VARIABLE SUBSTITUTION

1. Unsafe substitution

1.1. **Definition.** Let $\mathcal{A}$ be a first-order signature, X, Y be two sets of variables. A **variable substitution** from X to Y is a function $\sigma : X \to \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$. Given such a σ, we define, for each term $t \in \mathcal{L}^X_{\mathsf{term}}(\mathcal{A})$, a term $t[\sigma] \in \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$, called the **substitution of σ into t**, as follows:

$$\begin{aligned} x[\sigma] &:= \sigma(x) && \text{for } x \in X, \\ f(t_1, \dots, t_n)[\sigma] &:= f(t_1[\sigma], \dots, t_n[\sigma]) && \text{for } f \in \mathcal{A}^n_{\mathsf{fun}} \text{ and } t_1, \dots, t_n \in \mathcal{L}^X_{\mathsf{term}}(\mathcal{A}). \end{aligned}$$

We then define, for each formula $\phi \in \mathcal{L}^X_{\mathsf{form}}(\mathcal{A})$, a formula $\phi[\sigma] \in \mathcal{L}^Y_{\mathsf{form}}(\mathcal{A})$, called the **substitution of σ into ϕ**, inductively as follows:

$$\begin{aligned} R(t_1, \dots, t_n)[\sigma] &:= R(t_1[\sigma], \dots, t_n[\sigma]) && \text{for } R \in \mathcal{A}^n_{\mathsf{rel}} \text{ (or } =), \ t_1, \dots, t_n \in \mathcal{L}^X_{\mathsf{term}}(\mathcal{A}), \\ (\phi \wedge \psi)[\sigma] &:= \phi[\sigma] \wedge \psi[\sigma], \\ (\phi \vee \psi)[\sigma] &:= \phi[\sigma] \vee \psi[\sigma], \\ (\neg\phi)[\sigma] &:= \neg\phi[\sigma], \\ \top[\sigma] &:= \top, \\ \bot[\sigma] &:= \bot, \\ (\exists x\, \phi)[\sigma] &:= \exists x\, \phi[\sigma\langle x \mapsto x\rangle] && \text{for } \phi \in \mathcal{L}^{X \cup \{x\}}_{\mathsf{form}}(\mathcal{A}). \end{aligned}$$

Here, as in Definition 2.12 from first-order logic, $\sigma\langle x \mapsto x\rangle : X \cup \{x\} \to Y \cup \{x\}$ means σ extended with the assignment $x \mapsto x$, replacing any previous value of $\sigma(x)$.

1.2. **Example.** Consider the $\mathcal{A}_{\mathsf{ordfield}}$-formula

$$\phi := (x \le y) \wedge \exists y\, (x + y = z).$$

Informally speaking, the two occurrences of y don't refer to the same thing: the second occurrence is bound by the $\exists y$, hence is "inaccessible from the outside". This is reflected in the substitution

$$\begin{aligned} \phi[x \mapsto 0,\, y \mapsto 1,\, z \mapsto z] &= (x \le y)[x \mapsto 0,\, y \mapsto 1,\, z \mapsto z] \wedge (\exists y\, (x + y = z))[x \mapsto 0,\, y \mapsto 1,\, z \mapsto z] \\ &= (x \le y)[x \mapsto 0,\, y \mapsto 1,\, z \mapsto z] \wedge \exists y\, (x + y = z)[x \mapsto 0,\, y \mapsto y,\, z \mapsto z] \\ &= (0 \le 1) \wedge \exists y\, (0 + y = z) \end{aligned}$$

where in the middle step we used $(x \mapsto 0,\, y \mapsto 1,\, z \mapsto z)\langle y \mapsto y\rangle = (x \mapsto 0,\, y \mapsto y,\, z \mapsto z)$.

As in this example, it is common to want to substitute for only a few variables, while leaving all others unchanged. We therefore adopt the following

1.3. **Convention.** When we write $\phi[\sigma]$, we allow σ to be defined on only a subset of the free variables of ϕ, in which case we implicitly extend σ via the identity on the remaining variables.

1.4. **Example.** With the same ϕ as in the previous example,

$$\begin{aligned} \phi[x \mapsto y,\, y \mapsto z] &= (x \le y)[x \mapsto y,\, y \mapsto z] \wedge (\exists y\, (x + y = z))[x \mapsto y,\, y \mapsto z] \\ &= (x \le y)[x \mapsto y,\, y \mapsto z] \wedge \exists y\, (x + y = z)[x \mapsto y] \\ &= (y \le z) \wedge \exists y\, (y + y = z). \end{aligned}$$

2. Substitution and interpretation

There is an elegant conceptual way to view (parts of) variable substitution in terms of other basic notions in first-order logic. Note that we may view a set of terms $\mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$ as an $\mathcal{A}$-structure, where each function symbol $f \in \mathcal{A}$ is interpreted as the syntactic f on terms, i.e., as "itself":

$$f^{\mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})}(t_1, \ldots, t_n) := f(t_1, \ldots, t_n).$$

(This "term model" was used in proving the completeness theorem; see Lemma 3.41 in the notes.) Thus, we may regard a substitution $\sigma : X \to \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$ as a variable assignment. Now comparing Definition 1.1 to the definition of interpretation of terms (2.7 in the first-order logic notes) reveals

$$t[\sigma] = t^{\mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})}_X(\sigma). \tag{2.1}$$

This perspective allows us to give slick proofs of several basic facts about substitution:

2.2. **Proposition** (interpretation of substituted terms)**.** For a substitution $\sigma : X \to \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$, $\mathcal{A}$-structure $\mathcal{M}$, variable assignment $\alpha : Y \to M$, and term $t \in \mathcal{L}^X_{\mathsf{term}}(\mathcal{A})$, we have

$$t[\sigma]^{\mathcal{M}}_Y(\alpha) = t^{\mathcal{M}}_X(\sigma^{\mathcal{M}}_Y(\alpha)) \tag{$*$}$$

where $\sigma^{\mathcal{M}}(\alpha) = \sigma^{\mathcal{M}}_Y(\alpha) : X \to M$ is given by interpreting each substituted term $\sigma(x)$ in $\mathcal{M}$:

$$\sigma^{\mathcal{M}}_Y(\alpha)(x) := \sigma(x)^{\mathcal{M}}_Y(\alpha).$$

Proof 1. By induction on t.

- For $t = x \in X$, we have $x[\sigma]^{\mathcal{M}}(\alpha) = \sigma(x)^{\mathcal{M}}(\alpha) = \sigma^{\mathcal{M}}(\alpha)(x) = x^{\mathcal{M}}(\sigma^{\mathcal{M}}(\alpha))$.
- For $t = f(t_1, \ldots, t_n)$ where the claim holds for $t_1, \ldots, t_n$, we have

$$\begin{aligned} f(t_1, \ldots, t_n)[\sigma]^{\mathcal{M}}(\alpha) &= f(t_1[\sigma], \ldots, t_n[\sigma])^{\mathcal{M}}(\alpha) \\ &= f^{\mathcal{M}}(t_1[\sigma]^{\mathcal{M}}(\alpha), \ldots, t_n[\sigma]^{\mathcal{M}}(\alpha)) \\ &= f^{\mathcal{M}}(t_1^{\mathcal{M}}(\sigma^{\mathcal{M}}(\alpha)), \ldots, t_n^{\mathcal{M}}(\sigma^{\mathcal{M}}(\alpha))) \quad \text{by IH} \\ &= f(t_1, \ldots, t_n)^{\mathcal{M}}(\sigma^{\mathcal{M}}(\alpha)). \end{aligned}$$

□

Proof 2. Observe that by definition of interpretation of terms (2.7 from first-order logic), the map

$$\begin{aligned} h : \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A}) &\longrightarrow \mathcal{M} \\ s &\longmapsto s^{\mathcal{M}}(\alpha) \end{aligned}$$

is a homomorphism, thus preserves the interpretation of terms, i.e.,

$$t[\sigma]^{\mathcal{M}}(\alpha) = h(t[\sigma]) = h(t^{\mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})}(\sigma)) = t^{\mathcal{M}}(h \circ \sigma) = t^{\mathcal{M}}(\sigma^{\mathcal{M}}(\alpha)).$$

□

2.3. **Corollary** (double substitution into terms)**.** For substitutions $\sigma : X \to \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$, $\tau : Y \to \mathcal{L}^Z_{\mathsf{term}}(\mathcal{A})$ and a term $t \in \mathcal{L}^X_{\mathsf{term}}(\mathcal{A})$, we have

$$t[\sigma][\tau] = t[\sigma[\tau]]$$

where $\sigma[\tau] : X \to \mathcal{L}^Z_{\mathsf{term}}(\mathcal{A})$ is given by substituting τ into each output of σ:

$$\sigma[\tau](x) := \sigma(x)[\tau].$$

Proof. Take $\mathcal{M} := \mathcal{L}^Z_{\mathsf{term}}(\mathcal{A})$ and $\alpha := \tau$ above; then $(*)$ becomes

$$t[\sigma]^{\mathcal{L}^Z_{\mathsf{term}}(\mathcal{A})}(\tau) = t^{\mathcal{L}^Z_{\mathsf{term}}(\mathcal{A})}(\sigma^{\mathcal{L}^Z_{\mathsf{term}}(\mathcal{A})}(\tau)),$$

which by (2.1) (applied three times) becomes the desired equation. □

2.4. **Exercise.**

(a) For substitution into *quantifier-free* formulas, we may describe them in a similar manner. Let $\mathcal{L}^X_{\mathrm{at}}(\mathcal{A}) \subseteq \mathcal{L}^X_{\mathsf{form}}(\mathcal{A})$ denote the atomic (first-order) formulas. Then a quantifier-free first-order $\mathcal{A}$-formula is the same thing as a propositional $\mathcal{L}^X_{\mathrm{at}}(\mathcal{A})$-formula, hence a term in the signature $\{\wedge, \vee, \neg, \top, \bot\}$ (called the signature $\mathcal{A}_{\mathsf{Bool}}$ of **Boolean algebras**; see Exercise 4.17 in propositional logic).

For a quantifier-free $\phi \in \mathcal{L}(\mathcal{L}^X_{\mathrm{at}}(\mathcal{A})) = \mathcal{L}^{\mathcal{L}^X_{\mathrm{at}}(\mathcal{A})}_{\mathsf{term}}(\mathcal{A}_{\mathsf{Bool}})$ and substitution $\sigma : X \to \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$, describe $\phi[\sigma]$ as $\phi[\tau]$ for some $\tau : \mathcal{L}^X_{\mathrm{at}}(\mathcal{A}) \to \mathcal{L}^{\mathcal{L}^Y_{\mathrm{at}}(\mathcal{A})}_{\mathsf{term}}(\mathcal{A}_{\mathsf{Bool}})$, and use this to prove

$$\begin{gathered}\phi[\sigma]^{\mathcal{M}}_Y(\alpha) = \phi^{\mathcal{M}}_X(\sigma^{\mathcal{M}}_Y(\alpha)),\\ \text{i.e., } \mathcal{M} \models_\alpha \phi[\sigma] \iff \mathcal{M} \models_{\sigma^{\mathcal{M}}(\alpha)} \phi,\\ \phi[\sigma][\tau] = \phi[\sigma[\tau]].\end{gathered}$$

(b) Find counterexamples to all the above statements when ϕ has quantifiers. [See Example 1.4.]

2.5. **Exercise** (free variables of substituted terms)**.**

(a) Let $\mathcal{A}$ be a signature, X be a set of variables. Define an $\mathcal{A}$-structure on $\mathcal{P}(X)$, such that the free variables map $\mathrm{FV} : \mathcal{L}^X_{\mathsf{term}}(\mathcal{A}) \to \mathcal{P}(X)$ becomes an $\mathcal{A}$-homomorphism.

(b) Using this, show that for $\sigma : X \to \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$ and $t \in \mathcal{L}^X_{\mathsf{term}}(\mathcal{A})$, we have

$$\mathrm{FV}(t[\sigma]) = \bigcup_{x \in \mathrm{FV}(t)} \mathrm{FV}(\sigma(x)).$$

(c) Show the same for quantifier-free $\phi \in \mathcal{L}^X_{\mathsf{form}}(\mathcal{A})$ in place of t.

(d) Give a counterexample when ϕ has quantifiers.

3. Safe substitution

3.1. **Definition.** For $\phi \in \mathcal{L}^X_{\mathsf{form}}(\mathcal{A})$ and $\sigma : X \to \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$, the substitution $\phi[\sigma]$ is **safe** if, informally speaking, whenever we substitute a term past a quantifier $\exists x$, that term does not contain x as a free variable. Formally, this is defined by induction on ϕ:

$$\begin{aligned} R(t_1, \dots, t_n)[\sigma] &\text{ is always safe,}\\ (\phi \wedge \psi)[\sigma], (\phi \vee \psi)[\sigma] \text{ safe } &:\iff \phi[\sigma], \psi[\sigma] \text{ are,}\\ (\neg\phi)[\sigma] \text{ safe } &:\iff \phi[\sigma] \text{ is,}\\ \top[\sigma], \bot[\sigma] &\text{ are always safe,}\\ (\exists x\, \phi)[\sigma] \text{ safe } &:\iff \underbrace{\forall y \in \mathrm{FV}(\exists x\, \phi) = \mathrm{FV}(\phi) \setminus \{x\}\, (x \notin \mathrm{FV}(\sigma(y)))}_{x \text{ is not captured}}, \text{ and } \phi[\sigma\langle x \mapsto x\rangle] \text{ safe.}\end{aligned}$$

In the last case, if the first condition fails, i.e., there is some $y \in \mathrm{FV}(\exists x\, \phi) = \mathrm{FV}(\phi) \setminus \{x\}$ such that $x \in \mathrm{FV}(\sigma(y))$, we say that the substitution of $y \mapsto \sigma(y)$ into $\exists x\, \phi$ **captures** the free x in $\sigma(y)$.

(Substitution into a term is always considered safe, since terms do not bind variables.)

3.2. **Example.** The substitution in Example 1.4 is *not* safe, because the substitution of $\sigma : x \mapsto y$ into $\exists y\, (x + y = z)$ captures y (because $x \in \mathrm{FV}(\exists y\, (x + y = z))$ and $y \in \mathrm{FV}(\sigma(x))$).

Safe substitutions are the ones which have the intended meaning of "substitution" from informal mathematical practice. Formally, this means that the kind of strange behavior in Exercise 2.4 does not occur:

3.3. **Proposition** (interpretation of substituted formulas)**.** For a substitution $\sigma : X \to \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$, $\mathcal{A}$-structure $\mathcal{M}$, variable assignment $\alpha : Y \to M$, and formula $\phi \in \mathcal{L}^X_{\mathsf{form}}(\mathcal{A})$, if $\phi[\sigma]$ is safe, then

$$\phi[\sigma]^{\mathcal{M}}(\alpha) = \phi^{\mathcal{M}}(\sigma^{\mathcal{M}}(\alpha)),$$
$$\text{i.e., } \mathcal{M} \models_\alpha \phi[\sigma] \iff \mathcal{M} \models_{\sigma^{\mathcal{M}}(\alpha)} \phi.$$

Proof. By induction on ϕ. The atomic and connective cases are straightforward (as in Exercise 2.4).

- For $\exists x\, \phi \in \mathcal{L}^X_{\mathsf{form}}(\mathcal{A})$, assuming the claim holds for $\phi \in \mathcal{L}^{X\cup\{x\}}_{\mathsf{form}}(\mathcal{A})$, we have

$$\begin{aligned}(\exists x\, \phi)[\sigma]^{\mathcal{M}}(\alpha) &= (\exists x\, \phi[\sigma\langle x \mapsto x\rangle])^{\mathcal{M}}(\alpha)\\ &= \max_{a\in M} \phi[\sigma\langle x \mapsto x\rangle]^{\mathcal{M}}(\alpha\langle x \mapsto a\rangle)\\ &= \max_{a\in M} \phi^{\mathcal{M}}(\sigma\langle x \mapsto x\rangle^{\mathcal{M}}(\alpha\langle x \mapsto a\rangle)) \quad \text{by IH;} \qquad (*)\end{aligned}$$

  we want to show

$$= \max_{a\in M} \phi(\sigma^{\mathcal{M}}(\alpha)\langle x \mapsto a\rangle) = (\exists x\, \phi)^{\mathcal{M}}(\sigma^{\mathcal{M}}(\alpha)). \qquad (\dagger)$$

  We have

$$\begin{aligned}\sigma^{\mathcal{M}}(\alpha)\langle x \mapsto a\rangle : X \cup \{x\} &\longrightarrow M\\ x &\longmapsto a\\ X \setminus \{x\} \ni y &\longmapsto \sigma^{\mathcal{M}}(\alpha)(y) = \sigma(y)^{\mathcal{M}}(\alpha),\end{aligned}$$

  while

$$\begin{aligned}\sigma\langle x \mapsto x\rangle^{\mathcal{M}}(\alpha\langle x \mapsto a\rangle) : X \cup \{x\} &\longrightarrow M\\ x &\longmapsto \sigma\langle x \mapsto x\rangle(x)^{\mathcal{M}}(\alpha\langle x \mapsto a\rangle) = x^{\mathcal{M}}(\alpha\langle x \mapsto a\rangle) = a\\ X \setminus \{x\} \ni y &\longmapsto \sigma\langle x \mapsto x\rangle(y)^{\mathcal{M}}(\alpha\langle x \mapsto a\rangle) = \sigma(y)^{\mathcal{M}}(\alpha\langle x \mapsto a\rangle);\end{aligned}$$

  if $y \in \mathrm{FV}(\phi)$, then this last interpretation $\sigma(y)^{\mathcal{M}}(\alpha\langle x \mapsto a\rangle)$ is the same as $\sigma(y)^{\mathcal{M}}(\alpha)$ above, since by the safety assumption, $x \notin \mathrm{FV}(\sigma(y))$ and so the interpretation $\sigma(y)^{\mathcal{M}}(\alpha)$ does not depend on $\alpha(x)$. Thus $(*) = (\dagger)$, since the two variable assignments agree on all those variables which actually occur free in ϕ. □

3.4. **Exercise** (double substitution into formulas)**.** Show that for substitutions $\sigma : X \to \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$ and $\tau : Y \to \mathcal{L}^Z_{\mathsf{term}}(\mathcal{A})$ and a formula $\phi \in \mathcal{L}^X_{\mathsf{form}}(\mathcal{A})$, if $\phi[\sigma]$ is safe, then

$$\phi[\sigma][\tau] = \phi[\sigma[\tau]].$$

3.5. **Remark.** It is possible to derive a slightly weaker version of this statement (namely, up to α-equivalence; see the following section) from a suitably generalized version of Proposition 3.3, similar to how Corollary 2.3 is a special case of Proposition 2.2 by interpreting in a term structure. Namely, since we are now interpreting formulas, not just terms, we need to consider structures in which "truth values" of relations may belong to some kind of $\mathcal{A}_{\mathsf{Bool}}$-structure, rather than $\{0, 1\}$. More precisely, there needs to be one $\mathcal{A}_{\mathsf{Bool}}$-structure for each possible variable set X, in which X-ary formulas take values, along with operations between these structures used for interpreting $\exists$. Such a structure is called a **hyperdoctrine**, and is well beyond the scope of this course.

3.6. **Exercise** (free variables of substituted formulas; cf. Exercise 2.5)**.** Show that for $\sigma : X \to \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$ and $\phi \in \mathcal{L}^X_{\mathsf{form}}(\mathcal{A})$, if $\phi[\sigma]$ is safe, then

$$\mathrm{FV}(\phi[\sigma]) = \bigcup_{x\in\mathrm{FV}(\phi)} \mathrm{FV}(\sigma(x)).$$

4. α-EQUIVALENCE

The way to deal with unsafe substitutions is familiar from informal mathematical practice: we change the bound variables to avoid clashes.

4.1. **Example.** The gamma function in real analysis is defined by

$$\Gamma(x) := \int_0^\infty t^{x-1}e^{-t}\,dt.$$

If we want to evaluate $\Gamma(t)$ in some context where t is a free variable, of course we should not blindly substitute x with t in the above formula; rather, we should first change the bound t to e.g., s:

$$\Gamma(x) = \int_0^\infty s^{x-1}e^{-s}\,ds,$$

whence

$$\Gamma(t) = \int_0^\infty s^{t-1}e^{-s}\,ds.$$

4.2. **Definition.** Two formulas ϕ, ψ are **α-equivalent**,[1] denoted $\phi \equiv_\alpha \psi$, if they may be converted into each other by repeatedly changing variables bound by $\exists$: namely, $\exists x\,\phi$ may be changed to $\exists y\,\phi[x \mapsto y]$, provided this substitution is safe and y does not already occur free in ϕ. In practice, this is achieved by ensuring that y occurs neither free nor bound in ϕ.

Formally, we need to define the binary relation $\equiv_\alpha$ in several steps. First, we define the relation $\sim_\alpha$ ("immediate α-equivalence") to consist of exactly the following pairs of formulas:

$$\exists x\,\phi \sim_\alpha \exists y\,\phi[x \mapsto y] \quad \text{where } y \notin \mathrm{FV}(\phi) \cup \{x\} \text{ and } \phi[x \mapsto y] \text{ is safe.}$$

We then define $\approx_\alpha$ ("one-step α-equivalence") to mean that two subformulas occurring in the same position are $\sim_\alpha$, which formally means the binary relation generated inductively via:

$$\begin{aligned}
\psi \sim_\alpha \phi &\implies \phi \approx_\alpha \psi,\\
\phi \approx_\alpha \psi &\implies \phi \wedge \theta \approx_\alpha \psi \wedge \theta,\\
\phi \approx_\alpha \psi &\implies \theta \wedge \phi \approx_\alpha \theta \wedge \psi,\\
\phi \approx_\alpha \psi &\implies \phi \vee \theta \approx_\alpha \psi \vee \theta,\\
\phi \approx_\alpha \psi &\implies \theta \vee \phi \approx_\alpha \theta \vee \psi,\\
\phi \approx_\alpha \psi &\implies \neg\phi \approx_\alpha \neg\psi,\\
\phi \approx_\alpha \psi &\implies \exists x\,\phi \approx_\alpha \exists x\,\psi.
\end{aligned}$$

Finally, we define $\equiv_\alpha$ to be the reflexive and transitive closure of $\sim_\alpha$:

$$\phi \equiv_\alpha \psi \;:\Longleftrightarrow\; \exists \phi_0, \phi_1, \ldots, \phi_n \text{ s.t. } \phi = \phi_0 \approx_\alpha \phi_1 \approx_\alpha \cdots \approx_\alpha \phi_n = \psi.$$

(Thus, when $n = 1$, this just means $\phi \approx_\alpha \psi$; when $n = 0$, it means $\phi = \psi$.)

4.3. **Example.** Recalling the formula from Example 1.4, we have

$$\exists y\,(x + y = z) \sim_\alpha \exists w\,(x + y = z)[y \mapsto w] = \exists w\,(x + w = z),$$

since the substitution $(x + y = z)[y \mapsto w]$ is clearly safe. Thus

$$(x \le y) \wedge \exists y\,(x + y = z) \approx_\alpha (x \le y) \wedge \exists w\,(x + w = z)$$

(hence also $\equiv_\alpha$). However,

$$\exists y\,(x + y = z) \not\sim_\alpha \exists x\,(x + y = z)[y \mapsto x] = \exists x\,(x + x = z)$$

since $x \in \mathrm{FV}(x + y = z)$.

[1] Here the α is not a variable assignment; it is just part of the terminology.

4.4. **Example.** We have

$$\forall x\, \exists y\, (x + y = 0) \equiv_\alpha \forall y\, \exists x\, (y + x = 0).$$

Since there are two quantifiers whose variables changed, these formulas cannot be $\approx_\alpha$; we need at least two steps. But two steps are not enough: we cannot immediately change the y in $\exists y\, (x + y = 0)$ to x, since $x \in \mathrm{FV}(x + y = 0)$. And we cannot immediately change the x in $\forall x$ to y either, since the substitution $(\exists y\, (x + y = 0))[x \mapsto y]$ is not safe. Instead, we need to go through a third variable:

$$\exists y\, (x + y = 0) \sim_\alpha \exists z\, (x + z = 0),$$

whence

(a) $$\forall x\, \exists y\, (x + y = 0) \approx_\alpha \forall x\, \exists z\, (x + z = 0);$$

and now the substitution $(\exists z\, (x + z = 0))[x \mapsto y]$ is safe, so

(b) $$\forall x\, \exists z\, (x + z = 0) \sim_\alpha \forall y\, \exists z\, (y + z = 0)$$

(hence also $\approx_\alpha$); finally, for similar reasons as in (a),

(c) $$\forall y\, \exists z\, (y + z = 0) \approx_\alpha \forall y\, \exists x\, (y + x = 0),$$

whence by chaining together (a), (b), and (c) we get the desired $\equiv_\alpha$.

We now state the key properties of $\equiv_\alpha$. The proofs are quite tedious and deferred to Section 5.

4.5. **Proposition.** $\equiv_\alpha$ is an equivalence relation on the set of all $\mathcal{A}$-formulas.

The next few results say that "everything important about formulas is well-defined modulo $\equiv_\alpha$":

4.6. **Proposition.** $\equiv_\alpha$ is a congruence relation with respect to the operations $\wedge, \vee, \neg$, as well as $\exists x$.

4.7. **Proposition.** If $\phi \equiv_\alpha \psi$, then $\mathrm{FV}(\phi) = \mathrm{FV}(\psi)$.

4.8. **Proposition.** If $\phi \equiv_\alpha \psi$, and both have free variables from X, then $\phi \mathrel{=\!|\!\models}_X \psi$.

4.9. **Proposition.** If $\phi \equiv_\alpha \psi$, both have free variables from X, and $\sigma : X \to \mathcal{L}^Y_{\mathsf{term}}(\mathcal{A})$ with both $\phi[\sigma], \psi[\sigma]$ safe, then $\phi[\sigma] \equiv_\alpha \psi[\sigma]$.

We now have the *raison d'être* for α-equivalence:

4.10. **Exercise.**

(a) Inductively define the set of **bound variables** $\mathrm{BV}(\phi)$ of a formula ϕ.
[For the answer, see the proof of Proposition 4.11 below.]

(b) Prove that if y gets captured during a substitution $\phi[\sigma]$, then

$$y \in \mathrm{BV}(\phi) \cap \bigcup_{x \in \mathrm{FV}(\phi)} \mathrm{FV}(\sigma(x)).$$

(c) Prove that for any variable substitution σ (even if unsafe), $\mathrm{BV}(\phi[\sigma]) = \mathrm{BV}(\phi)$.

4.11. **Proposition.** For any formula ϕ and infinite X, there is $\phi' \equiv_\alpha \phi$ such that $\mathrm{BV}(\phi') \subseteq X$.

4.12. **Corollary.** For any formula ϕ and substitution σ, there is $\phi' \equiv_\alpha \phi$ such $\phi'[\sigma]$ is safe. Thus, for each σ, the *safe* substitution operation $\phi \mapsto \phi[\sigma]$ (which is only defined for some ϕ) descends to a well-defined operation on α-equivalence classes of formulas $[\phi]_\alpha$.

4.13. **Exercise.**

(a) Show that Exercise 4.10(b) provides only a rough upper bound on which variables are captured, by giving an example of a ϕ, σ and y belonging to that set, yet y is not captured.

(b) Give a more precise characterization of variable capture, by defining, for each formula ϕ and $x \in \mathrm{FV}(\phi)$, a set of variables $\mathrm{BV}_x(\phi)$, such that for any substitution σ, $\phi[\sigma]$ captures y iff

$$y \in \bigcup_{x \in \mathrm{FV}(\phi)} (\mathrm{BV}_x(\phi) \cap \mathrm{FV}(\sigma(x))).$$

Finally, given ϕ, ψ, to figure out whether or not $\phi \equiv_\alpha \psi$, the definition is not that useful, since there could be an arbitrarily long "path" of $\approx_\alpha$'s between them. The following alternate characterization says that we may instead traverse the inductive structure of ϕ, ψ themselves:

4.14. **Proposition** (structural characterization of $\equiv_\alpha$)**.** Let $\phi \equiv_\alpha \psi$.

(a) If ϕ is atomic, $\top$, or $\bot$, then $\phi = \psi$.

(b) If ϕ is a conjunction, then so is ψ, and we have

$$\phi = \phi' \wedge \phi'' \equiv_\alpha \psi' \wedge \psi'' = \psi$$

for some $\phi' \equiv_\alpha \psi'$ and $\phi'' \equiv_\alpha \psi''$.

(c) If ϕ is a disjunction, then so is ψ, and we have

$$\phi = \phi' \vee \phi'' \equiv_\alpha \psi' \vee \psi'' = \psi$$

for some $\phi' \equiv_\alpha \psi'$ and $\phi'' \equiv_\alpha \psi''$.

(d) If ϕ is a negation, then so is ψ, and we have

$$\phi = \neg\phi' \equiv_\alpha \neg\psi' = \psi$$

for some $\phi' \equiv_\alpha \psi'$.

(e) If ϕ is an existential, then so is ψ, and we have

$$\phi = \exists x\, \phi' \sim_\alpha \exists z\, \phi'[x \mapsto z] \equiv_\alpha \exists z\, \psi'[y \mapsto z] \sim_\alpha \exists y\, \psi' = \psi$$

for some ϕ', ψ', such that $\phi'[x \mapsto z] \equiv_\alpha \psi'[y \mapsto z]$ for *any* variable z witnessing both of the outer $\sim_\alpha$'s (i.e., $z \notin \mathrm{FV}(\phi') \cup \mathrm{FV}(\psi') \cup \{x, y\}$ and $\phi'[x \mapsto z], \psi'[y \mapsto z]$ are safe).

4.15. **Example.** We claim that

$$\forall x\, (0 \le x \to \exists y\, (y \cdot y = x)) \not\equiv_\alpha \forall z\, (0 \le z \to \exists z\, (z \cdot z = z)).$$

Informally, this is because the last x on the LHS "refers" to the outermost quantifier, while the last z on the RHS, which is in the same "position", refers to the innermost quantifier; but if two formulas are α-equivalent, then the variables in the same "positions" must either both be free and equal, or both be bound by quantifiers in the same "positions". (See also Section 6 below.)

Formally, we may disprove the above α-equivalence by using Proposition 4.14 to "zone in" on the position of the variables that refer to two different things. Indeed, if the $\equiv_\alpha$ held, then by Proposition 4.14(e) (along with (d), using that $\forall$ is an abbreviation for $\neg\exists\neg$), we must have

$$0 \le y \to \exists y\, (y \cdot y = w) \equiv_\alpha 0 \le y \to \exists z\, (z \cdot z = z);$$

now by (c),

$$\exists y\, (y \cdot y = w) \equiv_\alpha \exists z\, (z \cdot z = z),$$

which is false because the two sides have different free variables (Proposition 4.7), or because we can apply (e) again to get

$$x \cdot x = w \equiv_\alpha x \cdot x = x$$

which is false by (a).

5. Proofs about α-equivalence

Proof of Proposition 4.5. The main thing to check is that $\sim_\alpha$ is symmetric. We claim that if

$$\exists x\, \phi \sim_\alpha \exists y\, \phi[x \mapsto y] \quad \text{where } y \notin \mathrm{FV}(\phi) \cup \{x\} \text{ and } \phi[x \mapsto y] \text{ is safe,}$$

then also

$$\exists y\, \phi[x \mapsto y] \sim_\alpha \exists x\, \phi[x \mapsto y][y \mapsto x] \quad \text{where } x \notin \mathrm{FV}(\phi[x \mapsto y]) \cup \{y\} \text{ and } \phi[x \mapsto y][y \mapsto x] \text{ is safe,}$$

which is enough because $\phi[x \mapsto y][y \mapsto x] = \phi[x \mapsto x] = \phi$ by Exercise 3.4. We have

$$\mathrm{FV}(\phi[x \mapsto y]) \subseteq (\mathrm{FV}(\phi) \setminus \{x\}) \cup \{y\}$$

by Exercise 3.6, which clearly does not contain x under our assumptions on y. Thus the only thing to check is that $\phi[x \mapsto y][y \mapsto x]$ is safe, which we do by induction on ϕ:

- If ϕ is atomic, $\top$, or $\bot$, then this is trivial.
- Suppose that $y \notin \mathrm{FV}(\phi) \cup \{x\}$ and $\phi[x \mapsto y]$ safe imply that $\phi[x \mapsto y][y \mapsto x]$ is safe, and same for ψ. Then $y \notin \mathrm{FV}(\phi \wedge \psi) \cup \{x\} = \mathrm{FV}(\phi) \cup \mathrm{FV}(\psi) \cup \{x\}$ and $(\phi \wedge \psi)[x \mapsto y]$ safe imply that $y \notin \mathrm{FV}(\phi) \cup \{x\}$, $y \notin \mathrm{FV}(\psi) \cup \{x\}$, and $\phi[x \mapsto y], \psi[x \mapsto y]$ are safe, which by the IH imply that $(\phi \wedge \psi)[x \mapsto y][y \mapsto x] = \phi[x \mapsto y][y \mapsto x] \wedge \psi[x \mapsto y][y \mapsto x]$ is safe.
- Similarly for $\vee, \neg$.
- Finally, suppose $y \notin \mathrm{FV}(\phi) \cup \{x\}$ and $\phi[x \mapsto y]$ safe imply $\phi[x \mapsto y][y \mapsto x]$ safe. Then if

  $$y \notin \mathrm{FV}(\exists z\, \phi) \cup \{x\} \text{ and } (\exists z\, \phi)[x \mapsto y] \text{ safe,}$$

  the former means $y \notin (\mathrm{FV}(\phi) \setminus \{z\}) \cup \{x\}$, while the latter means either:
  - $x \notin \mathrm{FV}(\exists z\, \phi)$, in which case $(\exists z\, \phi)[x \mapsto y] = \exists z\, \phi$, and so further substituting $y \mapsto x$ also has no effect by the former assumption, hence is trivially safe; or
  - $x \in \mathrm{FV}(\exists z\, \phi) = \mathrm{FV}(\phi) \setminus \{z\}$, in which case safety means $y \neq z$ and $\phi[x \mapsto y]$ is safe, whence the former assumption becomes $y \notin \mathrm{FV}(\phi) \cup \{x, z\}$, whence by the IH, $\phi[x \mapsto y][y \mapsto x]$ is safe, whence so is $(\exists z\, \phi)[x \mapsto y][y \mapsto x] = (\exists z\, \phi[x \mapsto y])[y \mapsto x] = \exists z\, \phi[x \mapsto y][y \mapsto x]$ since $x \neq z$.

  In both cases, we get that $(\exists z\, \phi)[x \mapsto y][y \mapsto x]$ is safe, as desired.

This completes the proof that $\sim_\alpha$ is symmetric. It is now obvious from the definition of $\approx_\alpha$ that it is also symmetric, hence $\equiv_\alpha$ is symmetric, by reversing the "path" of $\approx_\alpha$'s; and clearly $\equiv_\alpha$ is reflexive and transitive. □

Proof of Proposition 4.6. If $\phi \equiv_\alpha \psi$, then there is a "path" $\phi = \phi_0 \approx_\alpha \phi_1 \approx_\alpha \cdots \approx_\alpha \phi_n = \psi$; by definition of $\approx_\alpha$, we then have $\phi \wedge \theta \approx_\alpha \phi_1 \wedge \theta \approx_\alpha \cdots \approx_\alpha \psi \wedge \theta$, and similarly for the other connectives and $\exists x$. □

Proof of Proposition 4.7. Suppose $\exists x\, \phi \sim_\alpha \exists y\, \phi[x \mapsto y]$ with $y \notin \mathrm{FV}(\phi) \cup \{x\}$, $\phi[x \mapsto y]$ safe. Then

$$\mathrm{FV}(\exists y\, \phi[x \mapsto y]) = \mathrm{FV}(\phi[x \mapsto y]) \setminus \{y\}.$$

If $x \notin \mathrm{FV}(\phi)$, this is $\mathrm{FV}(\phi) \setminus \{y\} = \mathrm{FV}(\phi) = \mathrm{FV}(\exists x\, \phi)$ since $x, y \notin \mathrm{FV}(\phi)$. Otherwise, it is

$$\begin{aligned} &= ((\mathrm{FV}(\phi) \setminus \{x\}) \cup \{y\}) \setminus \{y\} && \text{by Exercise 3.6} \\ &= \mathrm{FV}(\phi) \setminus \{x\} && \text{since } y \notin \mathrm{FV}(\phi) \\ &= \mathrm{FV}(\exists x\, \phi). \end{aligned}$$

So $\sim_\alpha$ implies same free variables; by a trivial induction, so do $\approx_\alpha$ and $\equiv_\alpha$. □

Proof of Proposition 4.8. Suppose $\exists x\,\phi \sim_\alpha \exists y\,\phi[x \mapsto y]$ with $y \notin \mathrm{FV}(\phi) \cup \{x\}$, $\phi[x \mapsto y]$ safe. Then

$$\begin{aligned} \mathcal{M} \models_\alpha \exists x\,\phi &\iff \exists a \in M \text{ s.t. } \mathcal{M} \models_{\alpha\langle x \mapsto a\rangle} \phi, \\ \mathcal{M} \models_\alpha \exists y\,\phi[x \mapsto y] &\iff \exists a \in M \text{ s.t. } \mathcal{M} \models_{\alpha\langle y \mapsto a\rangle} \phi[x \mapsto y] \\ &\iff \exists a \in M \text{ s.t. } \mathcal{M} \models_{(x \mapsto y)^{\mathcal{M}}(\alpha\langle y \mapsto a\rangle)} \phi \end{aligned}$$

by Proposition 3.3; recalling that (by Convention 1.3) $x \mapsto y$ really means $\mathrm{id}_X\langle x \mapsto y\rangle$, we get that $(x \mapsto y)^{\mathcal{M}}(\alpha\langle y \mapsto a\rangle) : X \cup \{y\} \to M$ maps $x \mapsto y^{\mathcal{M}}(\alpha\langle y \mapsto a\rangle) = a$ and all other $z \in \mathrm{FV}(\phi) \setminus \{x\}$ to $z^{\mathcal{M}}(\alpha\langle y \mapsto a\rangle) = \alpha(z)$, since $y \notin \mathrm{FV}(\phi)$. So the first and third RHSs above are equivalent.

Thus $\sim_\alpha$ implies semantically equivalent; by a trivial induction, so do $\approx_\alpha$ and $\equiv_\alpha$. □

Proof of Proposition 4.11. By induction on ϕ.

- If ϕ is atomic, $\top$, or $\bot$, then $\mathrm{BV}(\phi) = \varnothing$, so $\phi' := \phi$ works
- If $\phi' \equiv_\alpha \phi$ and $\psi' \equiv_\alpha \psi$ with $\mathrm{BV}(\phi'), \mathrm{BV}(\psi') \subseteq X$, then $\phi' \wedge \psi' \equiv_\alpha \phi \wedge \psi$ (using Proposition 4.6) with $\mathrm{BV}(\phi' \wedge \psi') = \mathrm{BV}(\phi') \cup \mathrm{BV}(\psi') \subseteq X$.
- Similarly for $\vee, \neg$.
- Finally, suppose the claim holds for ϕ; we prove it for $\exists x\,\phi$. Pick any $x' \in X \setminus \{x\} \setminus \mathrm{FV}(\phi)$, and find $\phi' \equiv_\alpha \phi$ with $\mathrm{BV}(\phi') \subseteq X \setminus \{x'\}$. Then $\phi'[x \mapsto x']$ is safe by Exercise 4.10(b), and $x' \notin \mathrm{FV}(\phi) \cup \{x\}$, so we have $\exists x'\,\phi'[x \mapsto x'] \sim_\alpha \exists x\,\phi' \equiv_\alpha \exists x\,\phi$ with $\mathrm{BV}(\exists x'\,\phi'[x \mapsto x']) = \mathrm{BV}(\phi'[x \mapsto x']) \cup \{x'\} = \mathrm{BV}(\phi') \cup \{x'\} \subseteq X$ by Exercise 4.10(c). □

Proof of Corollary 4.12. By Proposition 4.11, let $\phi' \equiv_\alpha \phi$ with $\mathrm{BV}(\phi')$ disjoint from the finite set $\bigcup_{x \in \mathrm{FV}(\phi)} \mathrm{FV}(\sigma(x)) = \bigcup_{x \in \mathrm{FV}(\phi')} \mathrm{FV}(\sigma(x))$ (by Proposition 4.7); then no variable gets captured by $\phi'[\sigma]$ by Exercise 4.10(b). □

5.1. **Exercise.** A formula ϕ satisfies the **Barendregt variable convention** if the variables bound by different quantifiers in it are all distinct from each other and from all free variables.

(a) Define what this means precisely.
(b) Prove that any formula ϕ is α-equivalent to one satisfying the Barendregt variable convention.

It remains to prove Propositions 4.9 and 4.14.

5.2. **Exercise.** Try to prove Proposition 4.9 directly, similarly to the proof of Proposition 4.8, by first proving it for $\sim_\alpha$, then for $\approx_\alpha$, then for $\equiv_\alpha$. You will probably run into a "chicken-and-egg" type of obstacle.

Intuitively speaking, the issue here is that the $\equiv_\alpha$ between the two original formulas may be derived via a "path" of $\approx_\alpha$'s which is highly disorganized. This is why it's convenient to simultaneously prove Proposition 4.14, which tells us that the $\equiv_\alpha$ may be derived in a much more organized manner which reflects the inductive structure of the formulas in question.

Proof of Propositions 4.9 and 4.14. Since $\phi \equiv_\alpha \psi$, let

$$\phi = \phi_0 \approx_\alpha \phi_1 \approx_\alpha \cdots \approx_\alpha \phi_n = \psi.$$

In Proposition 4.14(e), we will first prove the weaker statement where z may be any variable outside of *some* finite set (while (e) says it is enough to take *any* z not appearing free in ϕ', ψ' and also making $\phi'[x \mapsto z], \psi'[y \mapsto z]$ safe). We proceed by induction on the height of ϕ. (Any other numerical measure of "size" of a formula, according to which a subformula is strictly smaller, and a formula obtained by substituting variables for variables has the same size, would work just as well.)

- If ϕ is atomic, $\top$, or $\bot$, there is no clause in Definition 4.2 of $\approx_\alpha$ which yields $\phi = \phi_0 \approx_\alpha \phi_1$; thus the above sequence must have length $n = 0$, i.e., $\phi = \psi$, whence clearly $\phi[\sigma] = \psi[\sigma]$.

- If $\phi = \phi' \wedge \phi''$, then by considering the possibilities in Definition 4.2 for $\phi = \phi_0 \approx_\alpha \phi_1$, we must have $\phi_1 = \phi_1' \wedge \phi_1''$ where either $\phi_0' \approx_\alpha \phi_1'$ and $\phi_0'' = \phi_1''$, or $\phi_0' = \phi_1'$ and $\phi_0'' \approx_\alpha \phi_1''$; in either case, we get $\phi_0' \equiv_\alpha \phi_1'$ and $\phi_0'' \equiv_\alpha \phi_1''$. Now apply similar reasoning to $\phi_1 \approx_\alpha \phi_2$, $\phi_2 \approx_\alpha \phi_3$, etc., to eventually get that $\psi = \psi' \wedge \psi''$ with $\phi' \equiv_\alpha \psi'$ and $\phi'' \equiv_\alpha \psi''$. Thus

$$\begin{aligned}\phi[\sigma] &= \phi'[\sigma] \wedge \phi''[\sigma] \\ &\equiv_\alpha \psi'[\sigma] \wedge \psi''[\sigma] \quad \text{by IH and Proposition 4.6} \\ &= \psi[\sigma].\end{aligned}$$

- The cases $\vee$ and $\neg$ are similar.
- Finally, suppose $\phi = \exists x\, \phi'$. There are two possibilities for $\phi = \phi_0 \approx_\alpha \phi_1$: either

$$\phi = \exists x\, \phi' \sim_\alpha \exists y\, \phi'[x \mapsto y] = \phi_1 \quad \text{with } y \notin \mathrm{FV}(\phi') \cup \{x\} \text{ and } \phi'[x \mapsto y] \text{ safe},$$

or

$$\phi = \exists x\, \phi' \approx_\alpha \exists x\, \phi_1' = \phi_1 \quad \text{with } \phi' \approx_\alpha \phi_1';$$

in both cases, call $\exists x_1 \phi_1' := \phi_1$. Similarly breaking down $\phi_1 \approx_\alpha \phi_2$, $\phi_2 \approx_\alpha \phi_3$, etc., we get

$$\phi = \exists x\, \phi' =: \underbrace{\exists x_0\, \phi_0'}_{\phi_0} \approx_\alpha \underbrace{\exists x_1\, \phi_1'}_{\phi_1} \approx_\alpha \underbrace{\exists x_2\, \phi_2'}_{\phi_2} \approx_\alpha \cdots \approx_\alpha \underbrace{\exists x_n\, \phi_n'}_{\phi_n} := \exists y\, \psi' = \psi$$

where each $\approx_\alpha$ is either because of $\sim_\alpha$ (in which case the variables are different), or because the inner formulas satisfy $\approx_\alpha$ (in which case the variables are the same).

Let z be any variable which is not any of the x_i's, and does not occur free or bound in any of the ϕ_i''s; thus by Exercise 4.10(b), each $\phi_i'[x_i \mapsto z]$ is safe. For each of the above $\approx_\alpha$'s, say $\exists x_i\, \phi_i' \approx_\alpha \exists x_{i+1}\, \phi_{i+1}'$, we claim that $\phi_i'[x_i \mapsto z] \equiv_\alpha \phi_{i+1}'[x_{i+1} \mapsto z]$:

– If $\exists x_i\, \phi_i' \sim_\alpha \exists x_{i+1}\, \phi_{i+1}'$, with $x_{i+1} \notin \mathrm{FV}(\phi_i') \cup \{x_i\}$ and $\phi_{i+1}' = \phi_i'[x_i \mapsto x_{i+1}]$, then

$$\phi_{i+1}'[x_{i+1} \mapsto z] = \phi_i'[x_i \mapsto x_{i+1}][x_{i+1} \mapsto z] = \phi_i'[x_i \mapsto z]$$

using Exercise 3.4 (where $(x_i \mapsto x_{i+1})[x_{i+1} \mapsto z] = (x_i \mapsto z)$, since $x_{i+1} \notin \mathrm{FV}(\phi_i')$).

– Otherwise, $\exists x_i\, \phi_i' \approx_\alpha \exists x_{i+1}\, \phi_{i+1}'$ holds because $x_i = x_{i+1}$ and $\phi_i' \approx_\alpha \phi_{i+1}'$. Then

$$\phi_i'[x_i \mapsto z] \equiv_\alpha \phi_{i+1}'[x_{i+1} \mapsto z]$$

since these substitutions are safe, and the IH gives us Proposition 4.9 for $\phi_i' \equiv_\alpha \phi_{i+1}'$. We have shown

$$\phi'[x \mapsto z] = \phi_0'[x_0 \mapsto z] \equiv_\alpha \phi_1'[x_1 \mapsto z] \equiv_\alpha \cdots \equiv_\alpha \phi_n'[x_n \mapsto z] = \psi'[y \mapsto z]$$

for all but finitely many z, which proves the weaker version of Proposition 4.14(e).

To complete the induction, we need to prove Proposition 4.9 for $\phi = \exists x\, \phi' \equiv_\alpha \exists y\, \psi' = \psi$. By restricting σ, we may assume $X = \mathrm{FV}(\exists x\, \phi') = \mathrm{FV}(\exists y\, \psi')$. Let z be one of the all-but-finitely-many variables as above, which is also not in either X or any term in the image of σ. Then since $\phi'[x \mapsto z] \equiv_\alpha \psi'[y \mapsto z]$ as shown above, and $\phi'[x \mapsto z]$ has the same height as ϕ' which is strictly less than that of ϕ, we may apply the IH to get

$$\phi'[x \mapsto z][\sigma] \equiv_\alpha \psi'[y \mapsto z][\sigma],$$

whence

$$\begin{aligned}
(\exists x\, \phi')[\sigma] &= \exists x\, \phi'[\sigma\langle x \mapsto x\rangle] \\
&\sim_\alpha \exists z\, \phi'[\sigma\langle x \mapsto x\rangle][x \mapsto z] && \text{since } z \text{ does not appear free or bound in } \phi \\
&= \exists z\, \phi'[\sigma\langle x \mapsto z\rangle] && \text{by Exercise 3.4, since } z \text{ does not appear in } \mathrm{im}(\sigma) \\
&= \exists z\, \phi'[x \mapsto z][\sigma] && \text{by Exercise 3.4, since } z \notin X \\
&\equiv_\alpha \exists z\, \psi'[y \mapsto z][\sigma] \\
&\sim_\alpha (\exists y\, \psi')[\sigma] && \text{similarly.}
\end{aligned}$$

We have now proved Proposition 4.9, as well as Proposition 4.14 with the weaker version of (e). To prove the original (e), where z is any variable such that $\phi'[x \mapsto z]$ and $\psi'[y \mapsto z]$ are both safe: by the weaker version, we may find some other variable $z' \notin \mathrm{FV}(\phi') \cup \mathrm{FV}(\psi')$ such that

$$\phi'[x \mapsto z'] \equiv_\alpha \psi'[y \mapsto z']$$

and these substitutions are both safe; now apply Proposition 4.9 and Exercise 3.4 to get

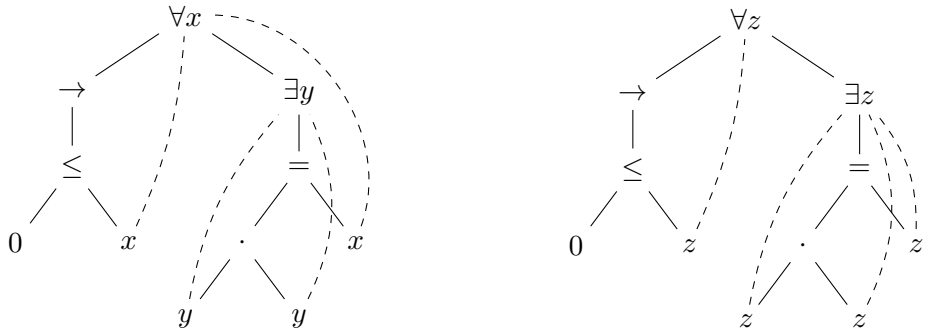
$$\phi'[x \mapsto z] = \phi'[x \mapsto z'][z' \mapsto z] \equiv_\alpha \psi'[y \mapsto z'][z' \mapsto z] = \psi'[y \mapsto z]. \qquad \square$$

6. Cleaner approaches to variable binding

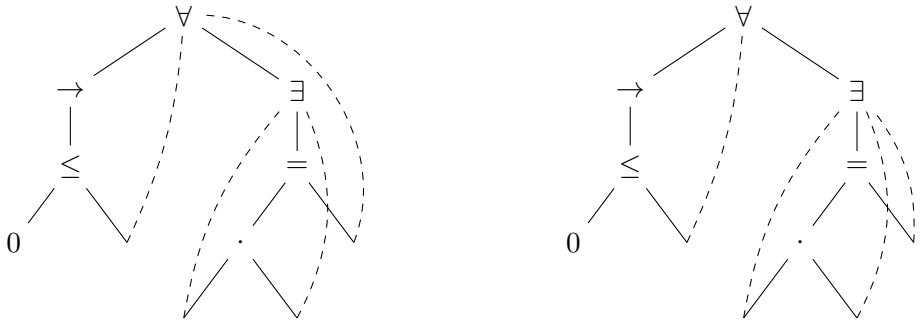
Recall the two formulas from Example 4.15:

$$\forall x\, (0 \le x \to \exists y\, (y \cdot y = x)) \not\equiv_\alpha \forall z\, (0 \le z \to \exists z\, (z \cdot z = z)). \tag{6.1}$$

These have the following parse trees (ignoring our conventions $\forall := \neg\exists\neg$ and $\phi \to \psi := \neg\phi \vee \psi$):



We may think of the only purpose of the bound variables as specifying the label of the quantifier node above which binds them (dashed curves). If we draw these curves, then we may erase the bound variables as well as the quantified variables entirely, leaving behind the graphs with cycles

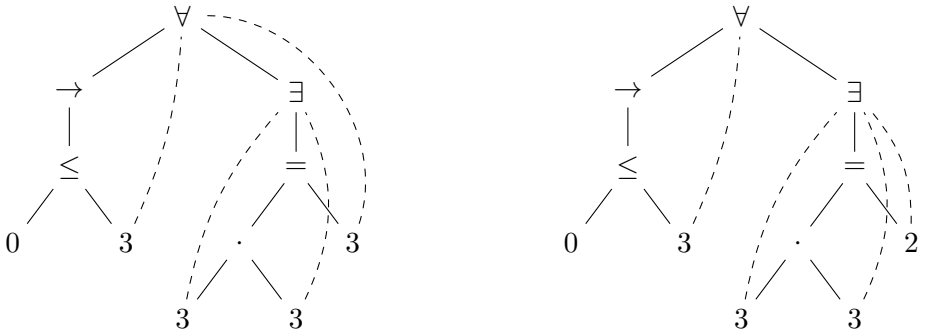


which make it obvious when two formulas are α-equivalent: iff the two graphs are the same.

This begs the question: why not take these graphs as our formal representation of formulas to begin with, so that there's no need to even speak of α-equivalence? The problem is, unlike trees,

graphs aren't a type of inductively constructed mathematical object. Since we definitely want to be able to do induction on formulas, we need to break these cycles somehow, yielding the above trees.

There is in fact a cleverer way to break and label these cycles. Instead of assigning an arbitrary letter like x, y, z as a label, we can *label each leaf with its distance to its binder above it*:[2]



If we "flatten" these trees again into linear expressions, we get

$$\forall\,(0 \leq 3 \to \exists\,(3 \cdot 3 = 3)) \neq \forall\,(0 \leq 3 \to \exists\,(3 \cdot 3 = 2))$$

with $\equiv_\alpha$ again becoming simply syntactic equality. This elegant representation of first-order formulas is known as **de Bruijn indices**, and is commonly used by compilers, proof checkers, and other computer programs that need to manipulate syntax precisely. But unfortunately, these expressions with numerical indices are much harder for a human mathematician to parse than usual formulas as in (6.1)! (Note, for instance, that the 3's on the LHS don't all represent the same variable, while the second and third 3's and last 2 on the RHS *do* represent the same variable. Counting only quantifiers as in Footnote 2 doesn't help much; the first 1 and last 2 on the LHS would represent the same variable.)

[2]The label should also specify clearly that the number represents a bound variable, rather than a constant symbol like 0; for example, we could write v_3 instead of just 3. Another common variation is to only count quantifier nodes, instead of all nodes on the path up to the binder; the tree on the left would have leaf labels $1, 1, 1, 2$ instead.