CHAPTER 3

A theory TCF of partial continuous functionals

After getting clear about the domains we intend to reason about, the partial continuous functionals, 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 will 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). Types and terms are defined as in Chapter 2. 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 closed base types considered in Section 2.2.2. 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 $X^{\mathrm{c}}, Y^{\mathrm{c}}, Z^{\mathrm{c}}, W^{\mathrm{c}} \dots$ and "non-computational" ones X^{nc}, Y^{nc}, Z^{nc}, W^{nc} ..., and use $X, Y, Z, W \dots$ for both.

Let Z be a predicate variable. By $\bar{Z}$ we denote the result of applying the predicate variable Z to a list of terms of fitting types, and by $\tilde{Z}$ lists of those.

DEFINITION (Clauses and predicate forms). Let X be a predicate variable. An *X-clause* is a formula

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

with all predicate variables Y_i^{c}, Z_i^{nc}, W_i^{nc} occurring exactly once and distinct from each other and from X, and all $\bar{X}_i$ coming from the fixed X. 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 *non-recursive* if it has no recursive premises.

Let $\vec{K}$ be a list of X-clauses. We call $I^{\mathrm{c}} := \mu_{X^{\mathrm{c}}}\vec{K}$ 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 $\mathrm{Nil}^{\mathbb{L}(\alpha)}$ and $\mathrm{Cons}^{\alpha\to\mathbb{L}(\alpha)\to\mathbb{L}(\alpha)}$ are the two constructors of the base type $\mathbb{L}(\alpha)$ of lists, written $[]$ and $::$ (infix).

1. 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. 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, 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* $\sim_{\mathbb{L}}(Y)$ defined by $\mu_X(K_0, K_1)$ and *bisimilarity* $\approx_{\mathbb{L}}(Y)$ defined by $\nu_X(K_0, K_1)$ are predicate forms with Y a parameter predicate variable.

Note that a predicate form I may contain type variables $\vec{\alpha}$ and predicate variables $\vec{Y}$. We write $I(\vec{\rho}, \vec{P})$ for the result of substituting in I the types $\vec{\rho}$ for $\vec{\alpha}$ and the predicates $\vec{P}$ for $\vec{Y}$.

DEFINITION (Constructor types of a predicate form). From every clause K we obtain a constructor type by

- omitting quantifiers,

- dropping all n.c. predicates and from the c.r. predicates their arguments, and
- replacing the remaining predicate variables by type variables.

That is, from the clause

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

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

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. 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.). 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 2.3.1, 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 ζ to c.r. predicate variables X^{c}.

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

In the I-case we have assumed $I = (\mu/\nu)_X \vec{K}$ with X-clauses $\vec{K}$. Every K_i has an assigned constructor type κ_i. Free in $\vec{\kappa}$ are the type variables $\vec{\alpha}$ from $\vec{K}$ and the type variables $\vec{\zeta}$ globally assigned to the c.r. predicate variables $\vec{Y}^{\mathrm{c}}$ in $\vec{K}$. Now $\vec{\kappa}(\vec{\rho}, \tau(\vec{P}^{\mathrm{c}}))$ is the result of substituting $\vec{\rho}$ for $\vec{\alpha}$ and of the (already generated) types $\tau(P_i^{\mathrm{c}})$ for ζ_i in $\vec{\kappa}$.

3.2. Examples of inductive predicates

A simple example of an inductive predicate is totality $T_{\mathbb{N}}$ of the natural numbers. It is defined as

$$T_{\mathbb{N}} := \mu_X(K_0, K_1)$$

with

$$\begin{aligned} K_0 &:= (0 \in X), \\ K_1 &:= \forall_n(n \in X \to Sn \in X). \end{aligned}$$

Depending on whether the predicate variable X is n.c. or c.r. we have an n.c. or a c.r. totality predicate.

Recall that a variable of type τ ranges over arbitrary objects of type τ, which may be partial. However, in practice we ofter want to argue on total objects only. To make such a restriction easy to read we introduce two sorts of variable names: a general one written $\hat{x}$ ranging over arbitrary (possibly partial) objects, and a special one written x ranging over total objects only. Then we use the abbreviation

$$\forall_x A(x) := \forall_{\hat{x}}(\hat{x} \in T_\tau \to A(\hat{x})).$$

We will follow this convention from now on. Hence the clause K_1 above should now be written

$$K_1 := \forall_{\hat{n}}(\hat{n} \in X \to S\hat{n} \in X).$$

Another particularly important example of an inductive predicate is *Leibniz equality*, defined simply by

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

We will use the abbreviation

$$(t \equiv s) := \mathrm{EqD}(t, s).$$

The missing logical connectives existence, disjunction and conjunction can also be defined inductively. Existence is defined inductively by

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

Then by definition

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

We use the abbreviation

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

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 $\exists$.

For a context where only total objects are of interest we have

$$\begin{aligned}\mathrm{ExDT}_{Y^{\mathrm{c}}} &:= \mu_{X^{\mathrm{c}}}(\forall_{\hat{x}}(\hat{x} \in T^{\mathrm{c}} \to \hat{x} \in Y^{\mathrm{c}} \to X^{\mathrm{c}})),\\ \mathrm{ExLT}_{Y^{\mathrm{c}}} &:= \mu_{X^{\mathrm{c}}}(\forall_{\hat{x}}(\hat{x} \in T^{\mathrm{c}} \to \hat{x} \in Y^{\mathrm{nc}} \to X^{\mathrm{c}})),\\ \mathrm{ExRT}_{Y^{\mathrm{c}}} &:= \mu_{X^{\mathrm{c}}}(\forall_{\hat{x}}(\hat{x} \in T^{\mathrm{nc}} \to \hat{x} \in Y^{\mathrm{c}} \to X^{\mathrm{c}})),\\ \mathrm{ExNcT}_{Y} &:= \mu_{X^{\mathrm{nc}}}(\forall_{\hat{x}}(\hat{x} \in T \to \hat{x} \in Y \to X^{\mathrm{nc}})).\end{aligned}$$

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

$$\begin{aligned}\tau(\mathrm{ExDT}) &= \mu_{\xi}(\tau \to \beta \to \xi) = \tau \times \beta\\ \tau(\mathrm{ExLT}) &= \mu_{\xi}(\tau \to \xi) \qquad = \mathbb{I}(\tau),\\ \tau(\mathrm{ExRT}) &= \mu_{\xi}(\beta \to \xi) \qquad = \mathbb{I}(\beta).\end{aligned}$$

To make these formulas more readable we can again use our convention concerning the two sorts $\hat{x}$ and x of variable names. Then the inductive predicates above are written as

$$\begin{aligned}\mathrm{ExDT}_{Y^{\mathrm{c}}} &:= \mu_{X^{\mathrm{c}}}(\forall_{x}(x \in Y^{\mathrm{c}} \to X^{\mathrm{c}})),\\ \mathrm{ExLT}_{Y^{\mathrm{c}}} &:= \mu_{X^{\mathrm{c}}}(\forall_{x}(x \in Y^{\mathrm{nc}} \to X^{\mathrm{c}})),\\ \mathrm{ExRT}_{Y^{\mathrm{c}}} &:= \mu_{X^{\mathrm{c}}}(\forall^{\mathrm{nc}}_{x}(x \in Y^{\mathrm{c}} \to X^{\mathrm{c}})),\\ \mathrm{ExNcT}_{Y} &:= \mu_{X^{\mathrm{nc}}}(\forall_{x}(x \in Y \to X^{\mathrm{nc}})).\end{aligned}$$

We use the abbreviations

$$\begin{array}{ll}
\exists^{\mathrm{d}}_x A := \mathrm{ExDT}_{\{x|A\}} & \text{if } A \text{ is c.r.,}\\
\exists^{\mathrm{l}}_x A := \mathrm{ExLT}_{\{x|A\}} & \text{if } A \text{ is n.c.,}\\
\exists^{\mathrm{r}}_x A := \mathrm{ExRT}_{\{x|A\}} & \text{if } A \text{ is c.r.,}\\
\exists^{\mathrm{nc}}_x A := \mathrm{ExNcT}_{\{x|A\}} & \text{for arbitrary } A.
\end{array}$$

Disjunction is a special case of union

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

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

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

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

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

We use the abbreviations

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

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

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

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 $\wedge$.

3.3. Axioms of TCF

We define a theory of continuous 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 distinction between n.c. and c.r. predicates and formulas 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).

3.3.1. Axioms for inductive predicates. 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

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

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^-:

$$I_i^+ \colon \forall_{\vec{x}_i}((A_{i\nu}(I))_{\nu<n_i} \to I\vec{t}_i), \tag{12}$$

$$I^- \colon (\forall_{\vec{x}_i}((A_{i\nu}(I\cap X))_{\nu<n_i} \to X\vec{t}_i))_{i<k} \to I \subseteq X \tag{13}$$

($I\cap X$ was inductively defined above). (13) 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 (13) 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 $\mathcal{R}^\tau_\iota$