CHAPTER 2

Computable functionals

The objects studied in mathematics have types, which in many cases are function types, possibly of a higher type. We are particularly interested in computable such objects. Their domains are partial continuous functionals (Scott, Ershov) which we study first, via Scott's information systems.

2.1. Information systems

The basic idea of information systems is to provide an axiomatic setting to describe approximations of abstract objects (like functions or functionals) by concrete, finite ones. We do not attempt to analyze the notion of "concreteness" or finiteness here, but rather take an arbitrary countable set A of "bits of data" or "tokens" as a basic notion to be explained axiomatically. In order to use such data to build approximations of abstract objects, we need a notion of "consistency", which determines when the elements of a finite set of tokens are consistent with each other. We also need an "entailment relation" between consistent sets U of data and single tokens a, which intuitively expresses the fact that the information contained in U is sufficient to compute the bit of information a. The axioms below are a minor modification of Scott's (1982), due to Larsen and Winskel (1991).

2.1.1. Ideals.

Definition. An *information system* is a structure $(A, \mathrm{Con}, \vdash)$ where A is an at most countable non-empty set (the *tokens*), Con is a set of finite subsets of A (the *consistent* sets) and $\vdash$ is a subset of $\mathrm{Con} \times A$ (the *entailment* relation), which satisfy

$$
\begin{aligned}
&U \subseteq V \in \mathrm{Con} \to U \in \mathrm{Con},\\
&\{a\} \in \mathrm{Con},\\
&U \vdash a \to U \cup \{a\} \in \mathrm{Con},\\
&a \in U \in \mathrm{Con} \to U \vdash a,\\
&U \in \mathrm{Con} \to \forall_{a \in V}(U \vdash a) \to V \vdash b \to U \vdash b.
\end{aligned}
$$

The elements of Con are called *formal neighborhoods*. We use U, V, W to denote *finite* sets, and write

$$
\begin{array}{lll}
U \vdash V & \text{for} & U \in \mathrm{Con} \land \forall_{a \in V}(U \vdash a), \\
a \uparrow b & \text{for} & \{a, b\} \in \mathrm{Con} \qquad (a, b \text{ are } \textit{consistent}), \\
U \uparrow V & \text{for} & \forall_{a \in U, b \in V}(a \uparrow b).
\end{array}
$$

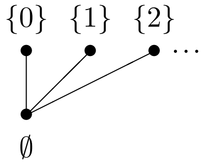
Definition. The *ideals* (also called *objects*) of an information system $\boldsymbol{A} = (A, \mathrm{Con}, \vdash)$ are defined to be those subsets x of A which satisfy

$$
\begin{array}{l}
U \subseteq x \to U \in \mathrm{Con} \quad (x \text{ is } \textit{consistent}), \\
U \vdash a \to U \subseteq x \to a \in x \quad (x \text{ is } \textit{deductively closed}).
\end{array}
$$

We write $x \in |\boldsymbol{A}|$ to mean that x is an ideal of $\boldsymbol{A}$.

Examples. The *deductive closure* $\overline{U} := \{\, a \in A \mid U \vdash a \,\}$ of $U \in \mathrm{Con}$ is an ideal.

Every countable set A can be turned into a "flat" information system by letting the set of tokens be A, $\mathrm{Con} := \{\emptyset\} \cup \{\, \{a\} \mid a \in A \,\}$ and $U \vdash a$ mean $a \in U$. In this case the ideals are just the elements of Con. For $A = \mathbb{N}$ we have the following picture of the Con-sets.



A rather important example is the following, which concerns approximations of functions from a countable set A into a countable set B. The tokens are the pairs (a, b) with $a \in A$ and $b \in B$, and

$$
\begin{array}{l}
\mathrm{Con} := \{\, \{\, (a_i, b_i) \mid i < k \,\} \mid \forall_{i,j<k}(a_i = a_j \to b_i = b_j) \,\}, \\
U \vdash (a, b) := (a, b) \in U.
\end{array}
$$

It is easy to verify that this defines an information system whose ideals are (the graphs of) all partial functions from A to B.

Remark. One can show that for an arbitrary information system $\boldsymbol{A} = (A, \mathrm{Con}, \vdash)$ the structure $(|\boldsymbol{A}|, \subseteq, \overline{\emptyset})$ is a "domain" (also called Scott-Ershov domain, or "bounded complete algebraic cpo"), whose set of "compact elements" can be represented as $|\boldsymbol{A}|_c = \{\, \overline{U} \mid U \in \mathrm{Con} \,\}$. The converse holds as well: every countable domain can be represented as an information system. We will not need this relation to standard (non-constructive) domain theory (cf. Abramsky and Jung (1994)), and hence not even define these notions here.

2.1.2. Function spaces. We define the "function space" $\boldsymbol{A} \to \boldsymbol{B}$ between two information systems $\boldsymbol{A}$ and $\boldsymbol{B}$.

Definition. Let $\boldsymbol{A} = (A, \mathrm{Con}_A, \vdash_A)$ and $\boldsymbol{B} = (B, \mathrm{Con}_B, \vdash_B)$ be information systems. Define $\boldsymbol{A} \to \boldsymbol{B} = (C, \mathrm{Con}, \vdash)$ by

$$C := \mathrm{Con}_A \times B,$$

$$\{\, (U_i, b_i) \mid i \in I \,\} \in \mathrm{Con} := \forall_{J \subseteq I} \Bigl(\bigcup_{j \in J} U_j \in \mathrm{Con}_A \to \{\, b_j \mid j \in J \,\} \in \mathrm{Con}_B \Bigr).$$

For the definition of the entailment relation $\vdash$ it is helpful to first define the notion of an *application* of $W := \{\, (U_i, b_i) \mid i \in I \,\} \in \mathrm{Con}$ to $U \in \mathrm{Con}_A$:

$$\{\, (U_i, b_i) \mid i \in I \,\} U := \{\, b_i \mid U \vdash_A U_i \,\}.$$

From the definition of Con we know that this set is in Con_B. Now define $W \vdash (U, b)$ by $WU \vdash_B b$.

Remark. Clearly application is *monotone in the second argument*, in the sense that $U \vdash_A U'$ implies ($WU' \subseteq WU$, hence also) $WU \vdash_B WU'$. In fact, application is also *monotone in the first argument*, i.e.,

$$W \vdash W' \quad \text{implies} \quad WU \vdash_B W'U.$$

To see this let $W = \{\, (U_i, b_i) \mid i \in I \,\}$ and $W' = \{\, (U'_j, b'_j) \mid j \in J \,\}$. By definition $W'U = \{\, b'_j \mid U \vdash_A U'_j \,\}$. Now fix j such that $U \vdash_A U'_j$; we must show $WU \vdash_B b'_j$. By assumption $W \vdash (U'_j, b'_j)$, hence $WU'_j \vdash_B b'_j$. Because of $WU \supseteq WU'_j$ the claim follows.

Lemma 2.1.1. *If $\boldsymbol{A}$ and $\boldsymbol{B}$ are information systems, then so is $\boldsymbol{A} \to \boldsymbol{B}$ defined as above.*

Proof. Let $\boldsymbol{A} = (A, \mathrm{Con}_A, \vdash_A)$ and $\boldsymbol{B} = (B, \mathrm{Con}_B, \vdash_B)$. The first, second and fourth property of the definition are clearly satisfied. For the third, suppose

$$\{(U_1, b_1), \dots, (U_n, b_n)\} \vdash (U, b), \quad \text{i.e.,} \quad \{\, b_j \mid U \vdash_A U_j \,\} \vdash_B b.$$

We have to show that $\{(U_1, b_1), \dots, (U_n, b_n), (U, b)\} \in \mathrm{Con}$. So let $I \subseteq \{1, \dots, n\}$ and suppose

$$U \cup \bigcup_{i \in I} U_i \in \mathrm{Con}_A.$$

We must show that $\{b\} \cup \{\, b_i \mid i \in I \,\} \in \mathrm{Con}_B$. Let $J \subseteq \{1, \dots, n\}$ consist of those j with $U \vdash_A U_j$. Then also

$$U \cup \bigcup_{i \in I} U_i \cup \bigcup_{j \in J} U_j \in \mathrm{Con}_A.$$

Since

$$\bigcup_{i\in I} U_i \cup \bigcup_{j\in J} U_j \in \mathrm{Con}_A,$$

from the consistency of $\{(U_1, b_1), \dots, (U_n, b_n)\}$ we can conclude that

$$\{\, b_i \mid i \in I \,\} \cup \{\, b_j \mid j \in J \,\} \in \mathrm{Con}_B.$$

But $\{\, b_j \mid j \in J \,\} \vdash_B b$ by assumption. Hence

$$\{\, b_i \mid i \in I \,\} \cup \{\, b_j \mid j \in J \,\} \cup \{b\} \in \mathrm{Con}_B.$$

For the final property, suppose

$$W \vdash W' \quad \text{and} \quad W' \vdash (U, b).$$

We have to show $W \vdash (U, b)$, i.e., $WU \vdash_B b$. We obtain $WU \vdash_B W'U$ by monotonicity in the first argument, and $W'U \vdash_B b$ by definition. □

We shall now give an alternative characterization of the ideals in $\boldsymbol{A} \to \boldsymbol{B}$, as "approximable maps". The basic idea for approximable maps is the desire to study "information respecting" maps from $\boldsymbol{A}$ into $\boldsymbol{B}$. Such a map is given by a relation r between Con_A and B, where $(U, b) \in r$ intuitively means that whenever we are given the information $U \in \mathrm{Con}_A$, then we know that at least the token b appears in the value.

Definition. Let $\boldsymbol{A} = (A, \mathrm{Con}_A, \vdash_A)$ and $\boldsymbol{B} = (B, \mathrm{Con}_B, \vdash_B)$ be information systems. A relation $r \subseteq \mathrm{Con}_A \times B$ is an *approximable map* if it satisfies the following:

(a) if $(U, b_1), \dots, (U, b_n) \in r$, then $\{b_1, \dots, b_n\} \in \mathrm{Con}_B$;
(b) if $(U, b_1), \dots, (U, b_n) \in r$ and $\{b_1, \dots, b_n\} \vdash_B b$, then $(U, b) \in r$;
(c) if $(U', b) \in r$ and $U \vdash_A U'$, then $(U, b) \in r$.

Theorem 2.1.2. *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be information systems. Then the ideals of $\boldsymbol{A} \to \boldsymbol{B}$ are exactly the approximable maps from $\boldsymbol{A}$ to $\boldsymbol{B}$.*

The proof is left as an exercise.

2.1.3. Continuous functions. We can also characterize approximable maps in a different way, which is closer to usual characterizations of continuity[1]:

Lemma 2.1.3. *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be information systems and $f\colon |\boldsymbol{A}| \to |\boldsymbol{B}|$ monotone (i.e., $x \subseteq y$ implies $f(x) \subseteq f(y)$). Then the following are equivalent.*

(a) *f satisfies the "principle of finite support" PFS: If $b \in f(x)$, then $b \in f(\overline{U})$ for some $U \subseteq x$.*

[1] In fact, approximable maps are exactly the continuous functions w.r.t. the so-called Scott topology. However, we will not enter this subject here.

(b) *f commutes with directed unions: for every directed $D \subseteq |\boldsymbol{A}|$ (i.e., for any $x, y \in D$ there is a $z \in D$ such that $x, y \subseteq z$)*

$$f\Big(\bigcup_{x \in D} x\Big) = \bigcup_{x \in D} f(x).$$

Note that in (b) the set $\{\, f(x) \mid x \in D \,\}$ is directed by monotonicity of f; hence its union is indeed an ideal in $|\boldsymbol{B}|$. Note also that from PFS and monotonicity of f it follows immediately that if $V \subseteq f(x)$, then $V \subseteq f(\overline{U})$ for some $U \subseteq x$.

Proof. Let f satisfy PFS, and $D \subseteq |\boldsymbol{A}|$ be directed. $f(\bigcup_{x \in D} x) \supseteq \bigcup_{x \in D} f(x)$ follows from monotonicity. For the reverse inclusion let $b \in f(\bigcup_{x \in D} x)$. Then by PFS $b \in f(\overline{U})$ for some $U \subseteq \bigcup_{x \in D} x$. From the directedness and the fact that U is finite we obtain $U \subseteq z$ for some $z \in D$. From $b \in f(\overline{U})$ and monotonicity infer $b \in f(z)$. Conversely, let f commute with directed unions, and assume $b \in f(x)$. Then

$$b \in f(x) = f(\bigcup_{U \subseteq x} \overline{U}) = \bigcup_{U \subseteq x} f(\overline{U}),$$

hence $b \in f(\overline{U})$ for some $U \subseteq x$. □

We call a function $f \colon |\boldsymbol{A}| \to |\boldsymbol{B}|$ continuous if it satifies the conditions in Lemma 2.1.3. Hence continuous maps $f \colon |\boldsymbol{A}| \to |\boldsymbol{B}|$ are those that can be completely described from the point of view of finite approximations of the abstract objects $x \in |\boldsymbol{A}|$ and $f(x) \in |\boldsymbol{B}|$: whenever we are given a finite approximation V to the value $f(x)$, then there is a finite approximation U to the argument x such that already $f(\overline{U})$ contains the information in V; note that by monotonicity $f(\overline{U}) \subseteq f(x)$.

Clearly the identity and constant functions are continuous, and also the *composition* $g \circ f$ of continuous functions $f \colon |\boldsymbol{A}| \to |\boldsymbol{B}|$ and $g \colon |\boldsymbol{B}| \to |\boldsymbol{C}|$.

Theorem 2.1.4. *Let $\boldsymbol{A} = (A, \mathrm{Con}_A, \vdash_A)$, $\boldsymbol{B} = (B, \mathrm{Con}_B, \vdash_B)$ be information systems. Then the ideals of $\boldsymbol{A} \to \boldsymbol{B}$ are in a natural bijective correspondence with the continuous functions from $|\boldsymbol{A}|$ to $|\boldsymbol{B}|$, as follows.*

(a) *With any approximable map $r \subseteq \mathrm{Con}_A \times B$ we can associate a continuous function $|r| \colon |\boldsymbol{A}| \to |\boldsymbol{B}|$ by*

$$|r|(z) := \{\, b \in B \mid (U, b) \in r \text{ for some } U \subseteq z \,\}.$$

We call $|r|(z)$ the application *of r to z.*

(b) *Conversely, with any continuous function $f \colon |\boldsymbol{A}| \to |\boldsymbol{B}|$ we can associate an approximable map $\hat{f} \subseteq \mathrm{Con}_A \times B$ by*

$$\hat{f} := \{\, (U, b) \mid b \in f(\overline{U}) \,\}.$$

These assignments are inverse to each other, i.e., $f = |\hat{f}|$ and $r = \widehat{|r|}$.

The proof is left as an exercise.

Consequently we can (and will) view approximable maps $r \subseteq \mathrm{Con}_A \times B$ as continuous functions from $|\boldsymbol{A}|$ to $|\boldsymbol{B}|$.

Equality of two subsets $r, s \subseteq \mathrm{Con}_A \times B$ means that they consist of the same tokens (U, b). We can characterize equality $r = s$ by extensional equality of the associated functions $|r|, |s|$. It even suffices that $|r|$ and $|s|$ coincide on all compact elements $\overline{U}$ for $U \in \mathrm{Con}_A$.

Lemma 2.1.5 (Extensionality). *Assume that $\boldsymbol{A} = (A, \mathrm{Con}_A, \vdash_A)$ and $\boldsymbol{B} = (B, \mathrm{Con}_B, \vdash_B)$ are information systems and $r, s \subseteq \mathrm{Con}_A \times B$ approximable maps. Then the following are equivalent.*

(a) $r = s$,
(b) $|r|(z) = |s|(z)$ *for all* $z \in |\boldsymbol{A}|$,
(c) $|r|(\overline{U}) = |s|(\overline{U})$ *for all* $U \in \mathrm{Con}_A$.

Proof. It suffices to prove (c) → (a). As above this follows from

$$\begin{aligned}(U, b) \in r &\leftrightarrow \exists_{V \subseteq \overline{U}} (V, b) \in r \quad \text{by axiom (c) for approximable maps}\\ &\leftrightarrow b \in |r|(\overline{U}).\end{aligned}$$ □

Moreover, one can easily check that

$$s \circ r := \{\, (U, c) \mid \exists_V ((V, c) \in s \wedge (U, V) \subseteq r) \,\}$$

is an approximable map (where $(U, V) := \{\, (U, b) \mid b \in V \,\}$), and

$$|s \circ r| = |s| \circ |r|, \quad \widehat{g \circ f} = \hat{g} \circ \hat{f}.$$

We usually write $r(z)$ for $|r|(z)$, and similarly $(U, b) \in f$ for $(U, b) \in \hat{f}$. It should always be clear from the context where the mods and hats should be inserted.

2.2. Partial continuous functionals

We now use information systems to define the Scott-Ershov model of partial continuous functionals, each of a given "type".

2.2.1. Types. What is a type? Clearly if τ and σ are types, then we want that also $\tau \to \sigma$ is a type, to be called "function type". But we have to start somewhere. The basic idea is that we consider finite lists of (named) "constructor types".

Types may involve type variables $\alpha, \beta, \gamma, \xi, \zeta$. Iterated arrows are understood as associated to the right. For example, $\alpha \to \beta \to \gamma$ means $\alpha \to (\beta \to \gamma)$, not $(\alpha \to \beta) \to \gamma$.

Definition. *Constructor types* κ have the form

$$\vec{\alpha} \to (\xi)_{i<n} \to \xi$$

with all type variables α_i distinct from each other and from ξ. An argument type of a constructor type is called a *parameter* argument type if it is different from ξ, and a *recursive* argument type otherwise. A constructor type is *recursive* if it has a recursive argument type. Each list of named constructor types with all of its parameter argument types distinct determines a *base type* $\iota_{\vec{\kappa}}$. A base type given by a list of named constructor is sometimes called *algebra*.

For some common lists of named constructor types there are standard names for the corresponding base types:

$\mathrm{Dummy}\colon \xi$	$\mathbb{U}$ (unit),
$\mathsf{tt}\colon \xi, \mathsf{ff}\colon \xi$	$\mathbb{B}$ (booleans),
$\mathrm{SdL}\colon \xi, \mathrm{SdM}\colon \xi, \mathrm{SdR}\colon \xi$	$\mathbb{D}$ (signed digits),
$\mathrm{Zero}\colon \xi, \mathrm{Succ}\colon \xi \to \xi$	$\mathbb{N}$ (natural numbers, unary),
$\mathrm{One}\colon \xi, \mathrm{S}_0\colon \xi \to \xi, \mathrm{S}_1\colon \xi \to \xi$	$\mathbb{P}$ (positive numbers, binary),
$\mathrm{L}\colon \xi, \mathrm{B}\colon \xi \to \xi \to \xi$	$\mathbb{Y}$ (binary trees)

and with parameter types

$\mathrm{Id}\colon \alpha \to \xi$	$\mathbb{I}(\alpha)$ (identity),
$\mathrm{Nil}\colon \xi, \mathrm{Cons}\colon \alpha \to \xi \to \xi$	$\mathbb{L}(\alpha)$ (lists),
$\mathrm{SCons}\colon \alpha \to \xi \to \xi$	$\mathbb{S}(\alpha)$ (streams),
$\mathrm{Pair}\colon \alpha \to \beta \to \xi$	$\alpha \times \beta$ (product),
$\mathrm{InL}\colon \alpha \to \xi, \mathrm{InR}\colon \beta \to \xi$	$\alpha + \beta$ (sum),
$\mathrm{DummyL}\colon \xi, \mathrm{Inr}\colon \alpha \to \xi$	$\mathrm{uysum}(\alpha)$ (for $\mathbb{U} + \alpha$),
$\mathrm{Inl}\colon \alpha \to \xi, \mathrm{DummyR}\colon \xi$	$\mathrm{ysumu}(\alpha)$ (for $\alpha + \mathbb{U}$).

Definition. *Types* are inductively defined by

(a) Every type variable α is a type.

(b) If $\vec{\kappa}(\vec{\alpha}\,)$ is a list of named constructor types and $\vec{\tau}$ are types where the length of $\vec{\tau}$ is the number of parameters in $\vec{\kappa}$, then $\iota_{\vec{\kappa}(\vec{\tau}\,)}$ is a type.

(c) It τ and σ are types, then so is $\tau \to \sigma$.

Types of the form $\tau \to \sigma$ are called *function types*, and types of the form $\iota_{\vec{\kappa}(\vec{\tau}\,)}$ *base types.*

If the base type corresponding to a list of named constructor types has a standard name, then we use this name to denote the base type.

A type is *closed* it it has no parameters. Let $\tau(\vec{\alpha}\,)$ be a type with $\vec{\alpha}$ its parameters, and let $\vec{\rho}$ be closed types. We define the *level* of $\tau(\vec{\rho}\,)$ by

$$\begin{aligned} &\mathrm{lev}(\iota_{\vec{\kappa}(\vec{\rho}\,)}) := \max(\mathrm{lev}(\vec{\rho}\,)), \\ &\qquad\qquad \text{where the length of } \vec{\rho} \text{ is the number of parameters in } \vec{\kappa}(\vec{\alpha}\,), \\ &\mathrm{lev}(\tau \to \sigma) := \max(\mathrm{lev}(\sigma)), 1 + \mathrm{lev}(\tau)). \end{aligned}$$

Examples of base types:

- $\mathbb{L}(\alpha)$, $\mathbb{L}(\mathbb{L}(\alpha))$, $\alpha \times \beta$ are base types of level 0.
- $\mathbb{L}(\mathbb{L}(\mathbb{N}))$, $\mathbb{N} + \mathbb{B}$, $\mathbb{Z} := \mathbb{P} + \mathbb{U} + \mathbb{P}$, $\mathbb{Q} := \mathbb{Z} \times \mathbb{P}$ are closed base types of level 0.
- $\mathbb{R} := (\mathbb{N} \to \mathbb{Q}) \times (\mathbb{P} \to \mathbb{N})$ is a closed base type of level 1.

2.2.2. The information systems $\boldsymbol{A}_\tau$. For every closed type τ we define the information system $\boldsymbol{A}_\tau = (A_\tau, \mathrm{Con}_\tau, \vdash_\tau)$. The ideals $x \in |\boldsymbol{A}_\tau|$ are the *partial continuous functionals* of type τ. Since we will have $\boldsymbol{A}_{\tau\to\sigma} = \boldsymbol{A}_\tau \to \boldsymbol{A}_\sigma$, the partial continuous functionals of type $\tau \to \sigma$ will correspond to the continuous functions from $|\boldsymbol{A}_\tau|$ to $|\boldsymbol{A}_\sigma|$.

Definition (Information system of a closed type τ). We simultaneously define $A_{\iota_{\vec{\kappa}(\vec{\tau}\,)}}$, $A_{\tau\to\sigma}$, $\mathrm{Con}_{\iota_{\vec{\kappa}(\vec{\tau}\,)}}$ and $\mathrm{Con}_{\tau\to\sigma}$.

(a) The *tokens* $a \in A_{\iota_{\vec{\kappa}(\vec{\tau}\,)}}$ are the type correct constructor expressions

$$\mathrm{C}U_1 \dots U_m a_1^* \dots a_n^*$$

with C the name of a constructor type $\vec{\alpha} \to (\xi)_{i<n} \to \xi$ from $\vec{\kappa}$, all U_j ($j < m$) from Con_{τ_j} and each a_i^* ($i < n$) an *extended token*, i.e., a token or the special symbol $*$ which carries no information.

(b) The tokens in $A_{\tau\to\sigma}$ are the pairs (U, b) with $U \in \mathrm{Con}_\tau$ and $b \in A_\sigma$.

(c) A finite set U of tokens in $A_{\iota_{\vec{\kappa}(\vec{\tau}\,)}}$ is *consistent* (i.e., $U \in \mathrm{Con}_{\iota_{\vec{\kappa}(\vec{\tau}\,)}}$) if

(i) all its elements start with the same constructor C, say of arity $\vec{\tau} \to (\iota_{\vec{\kappa}(\vec{\tau}\,)})_{i<n} \to \iota_{\vec{\kappa}(\vec{\tau}\,)}$,

(ii) the union V_j of all Con-sets at the j-th ($j < m$) argument position of some token in U is in Con_{τ_j}, and

(iii) all $U_i \in \mathrm{Con}_{\iota_{\vec{\kappa}(\vec{\tau}\,)}}$ ($i < n$), where U_i consists of all (proper) tokens at the $(m + i)$-th argument position of some token in U.

(d) $\{\, (U_i, b_i) \mid i \in I \,\} \in \mathrm{Con}_{\tau\to\sigma}$ is defined to mean

$$\forall_{J\subseteq I}(\bigcup_{j\in J} U_j \in \mathrm{Con}_\tau \to \{\, b_j \mid j \in J \,\} \in \mathrm{Con}_\sigma).$$

Building on this definition, we define $U \vdash_\tau a$ for $U \in \mathrm{Con}_\tau$ and $a \in A_\tau$.

(e) $\{\mathrm{C}\vec{U}_1\vec{a_1^*}, \dots, \mathrm{C}\vec{U}_l\vec{a_l^*}\} \vdash_{\iota_{\vec{\kappa}(\vec{\tau}\,)}} \mathrm{C}'\vec{V}\vec{a^*}$ is defined to mean $\mathrm{C} = \mathrm{C}'$, $l \geq 1$, V_j as in (c) above and $U_i \vdash a_i^*$, with U_i as in (c) above (and $U \vdash *$ taken to be true).

(f) $W \vdash_{\tau\to\sigma} (U, b)$ is defined to mean $WU \vdash_\sigma b$, where application WU of $W = \{\,(U_i, b_i) \mid i \in I\,\} \in \mathrm{Con}_{\tau\to\sigma}$ to $U \in \mathrm{Con}_\tau$ is defined to be $\{\,b_i \mid U \vdash_\tau U_i\,\}$; recall that $U \vdash V$ abbreviates $\forall_{a\in V}(U \vdash a)$.

Note that the present definition is by recursion on the *height* of the syntactic expressions involved, defined by

$$\begin{aligned}
|\alpha| &:= 0,\\
|\iota_{\vec{\kappa}(\vec{\tau})}| &:= 1 + \max\{\,|\tau_i| \mid \tau_i \in \vec{\tau}\,\},\\
|\tau \to \sigma| &:= \max\{1 + |\tau|, |\sigma|\},\\
|\mathrm{C}U_1 \dots U_m a_1^* \dots a_n^*| &:= 1 + \max(\{\,|U_j| \mid 1 \le j \le m\,\} \cup \{\,|a_i^*| \mid 1 \le i \le n\,\}),\\
|\,*\,| &:= 0,\\
|(U, b)| &:= 1 + \max\{|U|, |b|\},\\
|\{\,a_i \mid i \in I\,\}| &:= 1 + \max\{\,|a_i| \mid i \in I\,\},\\
|U \vdash a| &:= 1 + \max\{|U|, |a|\}.
\end{aligned}$$

It is easy to see that $(A_\tau, \mathrm{Con}_\tau, \vdash_\tau)$ is an information system. Observe that all the notions involved are computable: $a \in A_\tau$, $U \in \mathrm{Con}_\tau$ and $U \vdash_\tau a$.

Definition (Partial continuous functionals). For every closed type τ let $\boldsymbol{A}_\tau$ be the information system $(A_\tau, \mathrm{Con}_\tau, \vdash_\tau)$. The set $|\boldsymbol{A}_\tau|$ of ideals in $\boldsymbol{A}_\tau$ is the set of *partial continuous functionals* of type τ. A partial continuous functional $x \in |\boldsymbol{A}_\tau|$ is *computable* if it is recursively enumerable when viewed as a set of tokens.

Notice that $\boldsymbol{A}_{\tau\to\sigma} = \boldsymbol{A}_\tau \to \boldsymbol{A}_\sigma$ as defined generally for information systems.

For example, the tokens for the base type $\mathbb{N}$ are shown in Figure 1 (with 0 for Zero and S for Succ). For tokens a, b we have $\{a\} \vdash b$ if and only if there is a path from a (up) to b (down). As another example, consider the base type $\mathbb{Y}$ of binary trees with a nullary constructor L (for Leaf) and a binary B (for Branch). Then $\{B(L, *), B(*, L)\}$ is consistent, and it entails $B(L, L)$.

2.2.3. Ideals of a closed base type. Let τ be a closed base type, for simplicity without parameters. An example is the type $\mathbb{Y}$ of binary trees. We want to take a closer look at the elements of $|\boldsymbol{A}_\tau|$, i.e., the ideals in $\boldsymbol{A}_\tau$. Among them it seems natural to single out those with the following property, to be called "cototality":

Definition (Cototal ideal). Consider a token in the ideal, and in this token (a constructor expression) a position occupied by the symbol $*$ (indicating "no information"). Then it must be possible to further analyze the

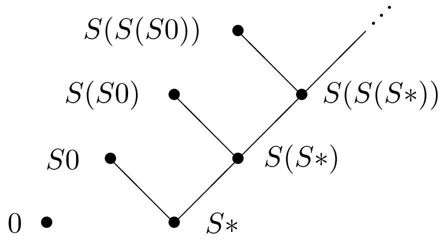


FIGURE 1. Tokens and entailment for ℕ

ideal, in the following sense. There must be another token in the ideal where this symbol $*$ is replaced by a constructor expression $\mathrm{C}\vec{*}$ with C the name of a constructor of the underlying base type.

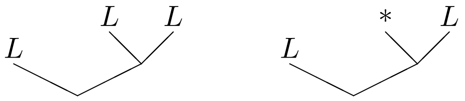
Clearly cototal ideals may be infinite. However, they can be analyzed (or “destructed”) up to an arbitrary depth. It may also happen that a cototal ideal is finite. In this case it is called “total”.

Hence in our model of partial continuous functionals already at base types we have ideals (i.e., objects) which are either

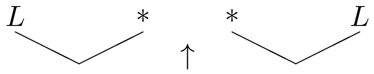
- cototal and infinite, or
- total (i.e., cototal and finite), or
- neither.

This is a somewhat uncommon view of what we should understand as an object of for instance the base type $\mathbb{Y}$ of binary trees. However, it fits perfectly with the type theory TCF allowing inductive and coinductive predicates to be studied in Chapter 3.

We give some examples for the base type $\mathbb{Y}$, with a more pictorial representation. Tokens in $\boldsymbol{A}_{\mathbb{Y}}$ (omitting B):



Consistency in $\boldsymbol{A}_{\mathbb{Y}}$:



Moreover