

Satisfaction

To assign a True/False value to a first-order logic formula, we must both define a mathematical structure and assign values to the variables. We will then be able to define logical consequence.

1. True/false

In true/false logic, we assigned a value T/F to a formula by assigning each variable a value T/F. For a formula in a first-order language, things are a little more complicated. We must first define a structure in order to interpret the formula in that mathematical setting, and if there are free variables, we must also assign a mathematical value to each of them.

1.1. For all & There exists

The major innovation of first-order logic is, of course, the introduction of quantifiers. Consider a language $\mathcal{L}$ and a predicate $P(x)$. Fix a structure $\mathcal{S}$ for this language with domain E .

“ $\forall x P(x)$ ” is true iff for every $x \in E$, $P^{\mathcal{S}}(x)$ is true

“ $\exists x P(x)$ ” is true iff there is (at least) one $x \in E$ such that $P^{\mathcal{S}}(x)$ is true

Notice that we are obliged to define these notions using sentences in English, that is, using a metalanguage. The other important point is that this True/False evaluation has no meaning at the level of the language itself, but is carried out at the level of a structure. Recall that $P^{\mathcal{S}}(x)$ is the interpretation of the predicate $P(x)$ in the structure $\mathcal{S}$. Of course, the result may depend on the choice of structure.

Example.

- The formula

$$\exists x \quad x^2 < x$$

is false if we consider a structure with domain $E_1 = [1, +\infty[$, but true if the domain is $\mathbb{R}$ (for example, $(\frac{1}{2})^2 < \frac{1}{2}$).

- The formula

$$\forall x \quad (\exists y \quad y^2 = x)$$

is true if we consider the domain $E_1 = [1, +\infty[$, but false if the domain is $\mathbb{R}$ (for example, -1 is not the square of a real number).

Remark: if we have a formula $\mathcal{P}$ whose only free variable is x , then we can also determine whether the sentences “ $\forall x \mathcal{P}$ ” and “ $\exists x \mathcal{P}$ ” are true or false in the structure. As usual, with a slight abuse of notation, we will often write $\mathcal{P}(x)$ for a formula $\mathcal{P}$ to indicate that it depends on the variable x .

1.2. True/False for closed formulas

Fix a language $\mathcal{L}$. Recall that a variable is *free* in a formula if one of its occurrences is not within the scope of a quantifier over that variable. A formula with no free variable is a *closed formula*.

Definition.

Let $\mathcal{S}$ be a structure and let Q be a closed formula. If the interpretation of Q in $\mathcal{S}$ is true, then we say that Q is *satisfied* in $\mathcal{S}$ and write:

$$\models_{\mathcal{S}} Q$$

We also say that $\mathcal{S}$ is a *model* for Q . We say that Q is *satisfiable* if there is a structure $\mathcal{S}$ such that $\models_{\mathcal{S}} Q$.

Example.

Consider a suitable language and the formula:

$$Q: \quad \forall x \quad (x \times x = a) \rightarrow x = a$$

For the structure $\mathcal{S}_1$ with domain $E = \mathbb{R}$ and $a^{\mathcal{S}_1} = 0$, the formula is true, so indeed $\models_{\mathcal{S}_1} Q$.

On the other hand, for the structure $\mathcal{S}_2$ with domain $E = \mathbb{R}$, but this time with $a^{\mathcal{S}_2} = 1$, the formula is false (indeed, if $x^2 = 1$, then $x = +1$ or $x = -1$).

1.3. Assignment

Consider a language $\mathcal{L}$ and a structure $\mathcal{S}$ with domain E . If x is a variable of $\mathcal{L}$ and d an element of the set E , then $x \mapsto d$ denotes the *assignment* of the value d to every free occurrence of x in the formulas of the language. We write $Q[x \mapsto d]$ for the formula, interpreted in $\mathcal{S}$, in which the free occurrences of x are replaced by the value d .

For example, consider the formula $Q: "x^2 \geq 7"$ for a structure with domain $\mathbb{R}$. Since the variable x is free, it makes no sense to ask whether Q is true or false. After assignment, however, the formula $Q[x \mapsto 3]$ becomes " $3^2 \geq 7$ " and is therefore true, whereas $Q[x \mapsto 2]$ becomes " $2^2 \geq 7$ " and is false.

More generally, an *assignment* is a function $s: \text{Var} \rightarrow E$ that associates with each variable x_i of the language a value d_i in the domain: $x_i \mapsto d_i$. We write $Q[s]$ for the application of an assignment s to a formula Q . Let us emphasize that only occurrences of free variables are assigned values.

1.4. Formulas with free variables

Definition.

Let Q be a formula. We say that Q is *satisfied* in $\mathcal{S}$ under the assignment s if the interpretation $Q[s]$ is true. We then write $\models_{\mathcal{S}} Q[s]$, or also:

$$\models_{\mathcal{S},s} Q$$

Of course, the assignment matters only if there are free variables. Assignment is the counterpart of evaluation in true/false logic (where we replaced each propositional variable by a value T/F): here it is more general because x is replaced by any value from the set E (the domain).

Example.

Consider a language in which we can write the formula:

$$Q: \quad \forall x \quad y \leq x$$

The variable y is free. We will consider a structure $\mathcal{S}_1$ with domain $\mathbb{N}$ and a structure $\mathcal{S}_2$ with domain $\mathbb{R}$ (and the usual order relation). It makes no sense to ask whether $\mathcal{Q}$ is true, either in $\mathcal{S}_1$ or in $\mathcal{S}_2$.

- In $\mathcal{S}_1$, consider the assignment $s : y \mapsto 0$. We indeed have

$$\mathcal{Q}[y \mapsto 0] : \quad \forall x \in \mathbb{N} \quad 0 \leq x$$

and so the statement $\mathcal{Q}[y \mapsto 0]$ is true; hence $\models_{\mathcal{S}_1, y \mapsto 0} \mathcal{Q}$.

- In $\mathcal{S}_2$, the assignment $s : y \mapsto 0$ gives:

$$\mathcal{Q}[y \mapsto 0] : \quad \forall x \in \mathbb{R} \quad 0 \leq x$$

which is, of course, false.

- The assignment $y \mapsto 1$ gives false statements for both structures.
- Finally, let s be the assignment such that $x \mapsto 5$ and $y \mapsto 1$. Since x is not a free variable, s does not assign values to the occurrences of x in the formula, which becomes $\mathcal{Q}[s]$: “ $\forall x \quad 1 \leq x$ ”.

Validity

Definition.

Let $\mathcal{Q}$ be a formula of a language $\mathcal{L}$. We say that $\mathcal{Q}$ is a **validity** if, for every structure $\mathcal{S}$ and every assignment s , the interpretation of $\mathcal{Q}[s]$ in $\mathcal{S}$ is true. We then write:

$$\models \mathcal{Q}$$

Thus, this means that $\models_{\mathcal{S}, s} \mathcal{Q}$ for every $\mathcal{S}$ and s .

A validity is the analogue of a tautology in propositional logic. If $\models \neg \mathcal{Q}$, this means that $\mathcal{Q}$ is always false (for every $\mathcal{S}$ and s); such a formula $\mathcal{Q}$ is called a **contradiction** or is said to be **unsatisfiable**.

Example.

- “ $\forall x \quad x = x$ ” is a validity.
- “ $x = x$ ” is also a validity.
- “ $\forall x \quad P(x) \wedge Q(x) \rightarrow P(x)$ ” is a validity (we do not need to know how the predicates P and Q will be interpreted).
- “ $\exists x \exists y \quad x \neq y$ ” is not a validity. It is very often true, but not if we choose a structure whose domain has only one element, for example $E = \{0\}$, in which there cannot be two distinct elements.
- “ $x < x$ ” is not a validity (it is false over $\mathbb{R}$ if we interpret “ $<$ ” as strict inequality, but true if we interpret it as non-strict inequality).
- “ $\exists x \quad P(x) \wedge \neg P(x)$ ” is a contradiction. Even if we do not know how the predicate P will be interpreted, we can never have both $P(x)$ and its negation at the same time. (Note that the domain of a structure is nonempty; this is important here.)

2. Logical consequence

2.1. Definition

We generalize the notion of satisfaction as follows: if the assumptions $\mathcal{P}_1, \dots, \mathcal{P}_l$ are satisfied, then the conclusion $\mathcal{Q}$ is also satisfied.

Definition.

We say that $\mathcal{Q}$ is a **logical consequence** of $\mathcal{P}_1, \dots, \mathcal{P}_l$ if, for every structure $\mathcal{S}$ and every assignment s that make $\mathcal{P}_1, \dots, \mathcal{P}_l$ simultaneously true, $\mathcal{Q}$ is also true.

We then write:

$$\mathcal{P}_1, \dots, \mathcal{P}_l \models \mathcal{Q}$$

Similarly, if Γ is a set of formulas, we write $\Gamma \models \mathcal{Q}$.

Here is an important example:

$$\forall x \mathcal{P}(x) \models \exists x \mathcal{P}(x)$$

If a formula is true for every element, then it is true for at least one element. Note once again that it is important that we required the domains of structures always to be nonempty.

There are natural logical consequences inherited from propositional logic:

- $\mathcal{P} \wedge \mathcal{Q} \models \mathcal{P}$
- $\mathcal{P}, \mathcal{Q} \models \mathcal{P} \wedge \mathcal{Q}$
- $\mathcal{P}, \mathcal{P} \rightarrow \mathcal{Q} \models \mathcal{Q}$
- if $\mathcal{P} \models \mathcal{Q}$ and $\mathcal{Q} \models \mathcal{R}$, then $\mathcal{P} \models \mathcal{R}$

2.2. Logical equivalence

Definition.

If $\mathcal{P} \models \mathcal{Q}$ and $\mathcal{Q} \models \mathcal{P}$, we say that $\mathcal{P}$ and $\mathcal{Q}$ are **logically equivalent** and write:

$$\mathcal{P} \equiv \mathcal{Q}$$

When this is the case, it means that $\mathcal{P}$ and $\mathcal{Q}$ are satisfied at the same time. Another way to define logical equivalence would be $\models (\mathcal{P} \leftrightarrow \mathcal{Q})$.

Of course, we recover equivalences from propositional logic, for example:

- $\neg(\mathcal{P} \vee \mathcal{Q}) \equiv \neg\mathcal{P} \wedge \neg\mathcal{Q}$ (De Morgan's law),
- $\mathcal{P} \rightarrow \mathcal{Q} \equiv \neg\mathcal{P} \vee \mathcal{Q}$ (definition of implication),
- etc.

2.3. Equivalences involving quantifiers

Let us examine logical equivalences involving quantifiers.

Negation of quantifiers.

$$\neg(\forall x \mathcal{P}(x)) \equiv \exists x \neg \mathcal{P}(x)$$

$$\neg(\exists x \mathcal{P}(x)) \equiv \forall x \neg \mathcal{P}(x)$$

Roughly speaking, remember that the negation of “for every” is “there exists”, and conversely. These equivalences imply that if we have a formula $\mathcal{Q}$ containing quantifiers, then we can always find an equivalent formula containing only the quantifier “for every” (or only the quantifier “there exists”).

Example.

$$\begin{aligned} & \neg(\forall x \ P(x) \wedge Q(x)) \\ \equiv & \exists x \ \neg(P(x) \wedge Q(x)) \\ \equiv & \exists x \ (\neg P(x)) \vee (\neg Q(x)) \end{aligned}$$

$$\begin{aligned} & \neg(\exists x \ P(x) \rightarrow Q(x)) \\ \equiv & \forall x \ \neg(P(x) \rightarrow Q(x)) \\ \equiv & \forall x \ \neg((\neg P(x)) \vee Q(x)) \\ \equiv & \forall x \ P(x) \wedge (\neg Q(x)) \end{aligned}$$

Order of quantifiers.

We can interchange two consecutive “for every” quantifiers, or two consecutive “there exists” quantifiers:

$$\forall x \forall y \mathcal{P}(x, y) \equiv \forall y \forall x \mathcal{P}(x, y)$$

$$\exists x \exists y \mathcal{P}(x, y) \equiv \exists y \exists x \mathcal{P}(x, y)$$

On the other hand, in general we cannot interchange a “for every” and a “there exists”:

$$\forall x \exists y \mathcal{P}(x, y) \text{ is not equivalent to } \exists y \forall x \mathcal{P}(x, y)$$

For example:

$$\forall x \in \mathbb{R} \ \exists y \in \mathbb{R} \quad x + y = 0 \quad \text{is true,}$$

whereas

$$\exists y \in \mathbb{R} \ \forall x \in \mathbb{R} \quad x + y = 0 \quad \text{is false.}$$

Everyday examples can also make this clear: “every student has a phone number,” but it is not true that “there is one phone number shared by all students.”

Other useful equivalences.

$$\begin{aligned}
\forall x \ (P(x) \wedge Q(x)) &\equiv (\forall x \ P(x)) \wedge (\forall x \ Q(x)) \\
\exists x \ (P(x) \vee Q(x)) &\equiv (\exists x \ P(x)) \vee (\exists x \ Q(x)) \\
\forall x \ (P(x) \rightarrow Q) &\equiv (\exists x \ P(x)) \rightarrow Q \quad \text{if } x \text{ does not occur in } Q
\end{aligned}$$

Finally, we can rename a variable x as a new variable y (which does not occur in $P(x)$):

$$\forall x \ P(x) \equiv \forall y \ P[y/x]$$

2.4. Satisfaction of quantifiers

Let us define the satisfaction of quantifiers a little more formally:

$$\models_{S,s} \forall x \ P(x) \quad \text{iff} \quad \text{for every } d \in E, \models_{S,s[x \mapsto d]} P(x)$$

The notation $s[x \mapsto d]$ denotes the assignment that maps x to d and agrees with s on all other variables.

$$\models_{S,s} \exists x \ P(x) \quad \text{iff} \quad \text{there exists } d \in E \text{ such that } \models_{S,s[x \mapsto d]} P(x)$$

2.5. Assignment vs. substitution

This section is more technical and may be skipped on a first reading. The important point is that assignment and substitution are two different operations, but they are compatible.

Consider a language $\mathcal{L}$ and a structure S with domain E . Recall:

- the **substitution** $P[t/x]$ replaces every free occurrence of x by the term t . We start with a formula P of the language $\mathcal{L}$ and obtain a formula $P[t/x]$ that is still in the same language.
- the **assignment** $P[x \mapsto d]$ replaces every free occurrence of x by the value $d \in E$. The formula P is still a formula of the language $\mathcal{L}$, but $P[x \mapsto d]$ is a formula interpreted in the structure S .

Once we have clearly understood the difference, it is reassuring to know that combining substitution followed by assignment is equivalent to combining them in the other order:

$$\models_{S,s} P[t/x] \quad \text{iff} \quad \models_{S,s[x \mapsto t^{S,s}]} P$$

Example.

Let $\mathcal{L} = (0, 1, \leq, +)$ be a language. Write $2 = 1 + 1$ and $3 = 2 + 1$. Consider the formula:

$$P(x): \quad \forall y \quad x \leq y + 3$$

Let the term be $t = x + 1$.

Consider the structure S with domain $E = \mathbb{N}$ naturally associated with $\mathcal{L}$.

- The substitution $P[t/x]$ gives the following formula, which remains in the language $\mathcal{L}$:

$$P[t/x]: \quad \forall y \quad x + 1 \leq y + 3$$

- Start again from P . The assignment $s : x \mapsto 1$ transforms a formula of $\mathcal{L}$ into a formula interpreted in the structure:

$$P[x \mapsto 1]: \quad \forall y \in \mathbb{N} \quad 1 \leq y + 3$$

- Substitution followed by assignment. Start from P , substitute t for x to obtain $P[t/x]$ as above,

and then assign 1 to the variable x . We then obtain:

$$\mathcal{P}[t/x][x \mapsto 1]: \quad \forall y \in \mathbb{N} \quad 2 \leq y + 3$$

- Assignment first. Start with the term $t = x + 1$ and assign the value 1 to x in this term. We obtain the term $t^{S,s} = 2 \in E$. Now, in $\mathcal{P}$, replace every free occurrence of x by $t^{S,s}$. We obtain the same formula as before:

$$\mathcal{P}[x \mapsto 2]: \quad \forall y \in \mathbb{N} \quad 2 \leq y + 3$$

2.6. Summary of the concepts

Notation	Terminology	Description
$\models_S Q$	Satisfaction (closed formula). Model	The closed formula Q is true in the structure S . We say that S is a model of Q .
$\models_{S,s} Q$	Satisfaction under an assignment	The formula Q is true in S under the assignment s .
	Satisfiable formula	Formula true in at least one structure (and under an assignment if necessary).
$\models Q$	Validity	Formula Q true in every structure and under every assignment.
$\models \neg Q$	Contradiction	Formula Q always false, in every structure and under every assignment.
$\mathcal{P}_1, \dots, \mathcal{P}_l \models Q$	Logical consequence	Every structure and assignment that simultaneously satisfy the $\mathcal{P}_i$ also satisfy Q .
$\mathcal{P} \equiv Q$	Logical equivalence	$\mathcal{P} \models Q$ and $Q \models \mathcal{P}$. For every structure S and every assignment s , $\mathcal{P}$ and Q have the same truth value.

Exercises

True/false

Exercise 1. For each of the following closed formulas, determine whether it is true or false when interpreted in the different structures; that is, does $\models_S Q$ hold? Consider $\mathcal{L} = (a, f(x, y))$ and the formulas:

$$Q_1: \quad \forall x \exists y \quad f(x, y) = a$$

$$Q_2: \quad \forall x \forall y \quad f(x, y) = a \rightarrow (x = a) \vee (y = a)$$

$$Q_3: \quad \forall x \quad (x \neq a) \rightarrow (\exists y \quad f(x, y) \neq a)$$

Here are the structures:

$$\mathcal{S}_i : \mathbb{N}, a = 0, f(x, y) = x + y$$

$$\mathcal{S}_{ii} : \mathbb{N}, a = 0, f(x, y) = xy$$

$$\mathcal{S}_{iii} : \mathbb{N}, a = 1, f(x, y) = xy$$

Exercise 2. Consider the language $\mathcal{L} = (<, f(x))$ and the structure $\mathcal{S}$ with domain $\mathbb{R}$ and $f(x) = e^x$. For the following formulas and given assignments, determine whether the interpretation is true or false; that is, does $\models_{\mathcal{S}, s} \mathcal{Q}$ hold?

$$1. \mathcal{Q}_1 : \quad \forall x \quad y < f(x) \quad s_1 : y \mapsto 0; \text{ then } s_2 : y \mapsto e$$

$$2. \mathcal{Q}_2 : \quad \exists x \quad x < f(y) \quad s_1 : y \mapsto 0; \text{ then } s_2 : y \mapsto e$$

$$3. \mathcal{Q}_3 : \quad \neg(f(x) < f(y)) \quad s : x \mapsto 0, y \mapsto e$$

Exercise 3. Consider the language $\mathcal{L} = (0, 1, <, +, \times)$ and the naturally associated structure $\mathcal{S}$ with domain $\mathbb{R}$. For each of the following formulas, find all assignments s to the free variables that make the formula $\mathcal{Q}$ true in the structure:

$$1. \mathcal{Q}_1 : \quad \forall x \quad (x \neq 0) \rightarrow x \times y \neq 0$$

$$2. \mathcal{Q}_2 : \quad \exists x \quad x \times x = y$$

$$3. \mathcal{Q}_3 : \quad \exists y \quad x < y \times y$$

For each of the following formulas, represent in the plane $\mathbb{R}^2$ the set of values that, when assigned to (x, y) , make the formula $\mathcal{Q}$ true:

$$4. \mathcal{Q}_4 : \quad x \times x < y$$

$$5. \mathcal{Q}_5 : \quad (\exists x \quad x + y = 0) \vee (\forall y \quad x \times y = 0)$$

$$6. \mathcal{Q}_6 : \quad (x \times x = y \times y) \rightarrow (x \neq y)$$

Exercise 4. Are the following formulas validities (true for every $\mathcal{S}$ and s), contradictions (false for every $\mathcal{S}$ and s), or neither (in which case justify your answer)?

$$1. (\forall x P(x)) \rightarrow (\exists x P(x))$$

$$2. (\exists x P(x)) \rightarrow (\forall x P(x))$$

$$3. \forall x \exists y \quad (x = y) \wedge (x \neq y)$$

$$4. \forall x \quad Q(x, y) \rightarrow Q(y, x)$$

$$5. x = y$$

$$6. \exists y \quad (x = y) \vee (x \neq y)$$

Logical consequence

Exercise 5. Determine whether each of the following formulas $\mathcal{Q}$ is a logical consequence of $\mathcal{P}_1, \dots, \mathcal{P}_l$; that is, does $\mathcal{P}_1, \dots, \mathcal{P}_l \models \mathcal{Q}$ hold?

$$1. \mathcal{P} : \forall x \quad x < y, \quad \mathcal{Q} : \exists x \quad x < y$$

$$2. \mathcal{P} : \neg(x = y), \quad \mathcal{Q} : x < y$$

$$3. \mathcal{P}_1 : \forall x \quad P(x), \quad \mathcal{P}_2 : \exists y \quad Q(y), \quad \mathcal{Q} : \exists x \quad P(x) \wedge Q(x)$$

$$4. \mathcal{P}_1 : \exists x \quad P(x), \quad \mathcal{P}_2 : \exists x \quad Q(x), \quad \mathcal{Q} : \exists x \quad P(x) \wedge Q(x)$$

Exercise 6. Write a sentence in English that explains or justifies each of the following properties:

1. $\mathcal{P} \models \mathcal{P}$
2. $\mathcal{P} \models \mathcal{Q}$ iff $\models (\mathcal{P} \rightarrow \mathcal{Q})$
3. if $\mathcal{P} \models \mathcal{Q}$ then $\mathcal{P}, \mathcal{R} \models \mathcal{Q}$
4. if $\mathcal{P} \models \mathcal{Q}$ and $\mathcal{Q} \models \mathcal{R}$ then $\mathcal{P} \models \mathcal{R}$

Exercise 7. For each of the following formulas (given their natural interpretation), write the negation of the statement and determine whether the original statement is true or false.

1. $\exists x \in \mathbb{R} \quad x^2 = 2$
2. $\forall x \in \mathbb{N} \quad x > 2$
3. $\exists x \in]0, +\infty[\quad \forall y \in]0, +\infty[\quad x + y = 1$
4. $\forall z \in \mathbb{C} \quad z \neq 0 \rightarrow z^2 \neq 0$
5. $\forall x \in \mathbb{R} \quad \forall y \in \mathbb{R} \quad xy = 0 \rightarrow (x = 0) \vee (y = 0)$
6. $\forall x \in \mathbb{R} \quad x > 1 \rightarrow (\exists y \in \mathbb{R} \quad (y > 1) \wedge (x > y))$

Exercise 8. For each pair of formulas $\mathcal{P}$, $\mathcal{Q}$, determine whether we have the logical consequence $\mathcal{P} \models \mathcal{Q}$, or $\mathcal{Q} \models \mathcal{P}$, or the logical equivalence $\mathcal{P} \equiv \mathcal{Q}$, or none of these.

1. $\mathcal{A} \vee \mathcal{B}$ and $\mathcal{B} \vee \mathcal{A}$
2. $\neg(\mathcal{A} \vee \mathcal{B})$ and $(\neg \mathcal{A}) \vee (\neg \mathcal{B})$
3. $\exists x \neg \mathcal{A}$ and $\forall x \mathcal{A}$
4. $\forall x \mathcal{A} \rightarrow \mathcal{B}$ and $\exists x \mathcal{B} \wedge (\neg \mathcal{A})$
5. $\forall x \mathcal{A}$ and $\mathcal{A}$
6. $\exists x \mathcal{A}$ and $\mathcal{A}$

Exercise 9. Show that the following are **not** logical equivalences:

1. $\exists x \mathcal{P} \wedge \mathcal{Q}$ and $(\exists x \mathcal{P}) \wedge (\exists x \mathcal{Q})$
2. $\forall x \mathcal{P} \vee \mathcal{Q}$ and $(\forall x \mathcal{P}) \wedge (\forall x \mathcal{Q})$
3. $\forall x \exists y \mathcal{P}$ and $\exists y \forall x \mathcal{P}$
4. $\forall x (\mathcal{P} \rightarrow \mathcal{Q})$ and $(\exists x \mathcal{P}) \rightarrow \mathcal{Q}$

Exercise 10. Let $\mathcal{L}$ be a language. An *axiom* is a closed formula $\mathcal{A}$ that we require to be true in the structures under consideration; that is, we consider only structures $\mathcal{S}$ such that $\models_{\mathcal{S}} \mathcal{A}$.

1. What can we say about a structure $\mathcal{S}$ that satisfies the axiom

$$\mathcal{A}: \quad \forall x \forall y \quad x = y$$

2. Same question for:

$$\mathcal{A}: \quad \exists x \exists y \quad (x \neq y) \wedge (\forall z \quad (z = x) \vee (z = y))$$