CHAPTER 3

A theory TCF of computable functionals

After getting clear about the domains we intend to reason about, the partial continuous functionals and in particular the computable ones, we now set up a theory to prove their properties. The main concepts are those of inductively and coinductively defined predicates.

3.1. Formulas and their computational content

Formulas are built up from prime formulas $P\vec{t}$ by implication $\to$ and universal quantification $\forall_x$; here the t_i are terms, x is a variable and P is a predicate of a certain arity (a list of types). We often write $\vec{t} \in P$ for $P\vec{t}$. Predicates can be inductively or coinductively defined. An example for the former is $T_{\mathbb{L}(\mathbb{N})}$, which is defined by the clauses (i) $[] \in T_{\mathbb{L}(\mathbb{N})}$ and (ii) $\forall_{n\in\mathbb{N}}(\ell \in T_{\mathbb{L}(\mathbb{N})} \to n :: \ell \in T_{\mathbb{L}(\mathbb{N})})$. An example for the latter is ${}^{\mathrm{co}}T_{\mathbb{L}(\mathbb{N})}$ defined by a closure axiom saying that every $\ell \in {}^{\mathrm{co}}T_{\mathbb{L}(\mathbb{N})}$ is of the form $n :: \ell'$ with $n \in \mathbb{N}$ and $\ell' \in {}^{\mathrm{co}}T_{\mathbb{L}(\mathbb{N})}$ again. According to Kolmogorov (1932) a formula can be seen as a problem, asking for a solution. In the inductive example a solution for $4 :: 2 :: 0 :: [] \in T_{\mathbb{L}(\mathbb{N})}$ would be the generating (finite) sequence $[]$, $0 :: []$, $2 :: 0 :: []$, $4 :: 2 :: 0 :: []$, and in the coinductive example a solution for a prime formula $t \in {}^{\mathrm{co}}T_{\mathbb{L}(\mathbb{N})}$ would be an (infinite) stream of natural numbers. Generally, a solution for an inductive predicate is a finite construction tree, and for a coinductive predicate a finitely branching possibly infinite destruction tree. Such trees can be seen as ideals of the free algebras considered in Section 2.1.4. A solution for a problem posed by the formula $A \to B$ is a computable functional mapping solutions of A into solutions of B.

Sometimes the solution of a problem does not need all available input. We therefore mark the sources of such computationally superfluous input – that is, some (co)inductive predicates – as "non-computational" (n.c.).

Assume an infinite supply of predicate variables, each of its own *arity* (a list of types). We distinguish two sorts of predicate variables, "computationally relevant" ones $X^c, Y^c, Z^c, W^c \dots$ and "non-computational" ones X^{nc}, Y^{nc}, Z^{nc}, W^{nc} ..., and use $X, Y, Z, W \dots$ for both.

DEFINITION (Clauses and predicate forms). *Clauses* K have the form

$$\forall_{\vec{x}}(\tilde{Y}^c \to \tilde{Z}^{\mathrm{nc}} \to (\forall_{\vec{y}_i}(\tilde{W}_i^{\mathrm{nc}} \to \bar{X}_i))_{i<n} \to \bar{X})$$

with all predicate variables Y_i^c, Z_i^{nc}, W_i^{nc} occurring exactly once and distinct from each other and from X. By $\bar{X}$ we denote the result of applying the predicate variable X to a list of terms of fitting types, and by $\tilde{X}$ lists of those. Iterated implications are understood as associated to the right. A premise of a clause is called a *parameter* premise if X does not occur in it, and a *recursive* premise otherwise. A clause K is *nullary* if it has no recursive premises. We call $I^c := \mu_{X^c}\vec{K}$ and $I^{\mathrm{nc}} := \mu_{X^{\mathrm{nc}}}\vec{K}$ with $\vec{K}$ not empty *predicate forms* (and use I for both), and similarly with ${}^{\mathrm{co}}I$ for I and ν for μ.

EXAMPLES. Recall that $[]^{\mathbb{L}(\alpha)}$ and $\mathrm{Cons}^{\alpha\to\mathbb{L}(\alpha)\to\mathbb{L}(\alpha)}$ are the two constructors of the algebra form $\mathbb{L}(\alpha)$ of lists, written $[]$ and $::$ (infix).

1. Let Y of arity (α,α) and X of arity $(\mathbb{L}(\alpha),\mathbb{L}(\alpha))$ be predicate variables. Then

$$K_0 := X([],[]),$$
$$K_1 := \forall_{x,x'}(Y(x,x') \to \forall_{\ell,\ell'}(X(\ell,\ell') \to X(x :: \ell, x' :: \ell')))$$

are clauses and both *similarity* $\sim_{\mathbb{L}}(Y)$ (or $\sim_{\mathbb{L},Y}$) defined by $\mu_X(K_0,K_1)$ and *bisimilarity* $\approx_{\mathbb{L}}(Y)$ (or $\approx_{\mathbb{L},Y}$) defined by $\nu_X(K_0,K_1)$ are predicate forms with Y a parameter predicate variable. Note that we can omit the type parameter α, since it can be read off from the arity of Y.

2. Alternatively let Y of arity (α) and X of arity $(\mathbb{L}(\alpha))$ be predicate variables. Then

$$K_0 := ([] \in X),$$
$$K_1 := \forall_x(x \in Y \to \forall_\ell(\ell \in X \to x :: \ell \in X))$$

are clauses and both *relative totality* $T_{\mathbb{L}}(Y)$ (or $T_{\mathbb{L},Y}$) defined by $\mu_X(K_0,K_1)$ and also *relative cototality* ${}^{\mathrm{co}}T_{\mathbb{L}}(Y)$ (or ${}^{\mathrm{co}}T_{\mathbb{L},Y}$) defined by $\nu_X(K_0,K_1)$ are predicate forms with Y a parameter predicate variable.

DEFINITION (Algebra form of a predicate form). From every clause K we obtain a constructor type by (i) omitting quantifiers, (ii) dropping all n.c. predicates and from the c.r. predicates their arguments, and (iii) replacing the remaining predicate variables by type variables. That is, from the clause

$$\forall_{\vec{x}}(\tilde{Y}^c \to \tilde{Z}^{\mathrm{nc}} \to (\forall_{\vec{y}_i}(\tilde{W}_i^{\mathrm{nc}} \to \bar{X}_i))_{i<n} \to \bar{X})$$

we obtain the constructor type $\vec{\alpha} \to (\xi)_{i<n} \to \xi$. With every predicate form $I^c := (\mu/\nu)_{X^c}\vec{K}$ we canonically associate the algebra form $\iota_{I^c} := \mu_\xi\vec{\kappa}$.

EXAMPLES. For each of the predicate forms $\sim_{\mathbb{L}}$, $\approx_{\mathbb{L}}$, $T_{\mathbb{L}}$, ${}^{\mathrm{co}}T_{\mathbb{L}}$ defined above the associated algebra form is $\mathbb{L}$. Notice that the distinction whether a predicate form I is defined as a least or greatest fixed point is ignored.

DEFINITION (Predicates and formulas).

$$\begin{aligned} P, Q &::= X \mid \{\, \vec{x} \mid A \,\} \mid I(\vec{\rho}, \vec{P}\,) \mid {}^{\mathrm{co}}I(\vec{\rho}, \vec{P}\,) && \text{(predicates)},\\ A, B &::= P\vec{t} \mid A \to B \mid \forall_x A && \text{(formulas)} \end{aligned}$$

with $I/{}^{\mathrm{co}}I$ a predicate form. $(I/{}^{\mathrm{co}}I)(\vec{\rho}, \vec{P}\,)$ is the result of substituting the types $\vec{\rho}$ and the (already generated) predicates $\vec{P}$ for its type and predicate variables. To take care of the difference between X^c and X^{nc} we define the final predicate of a predicate or formula by

$$\begin{aligned} \mathrm{fp}(X) &:= X, & \mathrm{fp}(P\vec{t}\,) &:= \mathrm{fp}(P),\\ \mathrm{fp}(\{\, \vec{x} \mid A \,\}) &:= \mathrm{fp}(A), & \mathrm{fp}(A \to B) &:= \mathrm{fp}(B),\\ \mathrm{fp}((I/{}^{\mathrm{co}}I)(\vec{\rho}, \vec{P}\,)) &:= I/{}^{\mathrm{co}}I, & \mathrm{fp}(\forall_x A) &:= \mathrm{fp}(A). \end{aligned}$$

We call a predicate or formula C *non-computational* (n.c., or *Harrop*) if its final predicate $\mathrm{fp}(C)$ is of the form X^{nc} or I^{nc}, else *computationally relevant* (c.r.). Now we require that all predicate substitutions involved in $(I/{}^{\mathrm{co}}I)(\vec{\rho}, \vec{P}\,)$ substitute c.r. predicates for c.r. predicate variables and n.c. predicates for n.c. predicate variables. Such predicate substitutions are called *sharp*.

Predicates of the form $I(\vec{\rho}, \vec{P}\,)$ are called *inductive*, and predicates of the form ${}^{\mathrm{co}}I(\vec{\rho}, \vec{P}\,)$ *coinductive*.

The terms $\vec{t}$ are those introduced in Section 2.2.1, i.e., typed terms built from typed variables and constants by abstraction and application, and (importantly) those with a common reduct are identified.

A predicate of the form $\{\, \vec{x} \mid C \,\}$ is called a *comprehension term*. We identify $\{\, \vec{x} \mid C(\vec{x}\,) \,\}\vec{t}$ with $C(\vec{t}\,)$. For a predicate C of arity $(\rho, \vec{\sigma}\,)$ we write Ct for $\{\, \vec{y} \mid Ct\vec{y} \,\}$.

It is a natural question to ask what the type of a "realizer" or "witness" of a c.r. predicate or formula C should be.

DEFINITION (Type $\tau(C)$ of a c.r. predicate or formula C). Assume a global injective assignment of type variables ξ to c.r. predicate variables X^c.

$$\begin{aligned} \tau(X^c) &:= \xi, & \tau(P\vec{t}\,) &:= \tau(P),\\ \tau(\{\, \vec{x} \mid A \,\}) &:= \tau(A), & \tau(A \to B) &:= \begin{cases} \tau(A) \to \tau(B) & (A \text{ c.r.})\\ \tau(B) & (A \text{ n.c.}) \end{cases}\\ \tau((I^c/{}^{\mathrm{co}}I^c)(\vec{\rho}, \vec{P}\,)) &:= \iota_{I^c}(\tau(\vec{P}^c)), & \tau(\forall_x A) &:= \tau(A) \end{aligned}$$

where $\vec{P}^c$ are the c.r. predicates among $\vec{P}$ and ι_{I^c} is the algebra form associated with the predicate form $I^c/{}^{\mathrm{co}}I^c$. We call $\iota_{I^c}(\tau(\vec{P}^c))$ the *algebra associated with* $(I^c/{}^{\mathrm{co}}I^c)(\vec{\rho},\vec{P})$.

Examples. 1. The *even numbers* are inductively defined by

$$\mathrm{Even} := \mu_{X^c}(0 \in X^c, \forall_n(n \in X^c \to S(Sn) \in X^c)).$$

The constructor types of $\tau(\mathrm{Even})$ are ξ and $\xi \to \xi$, hence $\tau(\mathrm{Even}) = \mathbb{N}$.

2. For every closed base type $\iota(\vec{\rho})$ we define c.r. *equality* predicates $\sim_\iota(\vec{P}\,)$ and $\approx_\iota(\vec{P}\,)$ of arity $(\iota(\vec{\rho}), \iota(\vec{\rho}))$ by induction on the height $|\iota(\vec{\rho})| := 1 + \max|\vec{\rho}|$. Here ι is the name of an algebra form (for instance $\mathbb{L}$), the ρ_i are closed base types and the P_i are predicates of arity (ρ_i, ρ_i).

(i). $\sim_{\mathbb{N}} := \mu_{X^c}(K_0, K_1)$ and $\approx_{\mathbb{N}} := \nu_{X^c}(K_0, K_1)$ with clauses

$$K_0 := X^c(0,0),$$
$$K_1 := \forall_{n,n'}(X^c(n,n') \to X^c(Sn, Sn')).$$

Then $\tau(\sim_{\mathbb{N}}) = \tau(\approx_{\mathbb{N}}) = \mathbb{N}$.

(ii). $\sim_{\mathbb{L}}(\approx_{\mathbb{N}}) := \mu_{X^c}(K_0, K_1)$ and $\approx_{\mathbb{L}}(\approx_{\mathbb{N}}) := \nu_{X^c}(K_0, K_1)$ with clauses

$$K_0 := X^c([\,],[\,]),$$
$$K_1 := \forall_{n,n'}(n \approx_{\mathbb{N}} n' \to \forall_{\ell,\ell'}(X^c(\ell,\ell') \to X^c(n :: \ell, n' :: \ell'))).$$

Then $\tau(\sim_{\mathbb{L}}(\approx_{\mathbb{N}})) = \tau(\approx_{\mathbb{L}}(\approx_{\mathbb{N}})) = \mathbb{L}(\mathbb{N})$.

(iii). $\sim_{\mathbb{L}}(\sim_{\mathbb{L}}(\approx_{\mathbb{N}})) := \mu_{X^c}(K_0, K_1)$ and $\approx_{\mathbb{L}}(\sim_{\mathbb{L}}(\approx_{\mathbb{N}})) := \nu_{X^c}(K_0, K_1)$ with clauses

$$K_0 := X^c([\,],[\,]),$$
$$K_1 := \forall_{\ell,\ell'}(\ell \sim_{\mathbb{L}}(\approx_{\mathbb{N}}) \ell' \to \forall_{u,u'}(X^c(u,u') \to X^c(\ell :: u, \ell' :: u'))).$$

Then $\tau(\sim_{\mathbb{L}}(\sim_{\mathbb{L}}(\approx_{\mathbb{N}}))) = \tau(\approx_{\mathbb{L}}(\sim_{\mathbb{L}}(\approx_{\mathbb{N}}))) = \mathbb{L}(\mathbb{L}(\mathbb{N}))$.

3. The *transitive closure* of a binary relation is defined as follows. Let

$$K_0 := \forall_{x,y}(Yxy \to Xxy),$$
$$K_1 := \forall_{x,y,z}(Yxy \to Xyz \to Xxz)$$

where x, y, z are variables of type α and Y, X predicate variables of arity (α, α). From these clauses we define the predicate form

$$\mathrm{TC} := \mu_X(K_0, K_1)$$

with parameter type variable α and parameter predicate variable Y.

Let $<$ be a c.r. "less than" relation on the natural numbers, of arity $(\mathbb{N}, \mathbb{N})$. Then $\mathrm{TC}(\mathbb{N}, <)$ is its transitive closure. The constructor types of $\tau(\mathrm{TC})$ are $\beta \to \xi$ and $\beta \to \xi \to \xi$, hence $\tau(\mathrm{TC}(\mathbb{N}, <))$ is the type of non-empty lists of natural numbers.

4. An important example of an inductive predicate is *Leibniz equality*, defined simply by

$$\mathrm{EqD} := \mu_{X^{\mathrm{nc}}}(\forall_x X^{\mathrm{nc}} xx) \qquad (\text{D for "inductively } \mathbf{d}\text{efined"}).$$

We will use the abbreviation

$$(x \equiv y) := \mathrm{EqD}(x, y).$$

The missing logical connectives disjunction, conjunction and existence can now be defined inductively. Disjunction is a special case of union

$$\mathrm{Cup}_{Y,Z} := \mu_{X^c}(\forall_{\vec{x}}(Y\vec{x} \to X^c\vec{x}),\ \forall_{\vec{x}}(Z\vec{x} \to X^c\vec{x})).$$

Since the parameter predicates Y, Z can be chosen as either c.r. or n.c. we obtain the variants

$$\begin{aligned}
\mathrm{CupD}_{Y^c,Z^c} &:= \mu_{X^c}(\forall_{\vec{x}}(Y^c\vec{x} \to X^c\vec{x}),\ \forall_{\vec{x}}(Z^c\vec{x} \to X^c\vec{x})),\\
\mathrm{CupL}_{Y^c,Z^{\mathrm{nc}}} &:= \mu_{X^c}(\forall_{\vec{x}}(Y^c\vec{x} \to X^c\vec{x}),\ \forall_{\vec{x}}(Z^{\mathrm{nc}}\vec{x} \to X^c\vec{x})),\\
\mathrm{CupR}_{Y^{\mathrm{nc}},Z^c} &:= \mu_{X^c}(\forall_{\vec{x}}(Y^{\mathrm{nc}}\vec{x} \to X^c\vec{x}),\ \forall_{\vec{x}}(Z^c\vec{x} \to X^c\vec{x})),\\
\mathrm{CupU}_{Y^{\mathrm{nc}},Z^{\mathrm{nc}}} &:= \mu_{X^c}(\forall_{\vec{x}}(Y^{\mathrm{nc}}\vec{x} \to X^c\vec{x}),\ \forall_{\vec{x}}(Z^{\mathrm{nc}}\vec{x} \to X^c\vec{x})),\\
\mathrm{CupNc}_{Y,Z} &:= \mu_{X^{\mathrm{nc}}}(\forall_{\vec{x}}(Y\vec{x} \to X^{\mathrm{nc}}\vec{x}),\ \forall_{\vec{x}}(Z\vec{x} \to X^{\mathrm{nc}}\vec{x})).
\end{aligned}$$

Here D is for "double", L for "left", R for "right" and U for "uniform". Then by definition

$$\begin{aligned}
\tau(\mathrm{CupD}) &= \mu_\xi(\beta_0 \to \xi, \beta_1 \to \xi) &&= \beta_0 + \beta_1,\\
\tau(\mathrm{CupL}) &= \mu_\xi(\beta \to \xi, \xi) &&= \beta + \mathbb{U},\\
\tau(\mathrm{CupR}) &= \mu_\xi(\xi, \beta \to \xi) &&= \mathbb{U} + \beta,\\
\tau(\mathrm{CupU}) &= \mu_\xi(\xi, \xi) &&= \mathbb{B}.
\end{aligned}$$

We use the abbreviations

$$\begin{aligned}
P \cup^d Q &:= \mathrm{CupD}_{P,Q},\\
P \cup^l Q &:= \mathrm{CupL}_{P,Q},\\
P \cup^r Q &:= \mathrm{CupR}_{P,Q},\\
P \cup^u Q &:= \mathrm{CupU}_{P,Q},\\
P \cup^{\mathrm{nc}} Q &:= \mathrm{CupNc}_{P,Q}.
\end{aligned}$$

In case of nullary predicates we use

$$\begin{aligned} A \vee^{\mathrm{d}} B &:= \mathrm{CupD}_{\{|A\},\{|B\}}, \\ A \vee^{\mathrm{l}} B &:= \mathrm{CupL}_{\{|A\},\{|B\}}, \\ A \vee^{\mathrm{r}} B &:= \mathrm{CupR}_{\{|A\},\{|B\}}, \\ A \vee^{\mathrm{u}} B &:= \mathrm{CupU}_{\{|A\},\{|B\}}, \\ A \vee^{\mathrm{nc}} B &:= \mathrm{CupNc}_{\{|A\},\{|B\}}. \end{aligned}$$

Since the "decoration" is determined by the c.r./n.c. status of the two parameter predicates we usually leave it out in $\vee^{\mathrm{d}}, \vee^{\mathrm{l}}, \vee^{\mathrm{r}}, \vee^{\mathrm{u}}$ and just write $\vee$. However in the final nc-variant we suppress even the information which clause has been used, and hence must keep the notation $\vee^{\mathrm{nc}}$.

Similarly conjunction is a special case of intersection

$$\mathrm{Cap}_{Y,Z} := \mu_{X^{\mathrm{c}}}(\forall_{\vec{x}}(Y\vec{x} \to Z\vec{x} \to X^{\mathrm{c}}\vec{x})),$$

and we obtain the variants

$$\begin{aligned} \mathrm{CapD}_{Y^{\mathrm{c}},Z^{\mathrm{c}}} &:= \mu_{X^{\mathrm{c}}}(\forall_{\vec{x}}(Y^{\mathrm{c}}\vec{x} \to Z^{\mathrm{c}}\vec{x} \to X^{\mathrm{c}}\vec{x})), \\ \mathrm{CapL}_{Y^{\mathrm{c}},Z^{\mathrm{nc}}} &:= \mu_{X^{\mathrm{c}}}(\forall_{\vec{x}}(Y^{\mathrm{c}}\vec{x} \to Z^{\mathrm{nc}}\vec{x} \to X^{\mathrm{c}}\vec{x})), \\ \mathrm{CapR}_{Y^{\mathrm{nc}},Z^{\mathrm{c}}} &:= \mu_{X^{\mathrm{c}}}(\forall_{\vec{x}}(Y^{\mathrm{nc}}\vec{x} \to Z^{\mathrm{c}}\vec{x} \to X^{\mathrm{c}}\vec{x})), \\ \mathrm{CapNc}_{Y,Z} &:= \mu_{X^{\mathrm{nc}}}(\forall_{\vec{x}}(Y\vec{x} \to Z\vec{x} \to X^{\mathrm{nc}}\vec{x})). \end{aligned}$$

Then by definition

$$\begin{aligned} \tau(\mathrm{CapD}) = \mu_\xi(\beta_0 \to \beta_1 \to \xi) &= \beta_0 \times \beta_1 \\ \tau(\mathrm{CapL}) = \tau(\mathrm{CapR}) = \mu_\xi(\beta \to \xi) &= \mathbb{I}(\beta). \end{aligned}$$

We use the abbreviations

$$\begin{aligned} P \cap^{d} Q &:= \mathrm{CapD}_{P,Q}, \\ P \cap^{l} Q &:= \mathrm{CapL}_{P,Q}, \\ P \cap^{r} Q &:= \mathrm{CapR}_{P,Q}, \\ P \cap^{\mathrm{nc}} Q &:= \mathrm{CapNc}_{P,Q}. \end{aligned}$$

In case of nullary predicates we use

$$\begin{aligned} A \wedge^{\mathrm{d}} B &:= \mathrm{CapD}_{\{|A\},\{|B\}}, \\ A \wedge^{\mathrm{l}} B &:= \mathrm{CapL}_{\{|A\},\{|B\}}, \\ A \wedge^{\mathrm{r}} B &:= \mathrm{CapR}_{\{|A\},\{|B\}}, \\ A \wedge^{\mathrm{nc}} B &:= \mathrm{CapNc}_{\{|A\},\{|B\}}. \end{aligned}$$

Again since the decoration is determined by the c.r./n.c. status of the two parameter predicates we usually leave out the decoration and just write $\wedge$.

Finally existence is defined inductively by

$$\begin{aligned}\mathrm{Ex}_{Y^c} &:= \mu_{X^c}(\forall_x(x \in Y^c \to X^c)),\\ \mathrm{ExNc}_Y &:= \mu_{X^{\mathrm{nc}}}(\forall_x(x \in Y \to X^{\mathrm{nc}})).\end{aligned}$$

Then by definition

$$\tau(\mathrm{Ex}) = \mu_\xi(\beta \to \xi) = \mathbb{I}(\beta).$$

We use the abbreviation

$$\begin{aligned}\exists_x A &:= \mathrm{Ex}_{\{x|A\}},\\ \exists^{\mathrm{nc}}_x A &:= \mathrm{ExNc}_{\{x|A\}},\end{aligned}$$

and again since the decoration is determined by the c.r./n.c. status of the parameter predicate we usually leave out the decoration and just write $\exists$.

3.2. Axioms of TCF

We define a theory of computable functionals, called TCF. Formulas are the ones defined above, involving typed variables. Derivations use the rules of minimal logic for $\to$ and $\forall$, and the axioms introduced below. However, because of the presence of decorations we have an extra degree of freedom. By an *n.c. part* of a derivation we mean a subderivation with an n.c. end formula. Such n.c. parts will not contribute to the computational content of the whole derivation, and hence we can ignore all decorations in those parts (i.e., use a modified notion of equality of formulas there).

For each inductive predicate there are "clauses" or introduction axioms, together with a "least-fixed-point" or elimination axiom. To grasp the general form of these axioms it is convenient to write a clause

$$\forall_{\vec{x}}(\tilde{Y}^c \to \tilde{Z}^{\mathrm{nc}} \to (\forall_{\vec{y}_i}(\tilde{W}_i^{\mathrm{nc}} \to \bar{X}_i))_{i<n} \to \bar{X}) \quad \text{as} \quad \forall_{\vec{x}}((A_\nu(X))_{\nu<n} \to X\vec{t}\,).$$

Definition (Introduction and elimination axioms for inductive predicates). For an inductive predicate $\mu_X(\forall_{\vec{x}_i}((A_{i\nu}(X))_{\nu<n_i} \to X\vec{t}_i))_{i<k} =: I$ we have k introduction axioms I_i^+ $(i < k)$ and one elimination axiom I^-:

$$I_i^+ \colon \forall_{\vec{x}_i}((A_{i\nu}(I))_{\nu<n_i} \to I\vec{t}_i), \tag{9}$$

$$I^- \colon (\forall_{\vec{x}_i}((A_{i\nu}(I \cap X))_{\nu<n_i} \to X\vec{t}_i))_{i<k} \to I \subseteq X \tag{10}$$

($I \cap X$ was inductively defined above). (10) expresses that every competitor X satisfying the same clauses contains I. We take all substitution instances of I_i^+, I^- (w.r.t. substitutions for type and predicate variables) as axioms.

Remarks. (i) We use a "strengthened" form of the "step formula", namely $\forall_{\vec{x}_i}(A_{i\nu}(I \cap X))_{\nu<n_i} \to X\vec{t}_i$ rather than $\forall_{\vec{x}_i}(A_{i\nu}(X))_{\nu<n_i} \to X\vec{t}_i$. In applications of the least-fixed-point axiom this simplifies the proof of the "step", since we have an additional I-hypothesis available.

(ii) Notice that there is no circularity here for the inductive predicate $Y \cap Z := \mathrm{Cap}_{Y,Z}$, since there are no recursive calls in this particular inductive definition and hence $\cap$ does not occur in

$$\mathrm{Cap}^-_{Y,Z} \colon \forall_{\vec{x}}(Y\vec{x} \to Z\vec{x} \to X\vec{x}) \to \mathrm{Cap}_{Y,Z} \subseteq X.$$

(iii) The elimination axiom (10) could equivalently be written as

$$I^- \colon \forall_{\vec{x}}(I\vec{x} \to (\forall_{\vec{x}_i}((A_{i\nu}(I \cap X))_{\nu<n_i} \to X\vec{t_i}))_{i<k} \to X\vec{x}\,)$$

In this form it fits better with our (i.e., Gentzen's) way to write the logical elimination rules, where the main premise comes first. More importantly, its type (cf. Section 3.1) will then be the type of the recursion operator $\mathcal{R}^\tau_\iota$ taken as the "computational content" (cf. Section 4.1) of the elimination axiom for I. Therefore in the implementation of TCF this form is used. However, for readability we often prefer the form (10) in the present notes.

Examples. 1. For Even the introduction axioms are

$$0 \in \mathrm{Even}, \qquad \forall_n(n \in \mathrm{Even} \to S(Sn) \in \mathrm{Even})$$

and the elimination axiom is Even^-:

$$0 \in X \to \forall_n(n \in \mathrm{Even} \to n \in X \to S(Sn) \in X) \to \mathrm{Even} \subseteq X.$$

As an example of how to use these axioms we prove an "inversion" property:

$$n \in \mathrm{Even} \leftrightarrow n \equiv 0 \vee \exists_{n'}(n' \in \mathrm{Even} \wedge n \equiv S(Sn')).$$

"$\leftarrow$". Use the introduction axioms. "$\rightarrow$". Use Even^- with competitor predicate $X := \{\, n \mid n \equiv 0 \vee \exists_{n'}(n' \in \mathrm{Even} \wedge n \equiv S(Sn'))\,\}$. It suffices to prove the premises of Even^- for this X. For the first premise this is clear. For the second assume n with $n \in \mathrm{Even} \cap X$. The goal is $S(Sn) \in X$. It suffices to show $\exists_{n'}(n' \in \mathrm{Even} \wedge S(Sn) \equiv S(Sn'))$. Take n.

2. Consider the algebra $\mathbb{Y}$ of binary trees. We want to represent the set of total ideals by an inductively defined predicate $T_{\mathbb{Y}}$. The clauses are

$$(T_{\mathbb{Y}})^+_0 \colon - \in T_{\mathbb{Y}},$$
$$(T_{\mathbb{Y}})^+_1 \colon \forall_{t_1,t_2}(t_1, t_2 \in T_{\mathbb{Y}} \to \mathrm{C}t_1t_2 \in T_{\mathbb{Y}})$$

and the elimination axiom is

$$T^-_{\mathbb{Y}} \colon - \in X \to \forall_{t_1,t_2}(t_1, t_2 \in T_{\mathbb{Y}} \cap X \to \mathrm{C}t_1t_2 \in X) \to T_{\mathbb{Y}} \subseteq X.$$

From these axioms we prove the inversion property:

$$t \in T_{\mathbb{Y}} \leftrightarrow (t \equiv -) \vee \exists_{t_1,t_2}(t_1, t_2 \in T_{\mathbb{Y}} \wedge t \equiv \mathrm{C}t_1t_2).$$

"$\leftarrow$". Use the introduction axioms. "$\rightarrow$". Use $T^-_{\mathbb{Y}}$ with competitor predicate $X := \{\, t \mid (t \equiv -) \vee \exists_{t_1,t_2}(t_1, t_2 \in T_{\mathbb{Y}} \wedge t \equiv \mathrm{C}t_1t_2\,\}$. It suffices to prove the premises of $T^-_{\mathbb{Y}}$ for this X. For the first premise this is clear. For the second

assume t_1, t_2 with $t_1, t_2 \in T_\Upsilon \cap X$. The goal is $\mathrm{C}t_1t_2 \in X$. It suffices to show $\exists_{t_1',t_2'}(t_1', t_2' \in T_\Upsilon \wedge \mathrm{C}t_1't_2' \equiv \mathrm{C}t_1t_2)$. Take t_1, t_2.

3. For the transitive closure $\mathrm{TC}_\prec := \mathrm{TC}(\rho, \prec)$ of a binary relation $\prec$ on objects of type ρ the introduction axioms are

$$\forall_{x,y}(x \prec y \to \mathrm{TC}_\prec(x,y)),$$
$$\forall_{x,y,z}(x \prec z \to \mathrm{TC}_\prec(z,y) \to \mathrm{TC}_\prec(x,y))$$

and the elimination axiom is $\mathrm{TC}^-_\prec$:

$$\forall_{x,y}(x \prec y \to Xxy) \to \forall_{x,y,z}(x \prec z \to \mathrm{TC}_\prec(z,y) \to Xzy \to Xxy) \to \mathrm{TC}_\prec \subseteq X.$$

The inversion property

$$\mathrm{TC}_\prec(x,y) \leftrightarrow x \prec y \vee \exists_z(x \prec z \wedge \mathrm{TC}_\prec(z,y))$$

can be proved as above. For "$\leftarrow$" use the introduction axioms, and for "$\rightarrow$" use $\mathrm{TC}^-_\prec$ with competitor $X := \{\, x, y \mid x \prec y \vee \exists_z(x \prec z \wedge \mathrm{TC}_\prec(z,y)) \,\}$. It suffices to prove the premises of $\mathrm{TC}^-_\prec$ for this X. For the first premise this is clear. For the second assume x, y, z with $x \prec z$ and $\mathrm{TC}_\prec(z,y)$ and Xzy. The goal is Xxy. It suffices to show $\exists_z(x \prec z \wedge \mathrm{TC}_\prec(z,y))$. Take z.

4. For $\sim_{X\times Y} := \sim_\times(\rho, \sigma, X, Y)$ the introduction axiom is

$$\forall_{x,x',y,y'}(Xxx' \to Yyy' \to \langle x,y\rangle \sim_{X\times Y} \langle x',y'\rangle)$$

and the elimination axiom is $\sim^-_{X\times Y}$:

$$\forall_{x,x',y,y'}(Xxx' \to Yyy' \to Z(\langle x,y\rangle, \langle x',y'\rangle)) \to \sim_{X\times Y} \subseteq Z.$$

Hence $\sim_{X\times Y} = \{\, (\langle x,y\rangle, \langle x',y'\rangle) \mid Xxx', Yyy' \,\}$. The inversion property now is

$$p \sim_{X\times Y} p' \leftrightarrow \exists_{x,x',y,y'}(Xxx' \wedge Yyy' \wedge p \equiv \langle x,y\rangle \wedge p' \equiv \langle x',y'\rangle).$$

For "$\leftarrow$" use the introduction axioms, and for "$\rightarrow$" use $\sim^-_{X\times Y}$ with competitor $Z := \{\, p, p' \mid \exists_{x,x',y,y'}(Xxx' \wedge Yyy' \wedge p \equiv \langle x,y\rangle \wedge p' \equiv \langle x',y'\rangle \,\}$. It suffices to prove the premise of $\sim^-_{X\times Y}$ for this Z. Assume x, x', y, y' with Xxx' and Yyy' and $p \equiv \langle x,y\rangle$ and $p' \equiv \langle x',y'\rangle$. The goal is Zpp'. It suffices to show $\exists_{x,x',y,y'}(Xxx' \wedge Yyy' \wedge p \equiv \langle x,y\rangle \wedge p' \equiv \langle x',y'\rangle)$. Take x, x', y, y'.

Similarly for $\sim_{X+Y} := \sim_+(\rho, \sigma, X, Y)$ the introduction axioms are

$$\forall_{x,x'}(Xxx' \to \mathrm{InL}(x) \sim_{X+Y} \mathrm{InL}(x')),$$
$$\forall_{y,y'}(Yyy' \to \mathrm{InR}(y) \sim_{X+Y} \mathrm{InR}(y'))$$

and the elimination axiom is

$$\forall_{x,x'}(Xxx' \to Z(\mathrm{InL}(x), \mathrm{InL}(x')) \to$$
$$\forall_{y,y'}(Yyy' \to Z(\mathrm{InR}(y), \mathrm{InR}(y')) \to \sim_{X+Y} \subseteq Z.$$

Hence $\sim_{X+Y} = \{\,(\mathrm{InL}(x), \mathrm{InL}(x')) \mid Xxx'\,\} \cup \{\,(\mathrm{InR}(y), \mathrm{InR}(y')) \mid Yyy'\,\}$.

Recall the inductive definitions of existence, intersection and union given above. For nullary predicates $P = \{\mid A\,\}$ and $Q = \{\mid B\,\}$ we write $A \wedge B$ for $P \cap Q$ and $A \vee B$ for $P \cup Q$. Then – as in Chapter 1 – the introduction axioms are

$$\begin{aligned}
&\forall_x(A \to \exists_x A),\\
&A \to B \to A \wedge B,\\
&A \to A \vee B, \qquad B \to A \vee B
\end{aligned}$$

and the elimination axioms are (now written in the equivalent form mentioned above, where the main premise comes first)

$$\begin{aligned}
&\exists_x A \to \forall_x(A \to B) \to B \qquad (x \notin \mathrm{FV}(B)),\\
&A \wedge B \to (A \to B \to C) \to C,\\
&A \vee B \to (A \to C) \to (B \to C) \to C.
\end{aligned}$$

To understand the axioms for coinductive predicates note that the conjunction of the k clauses (9) of an inductive predicate I is equivalent to

$$\forall_{\vec{x}}(\bigvee_{i<k} \exists_{\vec{x}_i}(\bigwedge_{\nu<n_i} A_{i\nu}(I) \wedge \vec{x} \equiv \vec{t}_i) \to I\vec{x}\,).$$

Definition (Closure and greatest-fixed-point axioms). For an inductive predicate $\mu_X(\forall_{\vec{x}_i}((A_{i\nu}(X))_{\nu<n_i} \to X\vec{t}_i))_{i<k} =: I$ we define its dual ${}^{\mathrm{co}}I$ by the *closure axiom* ${}^{\mathrm{co}}I^-$ and the *greatest-fixed-point axiom* ${}^{\mathrm{co}}I^+$:

$$(11) \qquad {}^{\mathrm{co}}I^-\colon \forall_{\vec{x}}({}^{\mathrm{co}}I\vec{x} \to \bigvee_{i<k} \exists_{\vec{x}_i}(\bigwedge_{\nu<n_i} A_{i\nu}({}^{\mathrm{co}}I) \wedge \vec{x} \equiv \vec{t}_i))$$

$$(12) \qquad {}^{\mathrm{co}}I^+\colon \forall_{\vec{x}}(X\vec{x} \to \bigvee_{i<k} \exists_{\vec{x}_i}(\bigwedge_{\nu<n_i} A_{i\nu}({}^{\mathrm{co}}I \cup X) \wedge \vec{x} \equiv \vec{t}_i)) \to X \subseteq {}^{\mathrm{co}}I.$$

(${}^{\mathrm{co}}I \cup X$ was inductively defined above). The axiom expresses that every "competitor" X satisfying the closure axiom is contained in ${}^{\mathrm{co}}I$. We take all substitution instances of ${}^{\mathrm{co}}I^+$, ${}^{\mathrm{co}}I^-$ (w.r.t. substitutions for type and predicate variables) as axioms.

Again we have used a "strengthened" form of the "step formula", with $A_{i\nu}({}^{\mathrm{co}}I \cup X)$ rather than $A_{i\nu}(X)$. In applications of the greatest-fixed-point axiom this simplifies the proof of the "step", since its conclusion is weaker.

Remark. The greatest-fixed-point axiom (12) could be written as

$$\forall_{\vec{x}}(X\vec{x} \to \forall_{\vec{x}}(X\vec{x} \to \bigvee_{i<k} \exists_{\vec{x}_i}(\bigwedge_{\nu<n_i} A_{i\nu}({}^{\mathrm{co}}I \cup X) \wedge \vec{x} \equiv \vec{t}_i)) \to {}^{\mathrm{co}}I\vec{x}).$$

Then its type will be the type of the corecursion operator ${}^{\mathrm{co}}\mathcal{R}^\tau_\iota$ taken as the "computational content" (cf. Section 4.1) of the greatest-fixed-point axiom

for ${}^{\mathrm{co}}I$. Therefore in the implementation of TCF this form is used. However, for readability we prefer the form (12) in the present notes.

REMARK. Instead of Leibniz equality $\equiv$ in (11) and (12) we could also use a different equality relation, for instance the n.c. variant $\doteq^{\mathrm{nc}}$ of pointwise equality to be introduced in Section 3.3. This leads to a new variant of ${}^{\mathrm{co}}I$.

EXAMPLES. 1. To show how to construct the dual ${}^{\mathrm{co}}I$ of an inductive predicate I we consider the predicate Even. The conjunction of its two clauses is equivalent to

$$\forall_n(n \equiv 0 \vee \exists_{n'}(n' \in \mathrm{Even} \wedge n \equiv S(Sn')) \to n \in \mathrm{Even}).$$

Now the dual ${}^{\mathrm{co}}\mathrm{Even}$ of Even is defined by its closure axiom ${}^{\mathrm{co}}\mathrm{Even}^-$:

$$\forall_n(n \in {}^{\mathrm{co}}\mathrm{Even} \to n \equiv 0 \vee \exists_{n'}(n' \in {}^{\mathrm{co}}\mathrm{Even} \wedge n \equiv S(Sn')))$$

and its greatest-fixed-point axiom ${}^{\mathrm{co}}\mathrm{Even}^+$:

$$\forall_n(Xn \to n \equiv 0 \vee \exists_{n'}(n' \in ({}^{\mathrm{co}}\mathrm{Even} \cup X) \wedge n \equiv S(Sn'))) \to X \subseteq {}^{\mathrm{co}}\mathrm{Even}.$$

Also for ${}^{\mathrm{co}}\mathrm{Even}$ from these axioms we can prove an inversion property:

$$n \in {}^{\mathrm{co}}\mathrm{Even} \leftrightarrow n \equiv 0 \vee \exists_{n'}(n' \in {}^{\mathrm{co}}\mathrm{Even} \wedge n \equiv S(Sn')).$$

"$\to$". Use the closure axiom. "$\leftarrow$". Use ${}^{\mathrm{co}}\mathrm{Even}^+$, with competitor $X := \{\, n \mid n \equiv 0 \vee \exists_{n'}(n' \in {}^{\mathrm{co}}\mathrm{Even} \wedge n \equiv S(Sn')) \,\}$. It suffices to prove the premise of ${}^{\mathrm{co}}\mathrm{Even}^+$ for this X. Assume n with $n \equiv 0 \vee \exists_{n'}(n' \in {}^{\mathrm{co}}\mathrm{Even} \wedge n \equiv S(Sn'))$. The goal is $n \equiv 0 \vee \exists_{n'}(n' \in ({}^{\mathrm{co}}\mathrm{Even} \cup X) \wedge n \equiv S(Sn'))$. We argue by cases on the assumed disjunction. In the first case we are done. In the second case take the n' provided.

2. Consider the algebra $\mathbb{Y}$ of binary trees. We want to represent the set of cototal binary trees (cf. Section 2.1.6) by a coinductively defined predicate ${}^{\mathrm{co}}T_{\mathbb{Y}}$. Recall that the conjunction of the two clauses of $T_{\mathbb{Y}}$ is equivalent to

$$\forall_t((t \equiv -) \vee \exists_{t_1,t_2}(t_1, t_2 \in T_{\mathbb{Y}} \wedge t \equiv \mathrm{C}t_1t_2) \to t \in T_{\mathbb{Y}}.)$$

Since $T_{\mathbb{Y}}$ is the least predicate with this property we even have equivalence

$$\forall_t(t \in T_{\mathbb{Y}} \leftrightarrow (t \equiv -) \vee \exists_{t_1,t_2}(t_1, t_2 \in T_{\mathbb{Y}} \wedge t \equiv \mathrm{C}t_1t_2)),$$

called inversion property on page 52. Now how can we formally represent cototality of a binary tree? The idea is to define ${}^{\mathrm{co}}T_{\mathbb{Y}}$ as the *largest* set satisfying the equivalence. Formulated differently, cototal ideals are not built from initial objects by construction (synthesized), but rather defined by the property that they can always be destructed (analysed). Therefore we require

$${}^{\mathrm{co}}T^-_{\mathbb{Y}} : \forall_t(t \in {}^{\mathrm{co}}T_{\mathbb{Y}} \to (t \equiv -) \vee \exists_{t_1,t_2}(t_1, t_2 \in {}^{\mathrm{co}}T_{\mathbb{Y}} \wedge t \equiv \mathrm{C}t_1t_2))$$

A set built by construction steps (synthesis) is meant to be the least set closed under these steps. Similary, a set described by destruction (analysis) is meant to be the largest set closed under destruction. Hence we require

$$ {}^{\mathrm{co}}T^{+}_{\mathbb{Y}} : \forall_t(t \in X \to (t \equiv -) \vee \exists_{t_1,t_2}(t_1, t_2 \in {}^{\mathrm{co}}T_{\mathbb{Y}} \cup X \wedge t \equiv \mathrm{C}t_1t_2)) \to X \subseteq {}^{\mathrm{co}}T_{\mathbb{Y}}. $$

${}^{\mathrm{co}}T^{+}_{\mathbb{Y}}$ expresses that every competitor X satisfying the closure property is below ${}^{\mathrm{co}}T_{\mathbb{Y}}$. As an example of how to use these axioms we prove that the equivalence above holds for ${}^{\mathrm{co}}T_{\mathbb{Y}}$ as well:

$$ \forall_t(t \in {}^{\mathrm{co}}T_{\mathbb{Y}} \leftrightarrow (t \equiv -) \vee \exists_{t_1,t_2}(t_1, t_2 \in {}^{\mathrm{co}}T_{\mathbb{Y}} \wedge t \equiv \mathrm{C}t_1t_2)), $$

"$\to$". Use the closure axiom ${}^{\mathrm{co}}T^{-}_{\mathbb{Y}}$. "$\leftarrow$". Use ${}^{\mathrm{co}}T^{+}_{\mathbb{Y}}$, with competitor $X := \{\, t \mid (t \equiv -) \vee \exists_{t_1,t_2}(t_1, t_2 \in {}^{\mathrm{co}}T_{\mathbb{Y}} \wedge t \equiv \mathrm{C}t_1t_2) \,\}$. It suffices to prove the premise of ${}^{\mathrm{co}}T^{+}_{\mathbb{Y}}$ for this X. Assume t with $(t \equiv -) \vee \exists_{t_1,t_2}(t_1, t_2 \in {}^{\mathrm{co}}T_{\mathbb{Y}} \wedge t \equiv \mathrm{C}t_1t_2)$. The goal is $(t \equiv -) \vee \exists_{t_1,t_2}(t_1, t_2 \in {}^{\mathrm{co}}T_{\mathbb{Y}} \cup X \wedge t \equiv \mathrm{C}t_1t_2)$. We argue by cases on the assumed disjunction. In the first case we are done. In the second case take the t_1, t_2 provided.

For n.c. inductive or coinductive predicates the axioms are formed as in the c.r. case, using $\vee^{\mathrm{nc}}$ for the closure axiom of ${}^{\mathrm{co}}I^{\mathrm{nc}}$. But there is an important restriction: for I^{nc} with more than one clause the elimination axiom $(I^{\mathrm{nc}})^{-}$ can only be used with a *non-computational* competitor predicate. This is needed in the proof of the soundness theorem. However, this restriction does not apply to I^{nc} defined by one clause only. Important examples of such *one-clause-nc* inductive predicates are Leibniz equality and the non-computational variants of the existential quantifier and of conjunction.

Generally, an inductive predicate is always contained in its dual.

LEMMA 3.2.1. $I \subseteq {}^{\mathrm{co}}I$, $I^{\mathrm{nc}} \subseteq {}^{\mathrm{co}}I^{\mathrm{nc}}$.

PROOF. The least-fixed-point axiom (10) for I (i.e., I^{-}) is equivalent to

$$ \forall_{\vec{x}}(\bigvee_{i<k} \exists_{\vec{x}_i}(\bigwedge_{\nu<n_i} A_{i\nu}(I \cap X) \wedge \vec{x} \equiv \vec{t}_i) \to X\vec{x}\,) \to I \subseteq X. $$

It suffices that its premise holds with ${}^{\mathrm{co}}I$ for X. To this end we use the greatest-fixed-point axiom (12) (i.e., ${}^{\mathrm{co}}I^{+}$), with the competitor predicate

$$ X := \{\, \vec{x} \mid \bigvee_{i<k} \exists_{\vec{x}_i}(\bigwedge_{\nu<n_i} A_{i\nu}(I \cap {}^{\mathrm{co}}I) \wedge \vec{x} \equiv \vec{t}_i) \,\}. $$

This means that we have to show the premise of (12) with this X, i.e.,

$$ \forall_{\vec{x}}(X\vec{x} \to \bigvee_{i<k} \exists_{\vec{x}_i}(\bigwedge_{\nu<n_i} A_{i\nu}({}^{\mathrm{co}}I \cup X) \wedge \vec{x} \equiv \vec{t}_i)). $$

But if we unfold the premise $X\vec{x}$, this follows from $I \cap {}^{\mathrm{co}}I \subseteq {}^{\mathrm{co}}I \cup X$. For I^{nc} the proof is similar. □

REMARK. In case of an inductive predicate with non-recursive clauses only also the reverse inclusions ${}^{\mathrm{co}}I_\iota \subseteq I_\iota$, ${}^{\mathrm{co}}I^{\mathrm{nc}} \subseteq I^{\mathrm{nc}}$. Hence it is not necessary to consider ${}^{\mathrm{co}}I$. Examples are the inductively defined logical connectives $\exists$, $\wedge$, $\vee$ with assigned algebras product $\times$, sum $+$ and $\mathtt{uysum}$, $\mathtt{ysumu}$, where the algebras $\mathtt{uysum}(\alpha)$, $\mathtt{ysumu}(\alpha)$ replace $\mathbb{U} + \alpha$, $\alpha + \mathbb{U}$.

LEMMA 3.2.2. $I \subseteq I^{\mathrm{nc}}$, ${}^{\mathrm{co}}I \subseteq {}^{\mathrm{co}}I^{\mathrm{nc}}$.

PROOF. Let $I := \mu_X(\forall_{\vec{x}_i}((A_{i\nu}(X))_{\nu<n_i} \to X\vec{t}_i))_{i<k}$.

For $I \subseteq I^{\mathrm{nc}}$ we use the elimination axiom (10) with I^{nc} as competitor predicate:

$$(\forall_{\vec{x}_i}((A_{i\nu}(I \cap I^{\mathrm{nc}}))_{\nu<n_i} \to I^{\mathrm{nc}}\vec{t}_i))_{i<k} \to I \subseteq I^{\mathrm{nc}}.$$

It suffices to prove the premises. Let $i < k$, fix $\vec{x}_i$ and assume $A_{i\nu}(I \cap I^{\mathrm{nc}})$ for all $\nu < n_i$. Since $A_{i\nu}(X)$ is strictly positive in X we obtain $A_{i\nu}(I^{\mathrm{nc}})$ for all $\nu < n_i$ and hence $I^{\mathrm{nc}}\vec{x}_i$ by $(I^{\mathrm{nc}})^+_i$.

For ${}^{\mathrm{co}}I \subseteq {}^{\mathrm{co}}I^{\mathrm{nc}}$ we use the greatest-fixed-point axiom for ${}^{\mathrm{co}}I^{\mathrm{nc}}$ with ${}^{\mathrm{co}}I$ as competitor predicate:

$$\forall_{\vec{x}}(X\vec{x} \to \bigvee_{i<k} \exists_{\vec{x}_i}(\bigwedge_{\nu<n_i} A_{i\nu}({}^{\mathrm{co}}I^{\mathrm{nc}} \cup {}^{\mathrm{co}}I) \wedge \vec{x} \equiv \vec{t}_i)) \to {}^{\mathrm{co}}I \subseteq {}^{\mathrm{co}}I^{\mathrm{nc}}.$$

It suffices to prove the premise, which again follows from the fact that $A_{i\nu}(X)$ is strictly positive in X. □

Recall that for the n.c. Leibniz equality the introduction axiom is $x^\rho \equiv x^\rho$ and the elimination axiom is $x \equiv y \to \forall_X Xxx \to Xxy$. From this definition we can deduce the property Leibniz used as a definition.

LEMMA 3.2.3 (Compatibility of EqD). $\forall_{x,y}(x \equiv y \to A(x) \to A(y))$.

PROOF. By the elimination axiom with $X := \{\, x, y \mid A(x) \to A(y) \,\}$. □

Using compatibility of $\equiv$ one easily proves symmetry and transitivity. Define *falsity* by $\mathbf{F} := (\mathtt{ff} \equiv \mathtt{tt})$.

THEOREM 3.2.4 (Ex-falso-quodlibet). *For every formula A we can derive $\mathbf{F} \to A$ from assumptions $\mathrm{Ef}_Y\colon \forall_{\vec{x}}(\mathbf{F} \to Y\vec{x})$ for predicate variables Y strictly positive in A, and $\mathrm{Ef}_I\colon \forall_{\vec{x}}(\mathbf{F} \to I\vec{x})$ for inductive predicates I without a nullary clause.*

PROOF. We first show $\mathrm{Ef}_{\mathrm{EqD}}\colon \mathbf{F} \to x^\rho \equiv y^\rho$. To see this, we first obtain $\mathcal{R}^\rho_{\mathbb{B}}\mathtt{ff}xy \equiv \mathcal{R}^\rho_{\mathbb{B}}\mathtt{ff}xy$ from the introduction axiom. Then from $\mathtt{ff} \equiv \mathtt{tt}$ we get $\mathcal{R}^\rho_{\mathbb{B}}\mathtt{tt}xy \equiv \mathcal{R}^\rho_{\mathbb{B}}\mathtt{ff}xy$ by compatibility. Now $\mathcal{R}^\rho_{\mathbb{B}}\mathtt{tt}xy$ converts to x and $\mathcal{R}^\rho_{\mathbb{B}}\mathtt{ff}xy$ converts to y. Hence $x^\rho \equiv y^\rho$, since we identify terms with a common reduct.

The claim can now be proved by induction on A. *Case $I\vec{s}$.* If I has no nullary clause take Ef_I. Otherwise let K_i be the nullary clause, with final conclusion $I\vec{t}$. By induction hypothesis from $\mathbf{F}$ we can derive all parameter

premises. Hence $I\vec{t}$. From $\mathbf{F}$ we also obtain $s_i \equiv t_i$, by the remark above. Hence $I\vec{s}$ by compatibility. *Case* ${}^{\mathrm{co}}I\vec{s}$. Use Lemma 3.2.1. The cases $Y\vec{s}$, $A \to B$ and $\forall_x A$ are obvious. □

A crucial use of the equality predicate EqD is that it allows us to lift a boolean term $t^{\mathbb{B}}$ to a formula, using $\mathrm{atom}(t^{\mathbb{B}}) := (t^{\mathbb{B}} \equiv \mathtt{tt})$. This opens up a convenient way to deal with equality on algebras. The computation rules ensure that, for instance, the boolean term $St =_{\mathbb{N}} Ss$, or more precisely $=_{\mathbb{N}}(St, Ss)$, is identified with $t =_{\mathbb{N}} s$. We can now turn this boolean term into the formula $(St =_{\mathbb{N}} Ss) \equiv \mathtt{tt}$, which again is abbreviated by $St =_{\mathbb{N}} Ss$, but this time with the understanding that it is a formula. Then (importantly) the two formulas $St =_{\mathbb{N}} Ss$ and $t =_{\mathbb{N}} s$ are identified because the latter is a reduct of the first. Consequently there is no need to prove the implication $St =_{\mathbb{N}} Ss \to t =_{\mathbb{N}} s$ explicitly.

Remark. One might wonder whether the weak (or classical) existential quantifier $\tilde{\exists}_x A$ is the same as the non-computational existential quantifier $\exists^{\mathrm{nc}}_x A$, since both in some sense computationally disregard the quantified variable x and the kernel A. We will argue that both quantifiers are equivalent if and only if a certain (non-computational) Markov axiom holds.

Note first that in our comparison we should use the arithmetical form $\tilde{\exists}_x A := \forall_x(A \to \mathbf{F}) \to \mathbf{F}$ rather than the logical form $\forall_x(A \to \bot) \to \bot$ of the weak existential quantifier. Otherwise there can be no relation, because $\bot$ is a predicate variable with computational content.

Clearly $\exists^{\mathrm{nc}}_x A$ implies $\tilde{\exists}_x A$, by $(\exists^{\mathrm{nc}})^-$ (take $\mathbf{F}$ for B). However, the converse is problematic. We do have an elimination scheme for $\tilde{\exists}_x A$ as well, but only for formulas B satisfying $((B \to \mathbf{F}) \to \mathbf{F}) \to B$. This means that for the converse we need a Markov axiom for the (non-computational) formula $\exists^{\mathrm{nc}}_x A$:

$$((\exists^{\mathrm{nc}}_x A \to \mathbf{F}) \to \mathbf{F}) \to \exists^{\mathrm{nc}}_x A; \tag{13}$$

from $\tilde{\exists}_x A \to \exists^{\mathrm{nc}}_x A$ we can easily derive (13). Although usage of such an n.c. axiom does not have any effect on computational content, we prefer to avoid it.

3.3. Equality and extensionality w.r.t. cotypes

The notion of equality is of central importance in any mathematical theory involving higher type objects. In our setting allowing infinite base type data the distinction between least and greatest fixed points $\mu_X \vec{K}$ and $\nu_X \vec{K}$ must be taken into account. This is already so at closed base types: similarity $\sim_{\mathbb{N}}$ and bisimilarity $\approx_{\mathbb{N}}$ are very different concepts. However, this difference is ignored in the definition of the type $\tau(C)$ of a c.r. predicate or

formula C. We therefore refine the definition of type $\tau(C)$ to *cotype* $\varphi(C)$. Using ideas from Gandy (1953, 1956) and Takeuti (1953) we can define a more appropriate concept of (pointwise) equality $\doteq_\varphi$, which is relative to a cotype φ. Extensionality Ext_φ relative to φ is defined by $(x \in \mathrm{Ext}_\varphi) := (x \doteq_\varphi x)$. We write $\doteq_C$ for $\doteq_{\varphi(C)}$ and Ext_C for $\mathrm{Ext}_{\varphi(C)}$.

3.3.1. Equality for closed base types. We take the algebra $\mathbb{Y}$ of binary trees as an example of a closed base type. In previous examples we have seen that the set of total binary trees can be represented by an inductive predicate $T_\mathbb{Y}$ (page 52), and that the set of cototal binary trees can be represented by a coinductive predicate ${}^{\mathrm{co}}T_\mathbb{Y}$ (page 55). As candidates for equality we define binary versions of $T_\mathbb{Y}$ and ${}^{\mathrm{co}}T_\mathbb{Y}$, called similarity $\sim_\mathbb{Y}$ and bisimilarity $\approx_\mathbb{Y}$. The introduction axioms for $\sim_\mathbb{Y}$ are

$$(\sim_\mathbb{Y})_0^+ \colon - \sim_\mathbb{Y} -,$$
$$(\sim_\mathbb{Y})_1^+ \colon \forall_{t_1,t_1'}(t_1 \sim_\mathbb{Y} t_1' \to \forall_{t_2,t_2'}(t_2 \sim_\mathbb{Y} t_2' \to \mathrm{C}t_1t_2 \sim_\mathbb{Y} \mathrm{C}t_1't_2'))$$

and the elimination axiom is $\sim_\mathbb{Y}^-$:

$$X(-,-) \to$$
$$\forall_{t_1,t_2}((t_1 \sim_\mathbb{Y} t_2 \land Xt_1t_2) \to \forall_{t_1',t_2'}((t_1' \sim_\mathbb{Y} t_2' \land Xt_1't_2') \to X(\mathrm{C}t_1t_2, \mathrm{C}t_1't_2'))) \to$$
$$\sim_\mathbb{Y} \subseteq X.$$

The elimination (or closure) axiom for $\approx_\mathbb{Y}$ is $\approx_\mathbb{Y}^-$:

$$\forall_{t,t'}(t \approx_\mathbb{Y} t' \to ((t \equiv -) \land (t' \equiv -)) \lor$$
$$\exists_{t_1,t_2,t_1',t_2'}(t_1 \approx_\mathbb{Y} t_1' \land t_2 \approx_\mathbb{Y} t_2' \land t \equiv \mathrm{C}t_1t_2 \land t' \equiv \mathrm{C}t_1't_2'))$$

and the introduction (or greatest-fixed-point or coinduction) axiom is $\approx_\mathbb{Y}^+$:

$$\forall_{t,t'}(Xtt' \to ((t \equiv -) \land (t' \equiv -)) \lor$$
$$\exists_{t_1,t_2,t_1',t_2'}((t_1 \approx_\mathbb{Y} t_1' \lor Xt_1t_1') \land (t_2 \approx_\mathbb{Y} t_2' \lor Xt_2t_2') \land$$
$$t \equiv \mathrm{C}t_1t_2 \land t' \equiv \mathrm{C}t_1't_2')) \to$$
$$X \subseteq {}^{\mathrm{co}}T_\mathbb{Y}.$$

As an instance of Lemma 3.2.1 we have $\sim_\mathbb{Y} \subseteq \approx_\mathbb{Y}$.

We now aim at using $\sim_\mathbb{Y}$ and $\approx_\mathbb{Y}$ for a characterization of equality at $T_\mathbb{Y}$ and ${}^{\mathrm{co}}T_\mathbb{Y}$. This is useful because it gives us a tool (induction, coinduction) to prove equalities $t \equiv t'$, which otherwise would be difficult. We will need another axiom, the Bisimilarity Axiom, which is justified be the fact that it holds in our intended model (cf. Lemma 2.1.7).

Axiom (Bisimilarity). $\forall_{t,t'}(t \approx_\mathbb{Y} t' \to t \equiv t')$.

Lemma 3.3.1 (Characterization of equality at $T_\mathbb{Y}$ and ${}^{\mathrm{co}}T_\mathbb{Y}$).

(a) $\forall_{t,t'}(t \sim_{\mathbb{Y}} t' \leftrightarrow t, t' \in T_{\mathbb{Y}} \wedge t \equiv t')$.
(b) $\forall_{t,t'}(t \approx_{\mathbb{Y}} t' \leftrightarrow t, t' \in {}^{\mathrm{co}}T_{\mathbb{Y}} \wedge t \equiv t')$.

PROOF. (b). The proof of Lemma 2.1.8 has been given in enough detail to make its formalization immediate. We need ${}^{\mathrm{co}}T^{\pm}_{\mathbb{Y}}$, $\approx^{\pm}_{\mathbb{Y}}$.

(a). Similar to (b), using $T^{\pm}_{\mathbb{Y}}$, $\sim^{\pm}_{\mathbb{Y}}$ instead. For the proof of $t \sim_{\mathbb{Y}} t' \to t \equiv t'$ use (b) and $\sim_{\mathbb{Y}} \subseteq \approx_{\mathbb{Y}}$. □

COROLLARY 3.3.2.

(a) $\forall_t(t \sim_{\mathbb{Y}} t \leftrightarrow t \in T_{\mathbb{Y}})$.
(b) $\forall_t(t \approx_{\mathbb{Y}} t \leftrightarrow t \in {}^{\mathrm{co}}T_{\mathbb{Y}})$.

PROOF. Immediate from Lemma 3.3.1. □

COROLLARY 3.3.3.

(a) $\sim_{\mathbb{Y}}$ *is an equivalence relation on* $T_{\mathbb{Y}}$.
(b) $\approx_{\mathbb{Y}}$ *is an equivalence relation on* ${}^{\mathrm{co}}T_{\mathbb{Y}}$.

PROOF. Immediate from Lemma 3.3.1. □

REMARK. For closed base types like $\mathbb{Y}$ we can also relate $\sim_{\mathbb{Y}}$ to the binary boolean-valued function $=_{\mathbb{Y}} : \mathbb{Y} \to \mathbb{Y} \to \mathbb{B}$ defined in Section 2.2.1. One easily proves that

$$\forall_t(t \in T_{\mathbb{Y}} \to t = t),$$
$$\forall_t(t \in T_{\mathbb{Y}} \to \forall_{t'}(t' \in T_{\mathbb{Y}} \to t = t' \to t \sim_{\mathbb{Y}} t')).$$

Usage of $=_{\mathbb{Y}}$ has the advantage that proofs may become shorter, since we identify terms with a common reduct.

3.3.2. Equality at higher type levels. Using cotypes we refine the assignment given in Section 3.1 (page 47) of a type $\tau(C)$ to a predicate or formula C.

DEFINITION (Cotype $\varphi(C)$ of a c.r. predicate or formula C). Assume a global injective assignment of type variables ξ to c.r. predicate variables X^c.

$$\begin{aligned}
\varphi(X^c) &:= \xi,\\
\varphi(\{\,\vec{x} \mid A\,\}) &:= \varphi(A),\\
\varphi(I(\vec{\rho}, \vec{P})) &:= \iota_I(\varphi(\vec{P}^c)),\\
\varphi({}^{\mathrm{co}}I(\vec{\rho}, \vec{P})) &:= \begin{cases} \iota_I(\varphi(\vec{P}^c)) & \text{if } \iota_I \text{ is a non-recursive algebra}\\ {}^{\mathrm{co}}\iota_I(\varphi(\vec{P}^c)) & \text{otherwise,}\end{cases}\\
\varphi(P\vec{t}) &:= \varphi(P),
\end{aligned}$$

$$\varphi(A \to B) := \begin{cases} \varphi(A) \to \varphi(B) & \text{if } A \text{ is c.r.} \\ \varphi(B) & \text{if } A \text{ is n.c.,} \end{cases}$$
$$\varphi(\forall_x A) := \varphi(A)$$

where $\vec{P}^c$ are the c.r. predicates among $\vec{P}$ and ι_I is the algebra associated with the predicate I.

In case of a non-recursive algebra ι_I the marking ${}^{\mathrm{co}}\iota_I$ is not necessary, because we not only have $I_\iota \subseteq {}^{\mathrm{co}}I_\iota$, but also the reverse inclusion ${}^{\mathrm{co}}I_\iota \subseteq I_\iota$.

Examples. $\varphi(\sim_{\mathbb{N}}) = \mathbb{N}$ and $\varphi(\approx_{\mathbb{N}}) = {}^{\mathrm{co}}\mathbb{N}$ are cotypes, and

$$\varphi(\sim_{\mathbb{L}}(\sim_{\mathbb{N}})) = \mathbb{L}(\mathbb{N}),$$
$$\varphi(\sim_{\mathbb{L}}(\approx_{\mathbb{N}})) = \mathbb{L}({}^{\mathrm{co}}\mathbb{N}),$$
$$\varphi(\approx_{\mathbb{L}}(\sim_{\mathbb{N}})) = {}^{\mathrm{co}}\mathbb{L}(\mathbb{N}),$$
$$\varphi(\approx_{\mathbb{L}}(\approx_{\mathbb{N}})) = {}^{\mathrm{co}}\mathbb{L}({}^{\mathrm{co}}\mathbb{N}).$$

Similarly, $\varphi(T_{\mathbb{N}}) = \mathbb{N}$ and $\varphi({}^{\mathrm{co}}T_{\mathbb{N}}) = {}^{\mathrm{co}}\mathbb{N}$ are cotypes, and

$$\varphi(T_{\mathbb{L}}(T_{\mathbb{N}})) = \mathbb{L}(\mathbb{N}),$$
$$\varphi(T_{\mathbb{L}}({}^{\mathrm{co}}T_{\mathbb{N}})) = \mathbb{L}({}^{\mathrm{co}}\mathbb{N}),$$
$$\varphi({}^{\mathrm{co}}T_{\mathbb{L}}(T_{\mathbb{N}})) = {}^{\mathrm{co}}\mathbb{L}(\mathbb{N}),$$
$$\varphi({}^{\mathrm{co}}T_{\mathbb{L}}({}^{\mathrm{co}}T_{\mathbb{N}})) = {}^{\mathrm{co}}\mathbb{L}({}^{\mathrm{co}}\mathbb{N}).$$

As a preparation for the definition of pointwise equality at higher type levels we generalize the examples of the predicate forms $\sim_{\mathbb{L}}$, $\approx_{\mathbb{L}}$, $T_{\mathbb{L}}$ and ${}^{\mathrm{co}}T_{\mathbb{L}}$ on page 46 from lists $\mathbb{L}$ to arbitrary algebra forms.

Definition (Similarity and bisimilarity). For every algebra form ι with type parameters $\vec{\alpha}$ we define two predicate forms $\sim_\iota$, $\approx_\iota$ (called *relative similarity* and *relative bisimilarity*) with type parameters $\vec{\alpha}$ and predicate parameters $\vec{Y}$ (where Y_i has arity (α_i, α_i)) as follows. Let $\vec{\alpha} \to (\xi)_{i<n} \to \xi$ be a constructor type. Take $(\mu/\nu)_Z(\vec{K})$, where the clause for the constructor type above is

$$Y_1 u_1 u_1' \to \cdots \to Y_n u_n u_n' \to Z v_1 v_1' \to \cdots \to Z v_m v_m' \to Z(\mathrm{C}\vec{u}\vec{v}, \mathrm{C}\vec{u}'\vec{v})$$

with C the corresponding constructor of ι. (Absolute) similarity / bisimilarity predicates arise from the relative ones by substituting a similarity / bisimilarity predicate for Y.

Definition (Pointwise equality $\doteq_\varphi$ w.r.t. a cotype φ). By recursion on $\mathrm{lev}(\varphi)$ with a subordinate recursion on the height $|\varphi|$ we define *pointwise*

equality $\doteq_\varphi$ w.r.t. a cotype φ by

$$\begin{aligned}
(x \doteq_\alpha y) &:= Yxy \quad \text{with } Y \text{ uniquely assigned to } \alpha,\\
(x \doteq_{\iota(\vec\varphi)} y) &:= (x \sim y) \quad \text{with } \sim := \sim_\iota(\doteq_{\vec\varphi}),\\
(x \doteq_{{}^{\mathrm{co}}\iota(\vec\varphi)} y) &:= (x \approx y) \quad \text{with } \approx := \approx_\iota(\doteq_{\vec\varphi}),\\
(f \doteq_{\varphi\to\psi} g) &:= \forall_{x,y}(x \doteq_\varphi y \to fx \doteq_\psi gy).
\end{aligned}$$

where the *height* $|\varphi|$ of a cotype φ is defined by $|\alpha| := 0$, $|\iota(\vec\varphi)| := |{}^{\mathrm{co}}\iota(\vec\varphi)| := 1 + \max|\vec\varphi|$ and $|\varphi\to\psi| := 1 + \max(|\varphi|, |\psi|)$.

Examples. For $\mathbb{Y}$ pointwise equality $\doteq_{\mathbb{Y}}$ is $\sim_{\mathbb{Y}}$, and pointwise equality $\doteq_{({}^{\mathrm{co}}\mathbb{Y})}$ is $\approx_{\mathbb{Y}}$. For $\mathbb{L}(\alpha)$ we need the binary predicate Y of arity (α,α) uniquely assigned to α. Then pointwise equality $\doteq_{\mathbb{L}(\alpha)}$ is $\sim_{\mathbb{L}}(Y)$, and pointwise equality $\doteq_{({}^{\mathrm{co}}\mathbb{L}(\alpha))}$ is $\approx_{\mathbb{L}}(Y)$.

Definition (Extensionality Ext_φ w.r.t. a cotype φ). Extensionality w.r.t. a cotype φ is defined by

$$(x \in \mathrm{Ext}_\varphi) := (x \doteq_\varphi x).$$

Example (A non-extensional functional). Define f, g of type $\mathbb{N}\to\mathbb{N}$ by the computation rules $fn = 0$ and $g0 = 0$, $g(Sn) = gn$. Then $f\bot_{\mathbb{N}} = 0$ because of the computation rules for f. For $g\bot_{\mathbb{N}}$ no computation rule fits, bur because of the inductive definition of $(\vec U, a) \in [\![\lambda_{\vec x} M]\!]$ in Section 2.3 $[\![g\bot_{\mathbb{N}}]\!]$ is the empty ideal $[\![\bot_{\mathbb{N}}]\!]$. Hence $f \doteq g$, i.e., $\forall_{n,m}(n \doteq_{\mathbb{N}} m \to fn \doteq_{\mathbb{N}} gm)$, since $n \doteq_{\mathbb{N}} m$ implies $n \in T_{\mathbb{N}}$ and $n \equiv m$. Therefore F defined by $Fh = h\bot_{\mathbb{N}}$ maps the pointwise equal f, g to different values.

By Corollary 3.3.2 we know the equivalence of $\mathrm{Ext}_{\mathbb{Y}}$ and $T_{\mathbb{Y}}$ (and of $\mathrm{Ext}_{({}^{\mathrm{co}}\mathbb{Y})}$ and ${}^{\mathrm{co}}T_{\mathbb{Y}}$); this also holds for arbitrary closed base cotypes. Since extensionality (i.e., the diagonalization of $\doteq$) can be reasonably defined at higher types as well (again by diagonalization of $\doteq$ at higher types) this gives us a way to generalize totality to higher types: just take extensionality. The equivalence of Ext_φ and T_φ on closed base cotypes can be extended to closed cotypes of level 1:

Lemma 3.3.4. *The predicates* Ext_φ *and* T_φ *are equivalent for closed cotypes of level* ≤ 1.

Proof. For closed base cotypes this has been proved in Corollary 3.3.2 (for the special case of the algebra $\mathbb{Y}$). In case of level 1 we use induction on the height of the cotype. Let $\varphi\to\psi$ be a closed cotype of level 1. The

following are equivalent.

$$
\begin{aligned}
&f \in \mathrm{Ext}_{\varphi\to\psi}\\
&f \doteq_{\varphi\to\psi} f\\
&\forall_{x,y}(x \doteq_\varphi y \to fx \doteq_\psi fy)\\
&\forall_{x\in T_\varphi}(fx \doteq_\psi fx) && \text{by Corollary 3.3.2, since } \mathrm{lev}(\varphi)=0\\
&\forall_{x\in T_\varphi}(fx \in \mathrm{Ext}_\psi).
\end{aligned}
$$

By induction hypothesis the final formula is equivalent to $f \in T_{\varphi\to\psi}$. □

For arbitrary closed cotypes the relation $\doteq_\varphi$ is a "partial equivalence relation", which means the following.

LEMMA 3.3.5. *For every closed cotype φ the relation $\doteq_\varphi$ is an equivalence relation on* Ext_φ.

PROOF. By induction on the height $|\varphi|$ of φ. *Case* $\iota(\vec\varphi)/^{\mathrm{co}}\iota(\vec\varphi)$. For $\sim_{\mathbb{Y}}$ and $\approx_{\mathbb{Y}}$ this was proved in Corollary 3.3.3. In the general case use the induction hypothesis and the inductive / coinductive definition of $\sim_\iota$ / $\approx_\iota$.

Case $\varphi\to\psi$. We first prove symmetry of $\doteq_{\varphi\to\psi}$. Let $f \doteq_{\varphi\to\psi} g$. The goal is $g \doteq_{\varphi\to\psi} f$. Assume $x \doteq_\varphi y$. The goal now is $gx \doteq_\psi fy$. From $x \doteq_\varphi y$ we obtain $y \doteq_\varphi x$ by symmetry of $\doteq_\varphi$, hence $fy \doteq_\psi gx$ from $f \doteq_{\varphi\to\psi} g$, hence $gx \doteq_\psi fy$ by symmetry of $\doteq_\psi$.

We finally prove transitivity of $\doteq_{\varphi\to\psi}$. Let $f \doteq_{\varphi\to\psi} g$ and $g \doteq_{\varphi\to\psi} h$. The goal is $f \doteq_{\varphi\to\psi} h$. Assume $x \doteq_\varphi y$. The goal now is $fx \doteq_\psi hy$. From $x \doteq_\varphi y$ we obtain $y \doteq_\varphi x$ by symmetry of $\doteq_\varphi$, hence $x \doteq_\varphi x$ by transitivity of $\doteq_\varphi$. Then $fx \doteq_\psi gx$ follows from $f \doteq_{\varphi\to\psi} g$. We also have $gx \doteq_\psi hy$ from $g \doteq_{\varphi\to\psi} h$. Using transitivity of $\doteq_\psi$ we obtain $fx \doteq_\psi hy$. □

LEMMA 3.3.6 (Compatibility of terms). *For every term $t(\vec x)$ with extensional constants and free variables among $\vec x$ we have*

$$\forall_{\vec x,\vec y}(\vec x \doteq_{\vec\chi} \vec y \to t(\vec x) \doteq_\varphi t(\vec y)).$$

PROOF. This is proved by induction on t. *Case* x. Immediate. *Case* c. By assumption $c \doteq_\varphi c$. *Case* $\lambda_x t(x,\vec x)$. Let $\vec x \doteq_{\vec\chi} \vec y$. The goal is $\lambda_x t(x,\vec x) \doteq_{\varphi\to\psi} \lambda_x t(x,\vec y)$, which by definition means

$$\forall_{x,y}(x \doteq_\varphi y \to t(x,\vec x) \doteq_\psi t(y,\vec y)).$$

Assume $x \doteq_\varphi y$. With $\vec x \doteq_{\vec\chi} \vec y$ the claim $t(x,\vec x) \doteq_\psi t(y,\vec y)$ holds by the IH. *Case* $t(\vec x)s(\vec x)$. Let $\vec x \doteq_{\vec\chi} \vec y$. By IH we have $t(\vec x) \doteq_{\varphi\to\psi} t(\vec y)$, i.e.,

$$\forall_{x,y}(x \doteq_\varphi y \to t(\vec x)x \doteq_\psi t(\vec y)y).$$

Again by IH we have $s(\vec x) \doteq_\varphi s(\vec y)$. Hence $t(\vec x)s(\vec x) \doteq_\psi t(\vec y)s(\vec y)$. □

LEMMA 3.3.7 (Extensionality of terms). *For every term $t(\vec{x}\,)$ with extensional constants and free variables among $\vec{x}$ we have*

$$\forall_{\vec{x}}(\vec{x} \in \mathrm{Ext}_{\vec{\chi}} \to t(\vec{x}\,) \in \mathrm{Ext}_{\varphi}).$$

PROOF. Let $t(\vec{x}\,)$ with free variables among $\vec{x}$ be given, and assume $\vec{x} \in \mathrm{Ext}_{\vec{\chi}}$. By Lemma 3.3.6 applied to $\vec{x}, \vec{x}$ we obtain $t(\vec{x}\,) \doteq_{\varphi} t(\vec{x}\,)$, hence $t(\vec{x}\,) \in \mathrm{Ext}_{\varphi}$. □