Plotkin Definability Theorem for Atomic-Coherent Information Systems

Basil Karádais

Institute of Mathematics
Ludwig-Maximilian University of Munich

June 18, 2008

Atomicity and Coherence in Scott Information Systems

Let $\alpha = (T, \mathsf{Con}, \vdash)$ be a *Scott information system* [Scott 1982].
Call it

- *atomic* when for all $U \in \mathsf{Con}$

$$U \vdash b \to \exists_{a \in U} \{a\} \vdash b$$

- *coherent* when for all $a_1, \dots, a_m \in T$

$$\left(\forall_{1 \le i,j \le m} \{a_i, a_j\} \in \mathsf{Con} \right) \to \{a_1, \dots, a_m\} \in \mathsf{Con}$$

On the level of *ideals* atomicity is benign, whereas coherence results in richer domains. For our purposes it is safe to require the latter as well.

Atomicity and Coherence in Scott Information Systems

Let $\alpha = (T, \mathrm{Con}, \vdash)$ be a *Scott information system* [Scott 1982]. Call it

- *atomic* when for all $U \in \mathrm{Con}$

$$U \vdash b \to \exists_{a \in U} \{a\} \vdash b$$

- *coherent* when for all $a_1, \ldots, a_m \in T$

$$\left(\forall_{1 \leq i,j \leq m} \{a_i, a_j\} \in \mathrm{Con} \right) \to \{a_1, \ldots, a_m\} \in \mathrm{Con}$$

On the level of *ideals* atomicity is benign, whereas coherence results in richer domains. For our purposes it is safe to require the latter as well.

Atomicity and Coherence in Scott Information Systems

Let $\alpha = (T, \mathsf{Con}, \vdash)$ be a *Scott information system* [Scott 1982]. Call it

- *atomic* when for all $U \in \mathsf{Con}$

$$U \vdash b \to \underset{a \in U}{\exists} \{a\} \vdash b$$

- *coherent* when for all $a_1, \dots, a_m \in T$

$$\left(\underset{1 \le i,j \le m}{\forall} \{a_i, a_j\} \in \mathsf{Con} \right) \to \{a_1, \dots, a_m\} \in \mathsf{Con}$$

On the level of *ideals* atomicity is benign, whereas coherence results in richer domains. For our purposes it is safe to require the latter as well.

Atomicity and Coherence in Scott Information Systems

Let $\alpha = (T, \mathsf{Con}, \vdash)$ be a *Scott information system* [Scott 1982]. Call it

- *atomic* when for all $U \in \mathsf{Con}$

$$U \vdash b \to \underset{a \in U}{\exists} \{a\} \vdash b$$

- *coherent* when for all $a_1, \ldots, a_m \in T$

$$\left(\underset{1 \leq i,j \leq m}{\forall} \{a_i, a_j\} \in \mathsf{Con} \right) \to \{a_1, \ldots, a_m\} \in \mathsf{Con}$$

On the level of *ideals* atomicity is benign, whereas coherence results in richer domains. For our purposes it is safe to require the latter as well.

Acises

An *atomic-coherent information system* (acis) [Schwichtenberg 2006] is a triple

$$\alpha = (T, \diamond, \triangleright)$$

where

- *consistency* $\diamond$ is a reflexive and symmetric binary relation
- *entailment* $\triangleright$ is a reflexive and transitive binary relation
- concistency *propagates* through entailment:

$$a \diamond b \wedge b \triangleright c \to a \diamond c$$

Retrieve the *consistent sets* (or *formal neighborhoods*) by

$$U \in \mathsf{Con} :\Leftrightarrow U \subseteq^f T \wedge \bigforall_{a,b \in U} a \diamond b$$

and define *ideals* by

$$u \in \mathsf{Ide} :\Leftrightarrow \bigforall_{a,b \in u} a \diamond b \wedge \bigforall_{a \in u} . \, a \triangleright b \to b \in u$$

Acises

An *atomic-coherent information system* (acis) [Schwichtenberg 2006] is a triple

$$\alpha = (T, \diamond, \triangleright)$$

where

- *consistency* $\diamond$ is a reflexive and symmetric binary relation
- *entailment* $\triangleright$ is a reflexive and transitive binary relation
- concistency *propagates* through entailment:

$$a \diamond b \wedge b \triangleright c \rightarrow a \diamond c$$

Retrieve the *consistent sets* (or *formal neighborhoods*) by

$$U \in \mathsf{Con} :\Leftrightarrow U \subseteq^f T \wedge \bigvee_{a,b \in U} a \diamond b$$

and define *ideals* by

$$u \in \mathsf{Ide} :\Leftrightarrow \bigvee_{a,b \in u} a \diamond b \wedge \bigvee_{a \in u} . \, a \triangleright b \rightarrow b \in u$$

Acises

An *atomic-coherent information system* (acis) [Schwichtenberg 2006] is a triple

$$\alpha = (T, \diamond, \triangleright)$$

where

- *consistency* $\diamond$ is a reflexive and symmetric binary relation
- *entailment* $\triangleright$ is a reflexive and transitive binary relation
- concistency *propagates* through entailment:

$$a \diamond b \wedge b \triangleright c \to a \diamond c$$

Retrieve the *consistent sets* (or *formal neighborhoods*) by

$$U \in \mathsf{Con} :\Leftrightarrow U \subseteq^f T \wedge \bigvee_{a,b \in U} a \diamond b$$

and define *ideals* by

$$u \in \mathsf{Ide} :\Leftrightarrow \bigvee_{a,b \in u} a \diamond b \wedge \bigvee_{a \in u} . \, a \triangleright b \to b \in u$$

Acises

An *atomic-coherent information system* (acis) [Schwichtenberg 2006] is a triple

$$\alpha = (T, \diamond, \triangleright)$$

where

- *consistency* $\diamond$ is a reflexive and symmetric binary relation
- *entailment* $\triangleright$ is a reflexive and transitive binary relation
- concistency *propagates* through entailment:

$$a \diamond b \wedge b \triangleright c \to a \diamond c$$

Retrieve the *consistent sets* (or *formal neighborhoods*) by

$$U \in \mathsf{Con} :\Leftrightarrow U \subseteq^f T \wedge \underset{a,b \in U}{\forall} a \diamond b$$

and define *ideals* by

$$u \in \mathsf{Ide} :\Leftrightarrow \underset{a,b \in u}{\forall} a \diamond b \wedge \underset{a \in u}{\forall} . a \triangleright b \to b \in u$$

Acises

An *atomic-coherent information system* (acis) [Schwichtenberg 2006] is a triple

$$\alpha = (T, \diamond, \triangleright)$$

where

- *consistency* $\diamond$ is a reflexive and symmetric binary relation
- *entailment* $\triangleright$ is a reflexive and transitive binary relation
- concistency *propagates* through entailment:

$$a \diamond b \wedge b \triangleright c \to a \diamond c$$

Retrieve the *consistent sets* (or *formal neighborhoods*) by

$$U \in \mathsf{Con} :\Leftrightarrow U \subseteq^f T \wedge \underset{a,b \in U}{\forall} a \diamond b$$

and define *ideals* by

$$u \in \mathsf{Ide} :\Leftrightarrow \underset{a,b \in u}{\forall} a \diamond b \wedge \underset{a \in u}{\forall} .\ a \triangleright b \to b \in u$$

Function Spaces

Let $\alpha = (T_\alpha, \Diamond_\alpha, \rhd_\alpha)$ and $\beta = (T_\beta, \Diamond_\beta, \rhd_\beta)$ be two acises. Define their *function space* $\alpha \to \beta = (T, \Diamond, \rhd)$ by

$$\begin{array}{rcl} T & := & \mathsf{Con}_\alpha \times T_\beta \\ (U,a) \Diamond (V,b) & :\Leftrightarrow & U \Diamond_\alpha V \to a \Diamond_\beta b \\ (U,a) \rhd (V,b) & :\Leftrightarrow & V \rhd_\alpha U \wedge a \rhd_\beta b \end{array}$$

The triple $\alpha \to \beta$ is again an acis.
Define *application* between ideals $u = \{\ldots, (U,a), \ldots\} \in \mathsf{Ide}_{\alpha\to\beta}$ and $v \in \mathsf{Ide}_\alpha$ by

$$u(v) := \left\{ b \in T_\beta \mid \underset{(U,a)\in u}{\exists} . \, v \rhd_\alpha U \wedge a \rhd_\beta b \right\}$$

Function Spaces

Let $\alpha = (T_\alpha, \diamond_\alpha, \rhd_\alpha)$ and $\beta = (T_\beta, \diamond_\beta, \rhd_\beta)$ be two acises. Define their *function space* $\alpha \to \beta = (T, \diamond, \rhd)$ by

$$\begin{aligned} T &:= \mathsf{Con}_\alpha \times T_\beta \\ (U,a) \diamond (V,b) &:\Leftrightarrow U \diamond_\alpha V \to a \diamond_\beta b \\ (U,a) \rhd (V,b) &:\Leftrightarrow V \rhd_\alpha U \wedge a \rhd_\beta b \end{aligned}$$

The triple $\alpha \to \beta$ is again an acis.

Define *application* between ideals $u = \{\dots, (U,a), \dots\} \in \mathsf{Ide}_{\alpha\to\beta}$ and $v \in \mathsf{Ide}_\alpha$ by

$$u(v) := \left\{ b \in T_\beta \mid \exists_{(U,a)\in u} . \, v \rhd_\alpha U \wedge a \rhd_\beta b \right\}$$

Function Spaces

Let $\alpha = (T_\alpha, \diamond_\alpha, \triangleright_\alpha)$ and $\beta = (T_\beta, \diamond_\beta, \triangleright_\beta)$ be two acises. Define their *function space* $\alpha \to \beta = (T, \diamond, \triangleright)$ by

$$\begin{array}{rcl} T & := & \mathsf{Con}_\alpha \times T_\beta \\ (U,a) \diamond (V,b) & :\Leftrightarrow & U \diamond_\alpha V \to a \diamond_\beta b \\ (U,a) \triangleright (V,b) & :\Leftrightarrow & V \triangleright_\alpha U \wedge a \triangleright_\beta b \end{array}$$

The triple $\alpha \to \beta$ is again an acis.
Define *application* between ideals $u = \{\ldots, (U,a), \ldots\} \in \mathsf{Ide}_{\alpha\to\beta}$ and $v \in \mathsf{Ide}_\alpha$ by

$$u(v) := \left\{ b \in T_\beta \mid \exists_{(U,a)\in u} \,.\, v \triangleright_\alpha U \wedge a \triangleright_\beta b \right\}$$

Continuity

Write $\overline{U}$ for the *deductive closure* of a neighborhood U. An *ideal mapping* $f : \mathsf{Ide}_\alpha \to \mathsf{Ide}_\beta$ is *continuous* if

- it is monotone
$$u \subseteq v \to f(u) \subseteq f(v)$$
- and it satisfies the *principle of finite support*
$$b \in f(u) \to \underset{U \subseteq^f u}{\exists} b \in f(\overline{U})$$

The continuous ideal mappings from Ide_α to Ide_β are exactly the ideals of $\mathsf{Ide}_{\alpha \to \beta}$.

Continuity

Write $\overline{U}$ for the *deductive closure* of a neighborhood U. An *ideal mapping* $f : \mathsf{Ide}_\alpha \to \mathsf{Ide}_\beta$ is *continuous* if

- it is monotone
$$u \subseteq v \to f(u) \subseteq f(v)$$

- and it satisfies the *principle of finite support*
$$b \in f(u) \to \exists_{U \subseteq^f u} b \in f(\overline{U})$$

The continuous ideal mappings from Ide_α to Ide_β are exactly the ideals of $\mathsf{Ide}_{\alpha\to\beta}$.

Continuity

Write $\overline{U}$ for the *deductive closure* of a neighborhood U. An *ideal mapping* $f : \mathsf{Ide}_\alpha \to \mathsf{Ide}_\beta$ is *continuous* if

- it is monotone
$$u \subseteq v \to f(u) \subseteq f(v)$$
- and it satisfies the *principle of finite support*
$$b \in f(u) \to \underset{U \subseteq^f u}{\exists} b \in f(\overline{U})$$

The continuous ideal mappings from Ide_α to Ide_β are exactly the ideals of $\mathsf{Ide}_{\alpha \to \beta}$.

Continuity

Write $\overline{U}$ for the *deductive closure* of a neighborhood U. An *ideal mapping* $f : \mathrm{Ide}_\alpha \to \mathrm{Ide}_\beta$ is *continuous* if

- it is monotone
$$u \subseteq v \to f(u) \subseteq f(v)$$
- and it satisfies the *principle of finite support*
$$b \in f(u) \to \underset{U \subseteq^f u}{\exists} b \in f(\overline{U})$$

The continuous ideal mappings from Ide_α to Ide_β are exactly the ideals of $\mathrm{Ide}_{\alpha\to\beta}$.

Arithmetical and Boolean Acises

Let $*$ be a (pre)atom meaning *least atomic information*.

The algebra $\mathbb{N} = \{0, S\}$ defines a *nonflat* acis by

$$T_{\mathbb{N}} := \{*, 0, S*, S0, S(S*), S(S0), \ldots\}$$

$$\left(\bigvee_{a \in T_{\mathbb{N}}} a \diamond_{\mathbb{N}} * \wedge * \diamond_{\mathbb{N}} a\right) \wedge \left(\bigvee_{a,b \in T_{\mathbb{N}}} . \; a \diamond_{\mathbb{N}} b \to Sa \diamond_{\mathbb{N}} Sb\right)$$

$$\left(\bigvee_{a \in T_{\mathbb{N}}} a \rhd_{\mathbb{N}} *\right) \wedge \left(\bigvee_{a,b \in T_{\mathbb{N}}} . \; a \rhd_{\mathbb{N}} b \to Sa \rhd_{\mathbb{N}} Sb\right)$$

and the algebra $\mathbb{B} = \{\mathtt{tt}, \mathtt{ff}\}$ defines an acis by

$$T_{\mathbb{B}} := \{*, \mathtt{tt}, \mathtt{ff}\}$$

$$\bigvee_{a \in T_{\mathbb{B}}} . \; a \diamond_{\mathbb{B}} a \wedge a \diamond_{\mathbb{B}} *$$

$$\bigvee_{a \in T_{\mathbb{B}}} . \; a \rhd_{\mathbb{B}} a \wedge a \rhd_{\mathbb{B}} *$$

Arithmetical and Boolean Acises

Let $*$ be a (pre)atom meaning *least atomic information*.
The algebra $\mathbb{N} = \{0, S\}$ defines a *nonflat* acis by

$$T_{\mathbb{N}} := \{*, 0, S*, S0, S(S*), S(S0), \ldots\}$$

$$\left(\forall_{a \in T_{\mathbb{N}}} a \diamond_{\mathbb{N}} * \wedge * \diamond_{\mathbb{N}} a\right) \wedge \left(\forall_{a,b \in T_{\mathbb{N}}} . \ a \diamond_{\mathbb{N}} b \to Sa \diamond_{\mathbb{N}} Sb\right)$$

$$\left(\forall_{a \in T_{\mathbb{N}}} a \rhd_{\mathbb{N}} *\right) \wedge \left(\forall_{a,b \in T_{\mathbb{N}}} . \ a \rhd_{\mathbb{N}} b \to Sa \rhd_{\mathbb{N}} Sb\right)$$

and the algebra $\mathbb{B} = \{\mathtt{tt}, \mathtt{ff}\}$ defines an acis by

$$T_{\mathbb{B}} := \{*, \mathtt{tt}, \mathtt{ff}\}$$

$$\forall_{a \in T_{\mathbb{B}}} . \ a \diamond_{\mathbb{B}} a \wedge a \diamond_{\mathbb{B}} *$$

$$\forall_{a \in T_{\mathbb{B}}} . \ a \rhd_{\mathbb{B}} a \wedge a \rhd_{\mathbb{B}} *$$

Arithmetical and Boolean Acises

Let $*$ be a (pre)atom meaning *least atomic information*.
The algebra $\mathbb{N} = \{0, S\}$ defines a *nonflat* acis by

$$T_{\mathbb{N}} := \{*, 0, S*, S0, S(S*), S(S0), \ldots\}$$

$$\left(\forall_{a \in T_{\mathbb{N}}} a \diamond_{\mathbb{N}} * \wedge * \diamond_{\mathbb{N}} a\right) \wedge \left(\forall_{a,b \in T_{\mathbb{N}}} . \ a \diamond_{\mathbb{N}} b \rightarrow Sa \diamond_{\mathbb{N}} Sb\right)$$

$$\left(\forall_{a \in T_{\mathbb{N}}} a \rhd_{\mathbb{N}} *\right) \wedge \left(\forall_{a,b \in T_{\mathbb{N}}} . \ a \rhd_{\mathbb{N}} b \rightarrow Sa \rhd_{\mathbb{N}} Sb\right)$$

and the algebra $\mathbb{B} = \{\mathtt{tt}, \mathtt{ff}\}$ defines an acis by

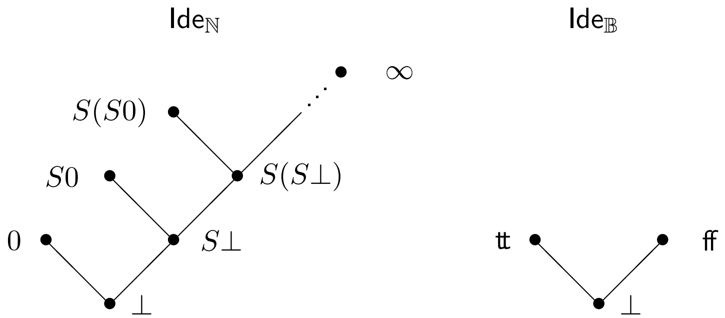
$$T_{\mathbb{B}} := \{*, \mathtt{tt}, \mathtt{ff}\}$$

$$\forall_{a \in T_{\mathbb{B}}} . \ a \diamond_{\mathbb{B}} a \wedge a \diamond_{\mathbb{B}} *$$

$$\forall_{a \in T_{\mathbb{B}}} . \ a \rhd_{\mathbb{B}} a \wedge a \rhd_{\mathbb{B}} *$$

Arithmetical and Boolean Acises (continued)

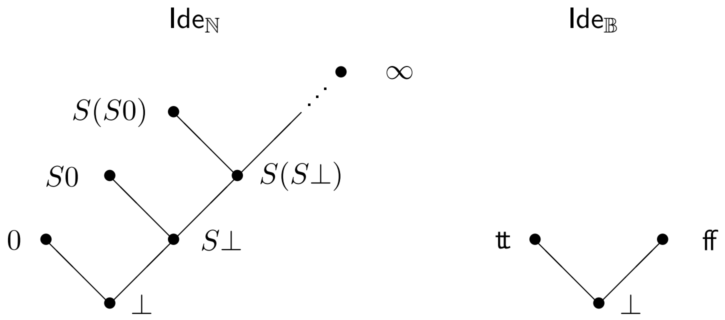
The corresponding ideals are structured like this:



- Lower ideals are included in (entailed by) higher ideals when a path connects them.
- The *total ideals* of $\mathbb{N}$, $G_{\mathbb{N}} = \{0, 1, 2, \ldots\}$, where $n := S^n 0$, can be used as *indices*.
- *Partial continuous functionals* are ideals of function spaces over $\mathbb{N}$ and $\mathbb{B}$.

Arithmetical and Boolean Acises (continued)

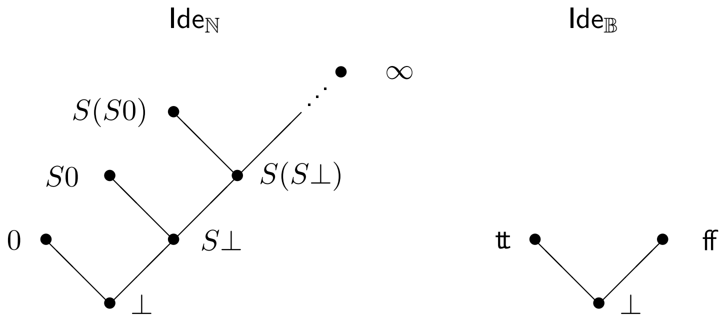
The corresponding ideals are structured like this:



- Lower ideals are included in (entailed by) higher ideals when a path connects them.
- The *total ideals* of $\mathbb{N}$, $G_{\mathbb{N}} = \{0, 1, 2, \ldots\}$, where $n := S^n 0$, can be used as *indices*.
- *Partial continuous functionals* are ideals of function spaces over $\mathbb{N}$ and $\mathbb{B}$.

Arithmetical and Boolean Acises (continued)

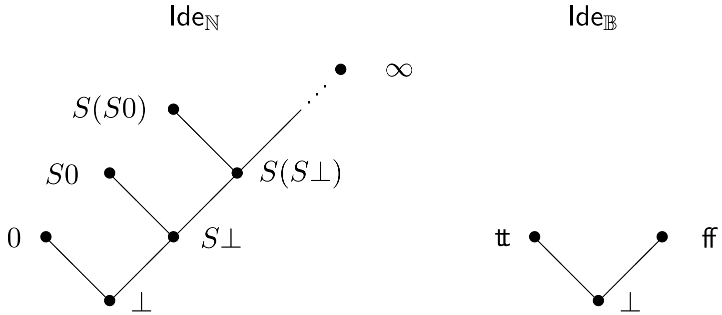
The corresponding ideals are structured like this:



- Lower ideals are included in (entailed by) higher ideals when a path connects them.
- The *total ideals* of $\mathbb{N}$, $G_{\mathbb{N}} = \{0, 1, 2, \ldots\}$, where $n := S^n 0$, can be used as *indices*.
- *Partial continuous functionals* are ideals of function spaces over $\mathbb{N}$ and $\mathbb{B}$.

Arithmetical and Boolean Acises (continued)

The corresponding ideals are structured like this:



- Lower ideals are included in (entailed by) higher ideals when a path connects them.
- The *total ideals* of $\mathbb{N}$, $G_{\mathbb{N}} = \{0, 1, 2, \ldots\}$, where $n := S^n 0$, can be used as *indices*.
- *Partial continuous functionals* are ideals of function spaces over $\mathbb{N}$ and $\mathbb{B}$.

Enter Syntax

Types, terms, and semantics

- Build arrow types $\alpha \to \beta$ based on $\mathbb{N}$ and $\mathbb{B}$.
- Use simply typed lambda terms, ie, typed variables, application and lambda abstraction.
- Interprete each *type* by the *set of ideals* of the corresponding acis; each lambda term will correspond to an ideal.

Computability

- Call an ideal of an acis *computable* if it is Σ^0_1-*definable* as a set of atoms.
- A simply typed lambda term corresponds to a computable ideal.
- What about the converse? *Is it always the case that a computable ideal can be defined in lambda terms?* [Plotkin 1977]

Enter Syntax

Types, terms, and semantics

- Build arrow types $\alpha \to \beta$ based on $\mathbb{N}$ and $\mathbb{B}$.
- Use simply typed lambda terms, ie, typed variables, application and lambda abstraction.
- Interprete each *type* by the *set of ideals* of the corresponding acis; each lambda term will correspond to an ideal.

Computability

- Call an ideal of an acis *computable* if it is Σ^0_1-*definable* as a set of atoms.
- A simply typed lambda term corresponds to a computable ideal.
- What about the converse? *Is it always the case that a computable ideal can be defined in lambda terms?* [Plotkin 1977]

Enter Syntax

Types, terms, and semantics

- Build arrow types $\alpha \to \beta$ based on $\mathbb{N}$ and $\mathbb{B}$.
- Use simply typed lambda terms, ie, typed variables, application and lambda abstraction.
- Interprete each *type* by the *set of ideals* of the corresponding acis; each lambda term will correspond to an ideal.

Computability

- Call an ideal of an acis *computable* if it is Σ^0_1-*definable* as a set of atoms.
- A simply typed lambda term corresponds to a computable ideal.
- What about the converse? *Is it always the case that a computable ideal can be defined in lambda terms?* [Plotkin 1977]

Enter Syntax

Types, terms, and semantics

- Build arrow types $\alpha \to \beta$ based on $\mathbb{N}$ and $\mathbb{B}$.
- Use simply typed lambda terms, ie, typed variables, application and lambda abstraction.
- Interprete each *type* by the *set of ideals* of the corresponding acis; each lambda term will correspond to an ideal.

Computability

- Call an ideal of an acis *computable* if it is Σ^0_1-*definable* as a set of atoms.
- A simply typed lambda term corresponds to a computable ideal.
- What about the converse? *Is it always the case that a computable ideal can be defined in lambda terms?* [Plotkin 1977]

Enter Syntax

Types, terms, and semantics

- Build arrow types $\alpha \to \beta$ based on $\mathbb{N}$ and $\mathbb{B}$.
- Use simply typed lambda terms, ie, typed variables, application and lambda abstraction.
- Interprete each *type* by the *set of ideals* of the corresponding acis; each lambda term will correspond to an ideal.

Computability

- Call an ideal of an acis *computable* if it is Σ^0_1-*definable* as a set of atoms.
- A simply typed lambda term corresponds to a computable ideal.
- What about the converse? *Is it always the case that a computable ideal can be defined in lambda terms?* [Plotkin 1977]

Enter Syntax

Types, terms, and semantics

- Build arrow types $\alpha \to \beta$ based on $\mathbb{N}$ and $\mathbb{B}$.
- Use simply typed lambda terms, ie, typed variables, application and lambda abstraction.
- Interprete each *type* by the *set of ideals* of the corresponding acis; each lambda term will correspond to an ideal.

Computability

- Call an ideal of an acis *computable* if it is Σ^0_1-*definable* as a set of atoms.
- A simply typed lambda term corresponds to a computable ideal.
- What about the converse? *Is it always the case that a computable ideal can be defined in lambda terms?* [Plotkin 1977]

Moving On to PCF

Introduce the following operators:

- *fixed points* Y : $(\alpha \to \alpha) \to \alpha$

$$\mathsf{Y}(u) := \bigcup_{n \in G_{\mathbb{N}}} u^n(\bot)$$

- *parallel conditional* pcond : $\mathbb{B} \to \mathbb{N} \to \mathbb{N} \to \mathbb{N}$

$$\mathsf{pcond}(p, u, v) := \begin{cases} u & p = \mathtt{tt} \\ v & p = \mathtt{ff} \\ u \cap v & p = \bot \end{cases}$$

- *parallel existential* exist : $(\mathbb{N} \to \mathbb{B}) \to \mathbb{B}$

$$\mathsf{exist}(u) := \begin{cases} \mathtt{ff} & \exists_{n \in G_{\mathbb{N}}} \,.\, u(S^n \bot) = \mathtt{ff} \wedge \forall_{k \le n}\, u(k) = \mathtt{ff} \\ \mathtt{tt} & \exists_{n \in G_{\mathbb{N}}}\, u(n) = \mathtt{tt} \\ \bot & \text{otherwise} \end{cases}$$

Moving On to PCF

Introduce the following operators:

- *fixed points* Y : $(\alpha \to \alpha) \to \alpha$

$$\mathsf{Y}(u) := \bigcup_{n \in G_{\mathbb{N}}} u^n(\bot)$$

- *parallel conditional* pcond : $\mathbb{B} \to \mathbb{N} \to \mathbb{N} \to \mathbb{N}$

$$\mathsf{pcond}(p, u, v) := \begin{cases} u & p = \mathtt{tt} \\ v & p = \mathtt{ff} \\ u \cap v & p = \bot \end{cases}$$

- *parallel existential* exist : $(\mathbb{N} \to \mathbb{B}) \to \mathbb{B}$

$$\mathsf{exist}(u) := \begin{cases} \mathtt{ff} & \exists_{n \in G_{\mathbb{N}}} .\ u(S^n \bot) = \mathtt{ff} \land \forall_{k \leq n}\, u(k) = \mathtt{ff} \\ \mathtt{tt} & \exists_{n \in G_{\mathbb{N}}}\, u(n) = \mathtt{tt} \\ \bot & \text{otherwise} \end{cases}$$

Moving On to PCF

Introduce the following operators:

- *fixed points* $\mathsf{Y} : (\alpha \to \alpha) \to \alpha$

$$\mathsf{Y}(u) := \bigcup_{n \in G_{\mathbb{N}}} u^n(\bot)$$

- *parallel conditional* pcond : $\mathbb{B} \to \mathbb{N} \to \mathbb{N} \to \mathbb{N}$

$$\mathsf{pcond}(p, u, v) := \begin{cases} u & p = \mathtt{tt} \\ v & p = \mathtt{ff} \\ u \cap v & p = \bot \end{cases}$$

- *parallel existential* exist : $(\mathbb{N} \to \mathbb{B}) \to \mathbb{B}$

$$\mathsf{exist}(u) := \begin{cases} \mathtt{ff} & \exists_{n \in G_{\mathbb{N}}} \,.\, u(S^n\bot) = \mathtt{ff} \wedge \forall_{k \leq n}\, u(k) = \mathtt{ff} \\ \mathtt{tt} & \exists_{n \in G_{\mathbb{N}}}\, u(n) = \mathtt{tt} \\ \bot & \text{otherwise} \end{cases}$$

Recursion in pcond and exist

Call an ideal $u \in \mathsf{Ide}_{\alpha \to \beta}$ *recursive in* pcond *and* exist if for all arguments $v \in \mathsf{Ide}_{\alpha}$ it can be defined by an equation

$$u(v) = M(v)$$

where M is a simply typed lambda term built up from variables, constructors, fixed points, parallel conditionals, and parallel existentials.

Examples

- *Sequential conditional operator*

$$\mathsf{cond}(p, u, v) := \mathsf{pcond}(p, \mathsf{pcond}(p, u, \bot), \mathsf{pcond}(p, \bot, v))$$

- *Disjunction operator*

$$\mathsf{or}(p, q) := \mathsf{pcond}(p, \mathsf{tt}, \mathsf{ff})$$

Recursion in pcond and exist

Call an ideal $u \in \mathsf{Ide}_{\alpha \to \beta}$ *recursive in* pcond *and* exist if for all arguments $v \in \mathsf{Ide}_\alpha$ it can be defined by an equation

$$u(v) = M(v)$$

where M is a simply typed lambda term built up from variables, constructors, fixed points, parallel conditionals, and parallel existentials.

Examples

- *Sequential conditional operator*

$$\mathsf{cond}(p, u, v) := \mathsf{pcond}(p, \mathsf{pcond}(p, u, \perp), \mathsf{pcond}(p, \perp, v))$$

- *Disjunction operator*

$$\mathsf{or}(p, q) := \mathsf{pcond}(p, \mathsf{tt}, \mathsf{ff})$$

Recursion in pcond and exist (continued)

For each type α assume an *enumeration* of Con_α that starts from the empty set and renders consistency, entailment, application, and union *primitive recursive*.

- *Extension enumeration operators* $\mathrm{en}_\alpha : \mathbb{N} \to \mathbb{N} \to \alpha$, with the property
$$\mathrm{en}_\alpha(m,n) = \overline{U_n}, \text{ when } U_n \rhd_\alpha U_m$$

- *Inconsistency operators* $\mathrm{incns}_\alpha : \alpha \to \mathbb{N} \to \mathbb{B}$, given by
$$\mathrm{incns}_\alpha(u,n) := \begin{cases} \mathrm{tt} & u \not\asymp_\alpha U_n \\ \mathrm{ff} & u \rhd_\alpha U_n \\ \bot & \text{otherwise} \end{cases}$$

These operators are simultaneously definable recursively in pcond and exist.

Recursion in pcond and exist (continued)

For each type α assume an *enumeration* of Con_α that starts from the empty set and renders consistency, entailment, application, and union *primitive recursive*.

- *Extension enumeration operators* $\mathsf{en}_\alpha : \mathbb{N} \to \mathbb{N} \to \alpha$, with the property

$$\mathsf{en}_\alpha(m, n) = \overline{U_n}, \text{ when } U_n \rhd_\alpha U_m$$

- *Inconsistency operators* $\mathsf{incns}_\alpha : \alpha \to \mathbb{N} \to \mathbb{B}$, given by

$$\mathsf{incns}_\alpha(u, n) := \begin{cases} \mathtt{tt} & u \not\asymp_\alpha U_n \\ \mathtt{ff} & u \rhd_\alpha U_n \\ \bot & \text{otherwise} \end{cases}$$

These operators are simultaneously definable recursively in pcond and exist.

Recursion in pcond and exist (continued)

For each type α assume an *enumeration* of Con_α that starts from the empty set and renders consistency, entailment, application, and union *primitive recursive*.

- *Extension enumeration operators* $\mathsf{en}_\alpha : \mathbb{N} \to \mathbb{N} \to \alpha$, with the property
$$\mathsf{en}_\alpha(m,n) = \overline{U_n}, \text{ when } U_n \rhd_\alpha U_m$$

- *Inconsistency operators* $\mathsf{incns}_\alpha : \alpha \to \mathbb{N} \to \mathbb{B}$, given by
$$\mathsf{incns}_\alpha(u,n) := \begin{cases} \mathtt{tt} & u \not\asymp_\alpha U_n \\ \mathtt{ff} & u \rhd_\alpha U_n \\ \bot & \text{otherwise} \end{cases}$$

These operators are simultaneously definable recursively in pcond and exist.

Definability Theorem

An ideal of type $\alpha \to \mathbb{N}$ over $\mathbb{N}$ and $\mathbb{B}$ is computable if and only if it is recursive in pcond and exist.

Proofsketch

Let $\Omega : \alpha \to \mathbb{N}$ be a computable ideal, represented as the primitive recursively enumerable set of atoms

$$\Omega = \left\{(U_{f(n)}, b_{g(n)})\right\}_{n \in G_{\mathbb{N}}} ,$$

where f, g are primitive recursive functions.

Definability Theorem

An ideal of type $\alpha \to \mathbb{N}$ over $\mathbb{N}$ and $\mathbb{B}$ is computable if and only if it is recursive in pcond and exist.

Proofsketch

Let $\Omega : \alpha \to \mathbb{N}$ be a computable ideal, represented as the primitive recursively enumerable set of atoms

$$\Omega = \left\{ (U_{f(n)}, b_{g(n)}) \right\}_{n \in G_{\mathbb{N}}} \ ,$$

where f, g are primitive recursive functions.

Definability Theorem (proofsketch continued)

For arbitrary $u \in \mathsf{Ide}_\alpha$ and $v \in \mathsf{Ide}_\mathbb{N}$, define the following tests:

- *argument inconsistency test*:

$$q_{u,f,n} := \mathsf{incns}_\alpha(u, f(n)) = \begin{cases} \mathtt{tt} & u \not\asymp_\alpha U_{f(n)} \\ \mathtt{ff} & u \rhd_\alpha U_{f(n)} \\ \bot & \text{otherwise} \end{cases}$$

- *value inconsistency test*:

$$q_{v,g,n} := \mathsf{incns}_\mathbb{N}(v, g(n)) = \begin{cases} \mathtt{tt} & v \not\asymp_\mathbb{N} b_{g(n)} \\ \mathtt{ff} & v \rhd_\mathbb{N} b_{g(n)} \\ \bot & \text{otherwise} \end{cases}$$

Definability Theorem (proofsketch continued)

For arbitrary $u \in \mathsf{Ide}_\alpha$ and $v \in \mathsf{Ide}_\mathbb{N}$, define the following tests:

- *argument inconsistency test*:

$$q_{u,f,n} := \mathsf{incns}_\alpha(u, f(n)) = \begin{cases} \mathtt{tt} & u \not\diamond_\alpha U_{f(n)} \\ \mathtt{ff} & u \rhd_\alpha U_{f(n)} \\ \bot & \text{otherwise} \end{cases}$$

- *value inconsistency test*:

$$q_{v,g,n} := \mathsf{incns}_\mathbb{N}(v, g(n)) = \begin{cases} \mathtt{tt} & v \not\diamond_\mathbb{N} b_{g(n)} \\ \mathtt{ff} & v \rhd_\mathbb{N} b_{g(n)} \\ \bot & \text{otherwise} \end{cases}$$

Definability Theorem (proofsketch continued)

For arbitrary $u \in \mathsf{Ide}_\alpha$ and $v \in \mathsf{Ide}_\mathbb{N}$, define the following tests:

- *argument inconsistency test*:

$$q_{u,f,n} := \mathsf{incns}_\alpha(u, f(n)) = \begin{cases} \mathtt{tt} & u \not\Diamond_\alpha U_{f(n)} \\ \mathtt{ff} & u \rhd_\alpha U_{f(n)} \\ \bot & \text{otherwise} \end{cases}$$

- *value inconsistency test*:

$$q_{v,g,n} := \mathsf{incns}_\mathbb{N}(v, g(n)) = \begin{cases} \mathtt{tt} & v \not\Diamond_\mathbb{N} b_{g(n)} \\ \mathtt{ff} & v \rhd_\mathbb{N} b_{g(n)} \\ \bot & \text{otherwise} \end{cases}$$

Definability Theorem (proofsketch continued)

Define a functional

$$\omega : \alpha \to (\mathbb{N} \to \mathbb{N}) \to G_{\mathbb{N}} \to \mathbb{N}$$

by

$$\begin{aligned} \omega_u(\psi)(n) \quad := \quad & \mathsf{pcond}\Big(q_{u,f,n}, \psi(n+1), \\ & \qquad \overline{b_{g(n)}} \cup \mathsf{pcond}\big(q_{\psi(n+1),g,n}, \bot, \psi(n+1)\big)\Big) \end{aligned}$$

Prove that

$$\bigvee_{n \in G_{\mathbb{N}}} . \ \Omega(u) \rhd_{\mathbb{N}} b_{g(n)} \leftrightarrow \mathsf{Y}(\omega_u)(0) \rhd_{\mathbb{N}} b_{g(n)}$$

$\square$

Definability Theorem (proofsketch continued)

Define a functional

$$\omega : \alpha \to (\mathbb{N} \to \mathbb{N}) \to G_{\mathbb{N}} \to \mathbb{N}$$

by

$$\begin{aligned}\omega_u(\psi)(n) \quad := \quad & \mathsf{pcond}\Big(q_{u,f,n}, \psi(n+1), \\ & \qquad \overline{b_{g(n)}} \cup \mathsf{pcond}\big(q_{\psi(n+1),g,n}, \bot, \psi(n+1)\big)\Big)\end{aligned}$$

Prove that

$$\forall_{n \in G_{\mathbb{N}}} . \; \Omega(u) \rhd_{\mathbb{N}} b_{g(n)} \leftrightarrow \mathsf{Y}(\omega_u)(0) \rhd_{\mathbb{N}} b_{g(n)}$$

$\square$

Definability Theorem (proofsketch continued)

Define a functional

$$\omega : \alpha \to (\mathbb{N} \to \mathbb{N}) \to G_{\mathbb{N}} \to \mathbb{N}$$

by

$$\begin{aligned}\omega_u(\psi)(n) \quad := \quad & \mathsf{pcond}\Big(q_{u,f,n}, \psi(n+1), \\ & \qquad\qquad \overline{b_{g(n)}} \cup \mathsf{pcond}\big(q_{\psi(n+1),g,n}, \bot, \psi(n+1)\big)\Big)\end{aligned}$$

Prove that

$$\underset{n \in G_{\mathbb{N}}}{\forall} . \ \Omega(u) \rhd_{\mathbb{N}} b_{g(n)} \leftrightarrow \mathsf{Y}(\omega_u)(0) \rhd_{\mathbb{N}} b_{g(n)}$$

$\square$

References

- Plotkin, Gordon: LCF considered as a programming language, *Theoretical Computer Science* **5**(3) (1997)
- Schwichtenberg, Helmut: Classifying recursive functions. In Griffor, E., ed.: *Handbook of computability theory*. Volume 140 of Studies in Logic and Foundations of Mathematics. North-Holland (1999)
- Schwichtenberg, Helmut: Recursion on the partial continuous functionals. In Dimitracopoulos, C., Newelski, L., Normann, D., Steel, J., eds.: *Logic Colloquium '05*. Volume 28 of Lecture Notes in Logic. Association for Symbolic Logic (2006)
- Scott, Dana: Domains for denotational semantics, in Nielsen, E. and Schmidt, E. M., eds.: *Automata, languages, and programming*. Volume 140 of Lecture Notes in Computer Science. Springer (1982)