Does the model $\mathcal{M}$ satisfy this formula? Well, it does not; for we may choose a for x and b for y. Since (a, a) is in the set $\mathsf{loves}^{\mathcal{M}}$ and (b, a) is in the set $\mathsf{loves}^{\mathcal{M}}$, we would need that the latter does not hold since it is the interpretation of $\mathsf{loves}(y, \mathsf{alma})$; this cannot be.

And what changes if we modify $\mathcal{M}$ to $\mathcal{M}'$, where we keep A and $\mathsf{alma}^{\mathcal{M}}$, but redefine the interpretation of loves as $\mathsf{loves}^{\mathcal{M}'} \stackrel{\text{def}}{=} \{(b, a), (c, b)\}$? Well, now there is exactly one lover of Alma's lovers, namely c; but c is not one of Alma's lovers. Thus, the formula in (2.8) holds in the model $\mathcal{M}'$.

2.4.2 Semantic entailment

In propositional logic, the semantic entailment $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$ holds iff: whenever all $\phi_1, \phi_2, \ldots, \phi_n$ evaluate to $\mathtt{T}$, the formula ψ evaluates to $\mathtt{T}$ as well. How can we define such a notion for formulas in predicate logic, considering that $\mathcal{M} \vDash_l \phi$ is indexed with an environment?

Definition 2.20 Let Γ be a (possibly infinite) set of formulas in predicate logic and ψ a formula of predicate logic.

1. Semantic entailment $\Gamma \vDash \psi$ holds iff for all models $\mathcal{M}$ and look-up tables l, whenever $\mathcal{M} \vDash_l \phi$ holds for all $\phi \in \Gamma$, then $\mathcal{M} \vDash_l \psi$ holds as well.
2. Formula ψ is satisfiable iff there is some model $\mathcal{M}$ and some environment l such that $\mathcal{M} \vDash_l \psi$ holds.
3. Formula ψ is valid iff $\mathcal{M} \vDash_l \psi$ holds for all models $\mathcal{M}$ and environments l in which we can check ψ.
4. The set Γ is consistent or satisfiable iff there is a model $\mathcal{M}$ and a look-up table l such that $\mathcal{M} \vDash_l \phi$ holds for all $\phi \in \Gamma$.

In predicate logic, the symbol $\vDash$ is overloaded: it denotes model checks '$\mathcal{M} \vDash \phi$' and semantic entailment '$\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$.' Computationally, each of these notions means trouble. First, establishing $\mathcal{M} \vDash \phi$ will cause problems, if done on a machine, as soon as the universe of values A of $\mathcal{M}$ is infinite. In that case, checking the sentence $\forall x\, \psi$, where x is free in ψ, amounts to verifying $\mathcal{M} \vDash_{[x \mapsto a]} \psi$ for infinitely many elements a.

Second, and much more seriously, in trying to verify that $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$ holds, we have to check things out for *all possible models*, all models which are equipped with the right structure (i.e. they have functions and predicates with the matching number of arguments). This task is impossible to perform mechanically. This should be contrasted to the situation in propositional logic, where the computation of the truth tables for the propositions involved was the basis for computing this relationship successfully.

However, we can sometimes reason that certain semantic entailments are valid. We do this by providing an argument that does not depend on the actual model at hand. Of course, this works only for a very limited number of cases. The most prominent ones are the *quantifier equivalences* which we already encountered in the section on natural deduction. Let us look at a couple of examples of semantic entailment.

Example 2.21 The justification of the semantic entailment

$$\forall x\,(P(x) \to Q(x)) \vDash \forall x\,P(x) \to \forall x\,Q(x)$$

is as follows. Let $\mathcal{M}$ be a model satisfying $\forall x\,(P(x) \to Q(x))$. We need to show that $\mathcal{M}$ satisfies $\forall x\,P(x) \to \forall x\,Q(x)$ as well. On inspecting the definition of $\mathcal{M} \vDash \psi_1 \to \psi_2$, we see that we are done if not every element of our model satisfies P. Otherwise, every element does satisfy P. But since $\mathcal{M}$ satisfies $\forall x\,(P(x) \to Q(x))$, the latter fact forces every element of our model to satisfy Q as well. By combining these two cases (i.e. either all elements of $\mathcal{M}$ satisfy P, or not) we have shown that $\mathcal{M}$ satisfies $\forall x\,P(x) \to \forall x\,Q(x)$.

What about the converse of the above? Is

$$\forall x\,P(x) \to \forall x\,Q(x) \vDash \forall x\,(P(x) \to Q(x))$$

valid as well? Hardly! Suppose that $\mathcal{M}'$ is a model satisfying $\forall x\,P(x) \to \forall x\,Q(x)$. If A' is its underlying set and $P^{\mathcal{M}'}$ and $Q^{\mathcal{M}'}$ are the corresponding interpretations of P and Q, then $\mathcal{M}' \vDash \forall x\,P(x) \to \forall x\,Q(x)$ simply says that, if $P^{\mathcal{M}'}$ equals A', then $Q^{\mathcal{M}'}$ must equal A' as well. However, if $P^{\mathcal{M}'}$ does not equal A', then this implication is vacuously true (remember that $\mathtt{F} \to \cdot = \mathtt{T}$ no matter what $\cdot$ actually is). In this case we do not get any additional constraints on our model $\mathcal{M}'$. After these observations, it is now easy to construct a counter-example model. Let $A' \stackrel{\text{def}}{=} \{a, b\}$, $P^{\mathcal{M}'} \stackrel{\text{def}}{=} \{a\}$ and $Q^{\mathcal{M}'} \stackrel{\text{def}}{=} \{b\}$. Then $\mathcal{M}' \vDash \forall x\,P(x) \to \forall x\,Q(x)$ holds, but $\mathcal{M}' \vDash \forall x\,(P(x) \to Q(x))$ does not.

2.4.3 The semantics of equality

We have already pointed out the open-ended nature of the semantics of predicate logic. Given a predicate logic over a set of function symbols $\mathcal{F}$ and a set of predicate symbols $\mathcal{P}$, we need only a non-empty set A equipped with concrete functions or elements $f^{\mathcal{M}}$ (for $f \in \mathcal{F}$) and concrete predicates $P^{\mathcal{M}}$ (for $P \in \mathcal{P}$) in A which have the right arities agreed upon in our specification. Of course, we also stressed that most models have natural interpretations of