,w), \quad y = w, \quad x = w\} \Longrightarrow {\scriptstyle Success} & &
\end{array}$$

The mgu is:

$$\{z \to f(w,w), \ y \to w, x \to w\}.$$

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

Problem 4.2. 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.

Solution:
For simplicity, we name the given clauses by numbers:

$$\begin{array}{ll}
(1) & \neg p(z,a) \vee \neg p(z,x) \vee \neg p(x,z) \\
(2) & p(y,a) \vee p(y, f(y)) \\
(3) & p(w,a) \vee p(f(w), w)
\end{array}$$

By negative factoring on (1), with the mgu $\{x \to a\}$, we get:

$$(4) \quad \neg p(z,a) \vee \neg p(a,z)$$

By negative factoring on (4), with the mgu $\{z \to a\}$, we get:

$$(5) \quad \neg p(a,a)$$

By resolution on (5) and (2), with the mgu $\{y \to a\}$, we get:

$$(6) \quad p(a, f(a))$$

By resolution on (4) and (6), with the mgu $\{z \to f(a)\}$, we get:

$$(7) \quad \neg p(f(a), a)$$

By resolution on (3) and (7), with the mgu $\{w \to a\}$, we get:

$$(8) \quad p(a,a)$$

By resolution on (5) and (8), we finally obtain the empty clause:

$$(9) \quad \square$$

Hence, our input set S of clauses (1), (2) and (3) is unsatisfiable.

Problem 4.3. Let p denote a unary predicate symbol, f a unary function symbol, x, y variables and c a constant. Let C_1 be the clause $p(x) \vee p(y)$ and consider C_2 to be the clause $p(x)$. Further, let D denote the clause $p(f(c))$.

(a) Does C_1 subsume D?

(b) Does C_2 subsume D?

Justify your answers!

Solution:

(a) No. For C_1 to subsume D, there must be a substitution θ such that $C_1\theta$ is a sub-multiset of D. However, since C_1 has two literals, also $C_1\theta$ has two literals and therefore it cannot be a sub-multiset of D, which only has one literal.

(b) Yes. For $\theta = \{x \mapsto f(c)\}$, it holds that $C_2\theta$ is a sub-multiset of D.

Problem 4.4. Let x denote a variable, a, b, c constants, and f a unary function symbol.
Give a superposition refutation of the following set of two clauses:

$$\{ \quad x = f(c), \\ a \neq b \quad \}$$

such that, in every inference, the premises and the conclusion of that inference do not use the symbols f, c together with the symbols a, b. That is, every inference has the following property: if the premise or the conclusion contain any of the symbols f, c, then the premise and the conclusion contain neither a nor b.
In your proof, use only the inference system of the superposition calculus $\mathbb{S}\mathrm{up}$ (without ordering and selection function); that is, no inferences of binary resolution $\mathbb{BR}$ should be used. For each newly derived clause, clearly label the clauses from which it was derived and indicate most general unifiers.

Solution. Note that if we apply a rule to two clauses, we consider the variables in both clauses to be distinct, even if the two clauses are actually the same one. Therefore we when we apply can use superposition using $f(c) = x$, as both of the premises it is the same thing as applying it to the premises $f(c) = x$ and $f(c) = y$. This means we can do the following derivation:

1)	$f(c) = x$	Axiom
2)	$a \neq b$	Axiom
3)	$x = y$	Superposition using (1), and (1), $\sigma = \emptyset$
4)	$a \neq x$	Superposition using (2), and (3), $\sigma = \{y \mapsto b\}$
5)	$\square$	Equality resolution using (4), $\sigma = \{x \mapsto a\}$

Problem 4.5.
Let f be a unary and g be a binary function symbol. Further let a, b, c be constants, and x, y, z be variables. We define the weight function $w(s) = w(v) = 1$, for every symbol s and variable v, and let $g \gg f \gg c \gg b \gg a$. Answer the following questions using a KBO with the weight function w and the precedence relation $\gg$ to order terms, and its extension to compare literals and clauses.

(a) Do the clauses C_1 and C_2 make the clause C_3 redundant?

$$\begin{aligned} C_1 &: \ a \neq b \vee f(a) \neq a \\ C_2 &: \ f(f(x)) = a \\ C_3 &: \ f(f(b)) \neq b \vee f(a) \neq a \end{aligned}$$

(b) Does the clause C_4 make the clause C_5 redundant?

$$\begin{aligned} C_4 &: \ f(g(x,a)) \neq f(y) \\ C_5 &: \ f(g(x,z)) \neq f(g(y,b)) \vee f(g(x,b)) \neq f(g(a,b)) \end{aligned}$$

(c) Does the clause C_6 make the clause C_7 redundant?

$$\begin{aligned} C_6 &: \ g(x,y) \neq f(x) \\ C_7 &: \ g(f(x), f(z)) \neq f(f(x)) \vee g(a,b) \neq c \end{aligned}$$

Solution.

(a) No, because $C_2 : f(f(x)) = a$ is incomparable to $C_3 : f(f(b)) \neq b \vee f(a) \neq a$.

(b) Yes, because $C_4 : f(g(x,a)) \neq f(y)$ is unsat, and smaller than C_5.

(c) Yes because firstly we can rename the variable z to y in the second clause. This means, since $g(x,y) \neq f(x) \prec g(f(x), f(z)) \neq f(f(x))$, that $C_6 \prec C_7$. Further $g(f(x), f(z)) \neq f(f(x))$ is an instance of $g(x,y) \neq f(x)$, which means that we C_6 implies C_7.

Problem 4.6. Consider the following inference:

$$\frac{x = f(c) \vee p(x) \quad f(h(b)) = h(g(y,y)) \vee h(g(d,b)) \neq f(c)}{p(h(g(d,b))) \vee f(h(b)) = h(g(y,y))}$$

in the non-ground superposition inference system $\mathbb{S}\mathrm{up}$ (**without** the rules of the non-ground binary resolution inference system $\mathbb{BR}$), where p is a predicate symbol, f, g, h are function symbols, b, c, d are constants, and x, y are variables.

(a) Prove that the above inference is a sound inference of $\mathbb{S}\mathrm{up}$.

(b) Is the above inference a simplifying inference of $\mathbb{S}\mathrm{up}$? Justify your answer.

Solution.

(a) Let M be a model of both assumptions of the rule:

$$M \models x = f(c) \vee p(x) \tag{1}$$
$$M \models f(h(b)) = h(g(y,y)) \vee h(g(d,b)) \neq f(c) \tag{2}$$

If $M \models f(h(b)) = h(g(y,y))$, then the conclusion is obviously true in M. Otherwise due to (2) we know that it must be the case that $M \models h(g(d,b)) \neq f(c)$. This means due to (1) that $M \models p(g(d,b))$ must hold, which means that the conclusion is true in this case as well.

(b) We will call the left assumption C_1 and the right assumption of the rule C_2, and the conclusion of the rule D.

In order for the rule to be a simplifying rule, it could either make C_1, or C_2 redundant, so there are two cases to check.

Let's first have a look at whether C_1 is being made redundant. In order for that to hold we would need to have that $C_2, D \models C_1$. This does not hold since the model M_1 is a counterexample.

$$
\begin{aligned}
M_1(\top) &= \{\mathsf{a}, \mathsf{b}\} \\
M_1(p) &= \emptyset \\
M_1(f)(x) &= \mathsf{a} \\
M_1(h)(x) &= \mathsf{a} \\
M_1(g)(x, y) &= \mathsf{a}
\end{aligned}
$$

(Note that by $M(\top)$ we denote the domain of the model here.)

Similarly we can build a counterexample M_2 for the statement $C_1, D \models C_2$.

$$
\begin{aligned}
M_2(p) = M_2(\top) &= \{\mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{f}, \mathsf{g}, \mathsf{h}\} \\
M_2(b) &= \mathsf{b} \\
M_2(c) &= \mathsf{c} \\
M_2(d) &= \mathsf{d} \\
M_2(f)(x) &= \begin{cases} \mathsf{h} & \text{if } x = \mathsf{c} \\ \mathsf{f} & \text{else} \end{cases} \\
M_2(g)(x, y) &= \mathsf{g} \\
M_2(h)(x) &= \mathsf{h}
\end{aligned}
$$

Problem 4.7. Recall that the inverse of the binary relation $r_1(x, y)$ is the binary relation $r_2(y, x)$ such that $r_1(x, y)$ if and only if $r_2(y, x)$.

Prove that the inverse of a dense order is also dense. For doing so, you are required to do the following steps:

- Formalize the problem in `TPTP` and prove it using `Vampire`.
- Explain the superposition reasoning part of the `Vampire` proof by detailing the superposition inferences, generated clauses and mgus in the poof. Use `Vampire` with the AVATAR option off, that is `-av off`.

Problem 4.8. Consider the group theory axiomatization used in the lecture. Prove that the group's left identity element e is also a right identity.

- Formalize the problem in `TPTP` and use it using `Vampire`, by running `Vampire` with the additional option `-av off` .
- Explain the superposition reasoning part of the `Vampire` proof by detailing the superposition inferences, generated clauses and mgus in the poof.