Exercises for the Logic Course

First Order Logic

Course Web Page
http://www.inf.unibz.it/~artale/DML/dml.htm

Computer Science

Free University of Bozen-Bolzano
January 22, 2018

1 Exercises

1.1 Equivalence and Entailment

(1) Use the tableaux procedure to prove that the following entailments hold.

1. $\exists x \forall y r(x, y) \models \forall y \exists x r(x, y)$;
2. $\forall x p(x) \vee \forall x q(x) \models \forall x(p(x) \vee q(x))$;
3. $\exists x(p(x) \wedge q(x)) \models \exists x p(x) \wedge \exists x q(x)$.

Solution. Given that $\varphi \models \psi$ iff $\models \varphi \to \psi$, proving the given **entailments** is equivalent to proving the **validity** of the following formulae:

$$\exists x \forall y\, r(x, y) \to \forall y \exists x\, r(x, y); \tag{1}$$

$$\forall x\, p(x) \vee \forall x\, q(x) \to \forall x\, (p(x) \vee q(x)); \tag{2}$$

$$\exists x\, (p(x) \wedge q(x)) \to \exists x\, p(x) \wedge \exists x\, q(x). \tag{3}$$

Note. Proving the **validity** of $\varphi \to \psi$ (every interpretation is a model of $\varphi \to \psi$) amounts to proving the **unsatisfiability** of $\neg(\varphi \to \psi) \equiv \varphi \wedge \neg\psi$ (no interpretation is a model of $\varphi \wedge \neg\psi$). Unsatisfiability of $\varphi \wedge \neg\psi$ is proved by building a **closed** tableau for the NNF of $\varphi \wedge \neg\psi$.

We do it for the first formula (1). Its negation in NNF is as follows: $\exists x \forall y\, r(x, y) \wedge \exists y \forall x \neg r(x, y)$.

$$\begin{array}{c}
[\exists x \forall y\, r(x, y) \wedge \exists y \forall x \neg r(x, y)] \\
| \\
[\exists x \forall y\, r(x, y)] \\
[\exists y \forall x \neg r(x, y)] \\
| \\
\forall y\, r(a, y) \\
| \\
\forall x \neg r(x, b) \\
| \\
r(a, a) \\
r(a, b) \\
\forall y\, r(a, y) \\
| \\
\neg r(a, b) \\
\neg r(b, b) \\
\forall x \neg r(x, b) \\
\text{closed}
\end{array}$$

(2) Use the tableaux procedure to prove that the following entailments do not hold.

1. $\forall x(p(x) \vee q(x)) \models \forall x p(x) \vee \forall x q(x)$;
2. $\exists x p(x) \wedge \exists x q(x) \models \exists x(p(x) \wedge q(x))$.

Solution. Reasoning as above, the **entailments do not hold** iff the negation of the following formulae

$$\forall x(p(x) \vee q(x)) \to \forall x p(x) \vee \forall x q(x) \tag{4}$$

$$\exists x p(x) \wedge \exists x q(x) \to \exists x(p(x) \wedge q(x)) \tag{5}$$

are **satisfiable**. We consider the formula (4). Its negation, in NNF, is $\forall x(p(x) \vee q(x)) \wedge (\exists x \neg p(x) \wedge \exists x \neg q(x))$. An **open** tableau for this is as follows.

$$[\forall x(p(x) \vee q(x)) \wedge (\exists x \neg p(x) \wedge \exists x \neg q(x))]$$
$$|$$
$$\forall x(p(x) \vee q(x))$$
$$[\exists x \neg p(x) \wedge \exists x \neg q(x)]$$
$$|$$
$$[\exists x \neg p(x)]$$
$$[\exists x \neg q(x)]$$
$$|$$
$$\neg p(a)$$
$$|$$
$$\neg q(b)$$
$$|$$
$$[p(a) \vee q(a)]$$
$$[p(b) \vee q(b)]$$
$$\forall x(p(x) \vee q(x))$$

$$\begin{array}{ccc} & & \\ p(a) & & q(a) \\ \text{closed} & & \begin{array}{cc} p(b) & q(b) \\ \text{open} & \text{closed} \end{array} \end{array}$$

The open branch gives us a counter-model for entailment, that is, an interpretation that satisfies (is a model of) $\forall x(p(x) \vee q(x))$ and does not satisfy $\forall x p(x) \vee \forall x q(x)$. The interpretation has domain $D = \{\mathbf{a}, \mathbf{b}\}$, $p^{\mathfrak{I}} = \{\mathbf{b}\}$, $q^{\mathfrak{I}} = \{\mathbf{a}\}$.

(3) Use the tableaux procedure to prove that the following equivalences hold.

1. $\forall x(p(x) \wedge q(x)) \equiv \forall x p(x) \wedge \forall x q(x)$;
2. $\exists x(p(x) \vee q(x)) \equiv \exists x p(x) \vee \exists x q(x)$.

Solution. The given **equivalences** hold iff the following formulae are **valid**:

$$\forall x(p(x) \wedge q(x)) \leftrightarrow \forall x p(x) \wedge \forall x q(x); \tag{6}$$
$$\exists x(p(x) \vee q(x)) \leftrightarrow \exists x p(x) \vee \forall x q(x). \tag{7}$$

As above, we take the NFF of the negation of each formula and build a **closed** tableau. We do it for the formula (6). Its negation in NNF

$$\big(\forall x(p(x) \wedge q(x)) \wedge (\exists x \neg p(x) \vee \exists x \neg q(x))\big) \vee \big(\exists x(\neg p(x) \vee \neg q(x)) \wedge (\forall x p(x) \wedge \forall x q(x))\big)$$

will label the root of the tableau and require the application of a non-determinstinc OR-rule, which gives us two branches. We first build the first branch and then the second branch.

The first branch is as follows:

$$
\begin{array}{c}
[\forall x(p(x) \land q(x)) \land (\exists x \neg p(x) \lor \exists x \neg q(x))] \\
| \\
\forall x(p(x) \land q(x)) \\
[\exists x \neg p(x) \lor \exists x \neg q(x)] \\
\begin{array}{cc}
\exists x \neg p(x) & \exists x \neg q(x) \\
| & | \\
\neg p(a) & \neg q(b) \\
| & | \\
[p(a) \land q(a)] & [p(b) \land q(b)] \\
\forall x(p(x) \land q(x)) & \forall x(p(x) \land q(x)) \\
| & | \\
p(a) & p(b) \\
q(a) & q(b) \\
\text{closed} & \text{closed}
\end{array}
\end{array}
$$

The second branch is as follows:

$$
\begin{array}{c}
[\exists x(\neg p(x) \lor \neg q(x)) \land (\forall x p(x) \land \forall x q(x))] \text{ c} \\
| \\
[\exists x(\neg p(x) \lor \neg q(x))] \\
[\forall x p(x) \land \forall x q(x)] \\
| \\
\forall x p(x) \\
\forall x q(x) \\
| \\
\neg p(a) \lor \neg q(a) \\
\begin{array}{cc}
\neg p(a) & \neg q(a) \\
| & | \\
p(a) & q(a) \\
\forall x p(x) & \forall x q(x) \\
\text{closed} & \text{closed}
\end{array}
\end{array}
$$

Since both branches are closed, the tableau is closed, hence the equivalence holds.

1.2 Satisfiability and Set Satisfiability

(1) Use the tableau procedure to check whether the formula

$$\exists x\Big(\big(p(x) \land (q(x) \lor \neg p(x))\big) \land \big((\neg q(x) \lor r(x)) \land \neg r(x)\big)\Big),$$

in a language with only unary predicate symbols, is satisfiable.

Solution. We use the tableau procedure to prove that the formula is not satisfiable. The tableau will start with the given formula and continue as follows:

$$
\begin{array}{c}
p(a) \\
[q(a) \lor \neg p(a)] \\
[\neg q(a) \lor r(a)] \\
\neg r(a)
\end{array}
$$

$$
\begin{array}{cc}
\begin{array}{c} q(a) \\ \begin{array}{cc} \neg q(a) & r(a) \\ \text{closed} & \text{closed} \end{array} \end{array}
&
\begin{array}{c} \neg p(a) \\ \text{closed} \\ \\ \end{array}
\end{array}
$$

(2) Use the tableau procedure to check whether the set of formulae

$$\{\exists x p(x), \exists x(q(x) \lor \neg p(x)), \exists x(\neg q(x) \lor r(x)), \exists x \neg r(x)\},$$

in a language with only unary predicate symbols, is satisfiable (i.e., the conjuction of the formulae in the set is satisfiable).

Solution. We use the tableau procedure to prove that the conjuctive formula

$$\exists x p(x) \land \exists x(q(x) \lor \neg p(x)) \land \exists x(\neg q(x) \lor r(x)) \land \exists x \neg r(x)$$

is satisfiable. The tableau will start with the above formula and continue as follows:

$$
\begin{array}{c}
[\exists x p(x)] \\
[\exists x(q(x) \lor \neg p(x))] \\
[\exists x(\neg q(x) \lor r(x))] \\
[\exists x \neg r(x)] \\
| \\
p(a) \\
| \\
\neg r(b) \\
| \\
q(c) \lor \neg p(c)
\end{array}
$$

$$
\begin{array}{cc}
\begin{array}{c} q(c) \\ | \\ \neg q(d) \lor r(d) \\ \begin{array}{cc} \neg q(d) & r(d) \end{array} \end{array}
&
\begin{array}{c} \neg p(c) \\ | \\ \neg q(d) \lor r(d) \\ \begin{array}{cc} \neg q(d) & r(d) \end{array} \end{array}
\end{array}
$$

Any of the open branches gives rise to a model with domain $D = \{a, b, c, d\}$, e.g., the left-most branch gives the interpretation with $p^{\mathcal{I}} = \{a\}$, $q^{\mathcal{I}} = \{c\}$ and $r^{\mathcal{I}} = \emptyset$.

1.3 Formalisation and Entailment

(1) Consider the following argument.

> All fruit is tasty if it is not cooked. This apple is not cooked. Therefore it is tasty.

(i) Formalise the above argument as an entailment of the form $\Sigma \models \varphi$. *(ii)* By means of the tableau procedure for the satisfiability of $\Sigma \cup \{\neg\varphi\}$, determine if the entailment holds, or not. In the latter case, build a counter-model for the entailment $\Sigma \models \varphi$, that is, an interpretation $\mathcal{I}$

so that $\mathcal{I} \models \Sigma$ and $\mathcal{I} \not\models \varphi$ (to this end, use an open branch of the tableau and build a model for $\Sigma \cup \{\neg\varphi\}$).

Solution. Take a first-order language with t a unary predicate for tasty fruit, c another unary predicate for cooked, and a a constant for apple. Then:
"all not-cooked fruit is tasty" is formalised as $\forall x(\neg c(x) \to t(x))$;
"this apple is not cooked" is formalised as $\neg c(a)$;
"this apple is tasty (fruit)" is formalised as $t(a)$.

We use the tableau procedure to prove that $\forall x(\neg c(x) \to t(x)) \wedge \neg c(a) \models t(a)$ (entailment) by building a closed tableau for $\big(\forall x(\neg c(x) \to t(x)) \wedge \neg c(a)\big) \wedge \neg t(a)$. Firstly, we transform this into NNF: $(\forall x(c(x) \vee t(x)) \wedge \neg c(a)) \wedge \neg t(a)$. Secondly, we build a tableau starting with the NNF formula and continuing as follows.

$$
\begin{array}{c}
\forall x(c(x) \vee t(x)) \\
\neg c(a) \\
\neg t(a) \\
| \\
{[c(a) \vee t(a)]} \\
\forall x(c(x) \vee t(x)) \\
\begin{array}{cc}
c(a) & t(a) \\
\text{closed} & \text{closed}
\end{array}
\end{array}
$$

(2) Consider the following argument.

All fruit is tasty if it is not cooked. This apple is cooked. Therefore it is not tasty.

(i) Formalise the above argument as an entailment of the form $\Sigma \models \neg\varphi$. *(ii)* By means of the tableau procedure for the satisfiability of $\Sigma \cup \{\neg\neg\varphi\}$, determine if the entailment holds, or not. In the latter case, build a a counter-model for the entailment $\Sigma \models \neg\varphi$ (see hints above).

Solution. Take the same language as in the previous exercise. The resulting formulae are now as follows: "all not-cooked fruit is tasty" is formalised as $\forall x(\neg c(x) \to t(x))$;
"this apple is cooked" is formalised as $c(a)$;
"this apple is not tasty (fruit)" is formalised as $\neg t(a)$.

We use the tableau procedure to prove that $\forall x(\neg c(x) \to t(x)) \wedge c(a) \not\models \neg t(a)$ (entailment) by building an open tableau for $(\forall x(\neg c(x) \to t(x)) \wedge c(a)) \wedge \neg\neg t(a)$. Firstly, we transform this formula into the equivalent NNF: $(\forall x(c(x) \vee t(x)) \wedge c(a)) \wedge t(a)$. Secondly, we build a tableau with root labelled with the NNF formula and continuing as follows.

$$
\begin{array}{c}
\forall x(c(x) \vee t(x)) \\
c(a) \\
t(a) \\
| \\
{[c(a) \vee t(a)]} \\
\forall x(c(x) \vee t(x)) \\
\begin{array}{cc}
c(a) & t(a) \\
\text{open} & \text{open}
\end{array}
\end{array}
$$

Since the tableau is open and there is no clash that can be derived via the $\forall$-rule, the entailment does not hold.

There are two open branches and therefore two interpretations can be build, which happen to coincide. Namely, we obtain an interpretation $\mathfrak{I}$ with domain $D = \{\mathbf{a}\}$, in which $\mathbf{a}$ interprets the constant symbol a and we have $c^{\mathfrak{I}} = \{\mathbf{a}\}$ and $t^{\mathfrak{I}} = \{\mathbf{a}\}$.

Namely, the apple is cooked and tasty, and still all non-cooked fruit is tasty, too.

1.4 Formalisation and Set Satisfiability

1. Consider the following situation—a reformulation of the Russel paradox.

> In the Alpha town there were a barber man and a man that the barber shaved. However, any barber man of Alpha shaved all and only the men of Alpha who did not shave themselves.

(a) Formalize the statements above in a suitable first-order language with precisely two predicates: a unary predicate b for being a barber man of Alpha, and a binary predicate s for a man of Alpha that shaves a man of Alpha. (b) Using tableaux, show that the resulting set of formulae is unsatisfiable.

Hint: the resulting set of formulae is

$$\left\{ \begin{array}{l} \exists x\big(b(x) \land \exists y\, s(x,y)\big), \\ \forall x\Big(b(x) \to \forall y\big(s(x,y) \leftrightarrow \neg s(y,y)\big)\Big) \end{array} \right\}.$$

Solution. The resulting set is $\big\{\exists x\big(b(x) \land \exists y s(x,y)\big), \forall x\big(b(x) \to \forall y(\neg s(y,y) \leftrightarrow s(x,y))\big)\big\}$ where b and s are the predicates introduced in the exercise. We build a **closed** tableau for the conjunction of all the formulae in the set, transformed into equivalent NNF, thereby proving that the given set is **unsatisfiable**.

$$[\exists x\big(b(x) \wedge \exists y\, s(x,y)\big)]$$
$$\forall x\Big(\neg b(x) \vee \big(\forall y(s(y,y) \vee s(x,y)) \wedge \forall y(\neg s(y,y) \vee \neg s(x,y))\big)\Big)$$
$$|$$
$$[b(c) \wedge \exists y\, s(c,y)]$$
$$|$$
$$b(c)$$
$$[\exists y\, s(c,y)]$$
$$|$$
$$s(c,d)$$
$$|$$
$$[\neg b(c) \vee \big(\forall y(s(y,y) \vee s(c,y)) \wedge \forall y(\neg s(y,y) \vee \neg s(c,y))\big)]$$
$$[\neg b(d) \vee \big(\forall y(s(y,y) \vee s(d,y)) \wedge \forall y(\neg s(y,y) \vee \neg s(d,y))\big)]$$
$$\forall x\Big(\neg b(x) \vee \big(\forall y(s(y,y) \vee s(x,y)) \wedge \forall y(\neg s(y,y) \vee \neg s(x,y))\big)\Big)$$

$\neg b(c)$ closed | $\forall y(s(y,y) \vee s(c,y)) \wedge \forall y(\neg s(y,y) \vee \neg s(c,y))$

$$|$$
$$\forall y(s(y,y) \vee s(c,y))$$
$$\forall y(\neg s(y,y) \vee \neg s(c,y))$$
$$|$$
$$(s(c,c) \vee s(c,c))$$
$$(s(d,d) \vee s(c,d))$$
$$\forall y(s(y,y) \vee s(c,y))$$
$$|$$
$$(\neg s(c,c) \vee \neg s(c,c))$$
$$(\neg s(d,d) \vee \neg s(c,d))$$
$$\forall y(\neg s(y,y) \vee \neg s(c,y))$$

$s(c,c)$ | $s(c,c)$

$\neg s(c,c)$ closed | $\neg s(c,c)$ closed | $\neg s(c,c)$ closed | $\neg s(c,c)$ closed

2. Consider the following situation.

> In the Alpha town there were a barber man and a man that the barber shaved. However, any barber man of Alpha shaved all the men of Alpha who did not shave themselves.

(a) Formalize the statements above in the language of the previous exercise. (b) Using tableaux, show that the resulting set of formulae is satisfiable.

Solution. **(2)** The resulting set of formulae Σ is

$$\left\{ \begin{array}{l} \exists x\big(b(x) \wedge \exists y s(x,y)\big), \\ \forall x\big(b(x) \to \forall y(\neg s(y,y) \to s(x,y))\big) \end{array} \right\}.$$

where b and s are the predicates introduced in the exercise. We build a tableau with an open branch that defines a model for the conjunction of all the formulae in the set, thereby proving that this set of formulae is satisfiable.

$$[\exists x\big(b(x) \wedge \exists y\, s(x,y)\big)]$$
$$\forall x\big(\neg b(x) \vee \forall y(s(y,y) \vee s(x,y))\big)$$
$$[b(c) \wedge \exists y\, s(c,y)]$$
$$b(c)$$
$$[\exists y\, s(c,y)]$$
$$s(c,d)$$
$$[\neg b(c) \vee \forall y(s(y,y) \vee s(c,y))]$$
$$[\neg b(d) \vee \forall y(s(y,y) \vee s(d,y))]$$
$$\forall x\big(\neg b(x) \vee \forall y(s(y,y) \vee s(x,y))\big)$$

$\neg b(c)$ closed | $\forall y(s(y,y) \vee s(c,y))$

$\neg b(d)$ | $\forall y(s(y,y) \vee s(d,y))$ $\vdots$

$$s(c,c) \vee s(c,c) \equiv [s(c,c)]$$
$$[s(d,d) \vee s(c,d)]$$
$$\forall y(s(y,y) \vee s(c,y))$$
$$s(c,c)$$

$s(d,d)$ open | $s(c,d)$ open

The open branches define models of Σ, e.g., the one ending with $s(c,d)$ defines the interpretation $\mathfrak{I}$ with domain $D = \{\mathbf{c}, \mathbf{d}\}$, $s^{\mathfrak{I}} = \{(\mathbf{c},\mathbf{c}), (\mathbf{c},\mathbf{d})\}$ and $b^{\mathfrak{I}} = \{\mathbf{c}\}$, which is a model of Σ—verify it.

1.5 First Order Logic: Formalisation and Satifiability or Entailment

(1) Roncisvalle is a land of paladins. Astolfo and Rinaldo are two paladins of Roncisvalle.

(1) Each paladin decorates Astolfo or Rinaldo.

(2) No paladin decorates himself.

(3) *Therefore* Astolfo decorates a certain paladin.

(i) Using a suitable first order language and the knowledge you are given of Roncisvalle, formalise (1), (2), (3). Let T be the set consisting of (1) and (2). *(ii)* Establish whether $T \models (3)$ or not.

Solution. The first-order language has two constant symbols: r for Rinaldo, and a for Astolfo. The language has also a binary relation symbol, say R, that stands for "to decorate". Sentences (1), (2), $\neg$(3) are formalised as the following set T:

$$\begin{array}{r} (1) \\ (2) \\ \neg(3) \end{array} \qquad \left\{ \begin{array}{ll} \forall x & \big((R(x,a) \vee R(x,r))\big), \\ \forall x & \neg R(x,x), \\ \forall x & \neg R(a,x) \end{array} \right\}.$$

A tableau for this set is as follows:

$$
\begin{array}{c}
\forall x\big(R(x,a) \vee R(x,r)\big) \\
\forall x \neg R(x,x) \\
\forall x \neg R(a,x) \\
| \\
\neg R(a,a) \\
\neg R(a,r) \\
[\forall x \neg R(a,x)] \\
| \\
\neg R(a,a) \\
\neg R(r,r) \\
[\forall x \neg R(x,x)] \\
| \\
[R(a,a) \vee R(a,r)] \\
R(r,a) \vee R(r,r) \\
\forall x\big(R(x,a) \vee R(x,r)\big) \\
\begin{array}{cc}
\swarrow & \searrow \\
R(a,a) & R(a,r) \\
\text{closed} & \text{closed}
\end{array}
\end{array}
$$

(2) We are given the following statements concerning solids on a table.

(1) There is a small cube.

(2) Any small solid is on top of a solid.

(3) *Therefore* there exists a solid on top of a solid.

(i) Formalise (1) and (2) as a set Σ of formulae and (3) as φ. Then verify whether $\Sigma \models \varphi$ holds or not by means of the tableau procedure. In case it does not, give an interpretation that satisfies Σ and falsifies φ.

Solution. We are given the following statements concerning solids on a table.

(1) There is a small cube.

(2) Any small solid is on top of a solid.

(3) *Therefore* there exists a solid on top of a solid.

The sentences (1), (2) and (3) can be formalised in a language with: C unary predicate for being a cube (solid); S unary predicate for being small (solid); T binary predicate between two solids, one on top of the other. The formulae of the language formalising (1), (2) and $\neg$(3) are at the root of the following tableau:

$$
\begin{array}{c}
[\exists x\big(S(x) \land C(x)\big)] \\
\forall x \exists y\big(\neg S(x) \lor T(x,y)\big) \\
\forall x \forall y \neg T(x,y) \\
| \\
[S(a) \land C(a)] \\
| \\
S(a) \\
C(a) \\
| \\
[\exists y\big(\neg S(a) \lor T(a,y)\big)] \\
\forall x \exists y\big(\neg S(x) \lor T(x,y)\big) \\
| \\
[\neg S(a) \lor T(a,b)]
\end{array}
$$

$$
\begin{array}{cc}
\neg S(a) & T(a,b) \\
\text{closed} & | \\
 & \forall y \neg T(a,y) \\
 & \forall y \neg T(b,y) \\
 & \forall x \forall y \neg T(x,y) \\
 & | \\
 & \neg T(a,a) \\
 & \neg T(a,b) \\
 & \forall y \neg T(a,y) \\
 & \text{closed}
\end{array}
$$

Since the tableau for $\{(1),(2),\neg(3)\}$ is closed we can conclude that $\{(1),(2)\} \models (3)$.

(3) Consider the following properties concerning a directed graph:

(1) every node has an adjacent node;

(2) the edge relation is symmetric;

(3) the edge relation is transitive;

(4) the edge relation is not reflexive.

Using the tableau procedure decide whether the set is satisfiable (and, in particular, if there exists a graph satisfying all those properties).

Solution. Let E be a binary predicate symbole that denotes the edge relation. The given properties can be formalised as follows:

(1) $\forall x \exists y E(x,y)$,

(2) $\forall x \forall y(\neg E(x,y) \lor E(y,x))$,

(3) $\forall x \forall y \forall z(\neg E(x,y) \lor \neg E(y,z) \lor E(x,z))$,

(4) $\exists x \neg E(x,x)$.

In order to show that the above set is unsatisfiable (and hence there is no graph with such properties), we build a closed tableau for $\{(1),(2),(3),(4)\}$.

$$\forall x \exists y E(x,y)$$
$$\forall x \forall y(\neg E(x,y) \vee E(y,x))$$
$$\forall x \forall y \forall z(\neg E(x,y) \vee \neg E(y,z) \vee E(x,z))$$
$$[\exists x \neg E(x,x)]$$
$$|$$
$$\neg E(a,a)$$
$$|$$
$$[\exists y E(a,y)]$$
$$\forall x \exists y E(x,y)$$
$$|$$
$$E(a,b)$$
$$|$$
$$\forall y(\neg E(a,y) \vee E(y,a))$$
$$\forall y(\neg E(b,y) \vee E(y,b))$$
$$\forall x \forall y(\neg E(x,y) \vee E(y,x))$$
$$|$$
$$\neg E(a,a) \vee E(a,a)$$
$$[\neg E(a,b) \vee E(b,a)]$$
$$\forall y(\neg E(a,y) \vee E(y,a))$$

$$\neg E(a,b) \qquad\qquad\qquad\qquad E(b,a)$$
closed

$$|$$
$$\forall y \forall z(\neg E(a,y) \vee \neg E(y,z) \vee E(a,z))$$
$$\forall y \forall z(\neg E(b,y) \vee \neg E(y,z) \vee E(b,z))$$
$$\forall x \forall y \forall z(\neg E(x,y) \vee \neg E(y,z) \vee E(x,z))$$
$$|$$
$$\forall z(\neg E(a,a) \vee \neg E(a,z) \vee E(a,z))$$
$$\forall z(\neg E(a,b) \vee \neg E(b,z) \vee E(a,z))$$
$$\forall y \forall z(\neg E(a,y) \vee \neg E(y,z) \vee E(a,z))$$
$$|$$
$$[\neg E(a,b) \vee \neg E(b,a) \vee E(a,a)]$$
$$\neg E(a,b) \vee \neg E(b,b) \vee E(a,b)$$
$$\forall z(\neg E(a,b) \vee \neg E(b,z) \vee E(a,z))$$

$$\neg E(a,b) \qquad [\neg E(b,a) \vee E(a,a)]$$
closed

$$\neg E(b,a) \qquad E(a,a)$$
closed closed