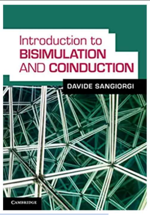
Recap



Induction and Co-induction

(co-)inductive definitions and the (co-)inductive method

1

Tarski-Knaster

Theorem *Let $f : L \to L$ be a monotonic function on a complete lattice. Then f has a greatest and a least fixed point expressed by:*

$$sup\{x \mid x \leq f(x)\} \qquad and \qquad inf\{x \mid f(x) \leq x\} \, .$$

From Tarski-Knaster Thm. Follow both:
- induction **proof principle**
- co-induction **proof principle**

2

Hence…

F(A) is the set of judgements that can be inferred in *one step* from the judjments in *A* by using the rules

A is

$closed$ if $F(A) \subseteq A$
$consistent$ if $A \subseteq F(A)$

The rules operator has both
a least fixed point and a greatest fixed point, which are
the smallest closed set and the **largest consistent set**:

$$lfp(F) = \bigcap\{A \mid F(A) \subseteq A\}; \quad \text{the least F -closed set.}$$
$$gfp(F) = \bigcup\{A \mid A \subseteq F(A)\}. \quad \text{the greatest F -consistent set}$$

3

Inductive and co-inductive interpretation of rules

$$lfp(F) = \bigcap\{A \mid F(A) \subseteq A\};$$
$$gfp(F) = \bigcup\{A \mid A \subseteq F(A)\}.$$

- If F(A) ⊆ A then $F_{ind} \subseteq A$ --- **Induction** proof principle
- If A ⊆ F(A) then $A \subseteq F_{coind}$ --- **Co-induction** proof principle

- **Induction principle**: to prove that all judgments in the inductive interpretation belong to a set A, show that A is F-closed.
- **Coinduction principle**: to prove that all judgments in a set A belong to the coinductive interpretation, show that A is F-consistent.

$$A \subseteq F(A) \Longrightarrow A \in \{B \mid B \subseteq F(B)\}$$
$$\Longrightarrow A \subseteq \bigcup\{B \mid B \subseteq F(B)\} = gfp(F).$$

4

Ex. 1 Consider the strings over an alphabet Σ

- **Consider the set S** coinductively defined by the following rules (where Σ is an alphabet)

$$\frac{}{\epsilon \in S} \qquad \frac{s \in S \quad \sigma \in \Sigma}{\sigma s \in S}$$

The largest set S such that $\epsilon \in S'$ and that if $\sigma s \in S$, then $s \in S$ (and $\sigma \in \Sigma$).

- Consider the relation on elements of S co-inductively defined by the rules

(where ≤ is th

$$\frac{}{\epsilon \leqslant \epsilon} \qquad \frac{\sigma_1 \leq \sigma_2 \qquad s_1 \leqslant s_2}{\sigma_1 s_1 \leqslant \sigma_2 s_2}$$

$$F(X) = \{(\epsilon, \epsilon)\} \cup \{(\sigma_1 s_1, \sigma_2 s_2) \mid \sigma_1 \leq \sigma_2 \wedge (s_1, s_2) \in X\}$$

EX. Prove that aaaaa··· ⩽ baaaa... (the two strings are infinite)

5

EX Lists (coinductive method)

$$\frac{}{\mathtt{nil} \in \mathcal{L}} \qquad \frac{\ell \in \mathcal{L} \qquad a \in A}{\mathtt{cons}(a, \ell) \in \mathcal{L}}$$

$$\mathcal{F}(S) \triangleq \{\mathtt{nil}\} \cup \{\mathtt{cons}(a, s) \ : \ a \in A, s \in S\}$$

Show that the infinite list $s1$ = **c b c b c …**
is in the set coinductively defined by the two rules above, assuming ***c, b*** ∈ *A*

1. Let us try *T* = {*s*1} and check that *T* is consistent with the rules, ie T ⊆ F(T)
2. We strengthen the hypothesis. Take s2 = **b c b c b …**
   Let us try *T* = {s_1,s_2}, and check that T ⊆ F(T)

Therefore, {s1,s2} ⊆ gfp F

for a given T,
if for all $x \in T$ there is a rule $(S, x) \in \mathcal{R}$ with $S \subseteq T$,
then $T \subseteq \mathtt{gfp}(\Phi_{\mathcal{R}})$

6

Induction and co-induction principle

A set R of rules on yields a monotone operator

$$\Phi_{\mathcal{R}}(T) = \{x \mid (T', x) \in \mathcal{R} \text{ for some } T' \subseteq T\}$$

$$\boxed{\begin{array}{l} \text{if } \Phi_{\mathcal{R}}(T) \subseteq T \text{ then } \mathtt{lfp}(\Phi_{\mathcal{R}}) \subseteq T, \\ \text{if } T \subseteq \Phi_{\mathcal{R}}(T) \text{ then } T \subseteq \mathtt{gfp}(\Phi_{\mathcal{R}}). \end{array}}$$

That is, each element of T is conclusion of a rule whose premises are satisfied in T.

7

More examples

8

Convergence (inductive)

Consider the definition of $\Downarrow_n$ in λ-calculus (convergence to a value):

$$\frac{}{\lambda x.\, e \Downarrow_n \lambda x.\, e} \qquad \frac{e_1 \Downarrow_n \lambda x.\, e_0 \qquad e_0\{e_2/x\} \Downarrow_n e'}{e_1(e_2) \Downarrow_n e'}$$

$$\begin{aligned} \Phi_{\Downarrow}(T) \stackrel{\text{def}}{=} & \{(e, e') \mid e = e' = \lambda x.e'', \text{ for some } e'' \in \Lambda \text{ and variable } x\} \\ & \bigcup \{(e, e') \mid \text{there are } e_1, e_2 \in \Lambda^0, e_0 \in \Lambda, \text{and a variable } x \text{ with} \\ & \qquad e = e_1\, e_2 \text{ and } (e_1, \lambda x.e_0) \in T \text{ and } (e_0\{e_2/x\}, e') \in T\}. \end{aligned}$$

for a given T,
if for all rules $(S, x) \in \mathcal{R}, S \subseteq T$ implies $x \in T$
then $\mathtt{lfp}(\Phi_{\mathcal{R}}) \subseteq T$.

9

Convergence (inductive)

Consider the definition of $\Downarrow_n$ in λ-calculus (convergence to a value):

$$\frac{}{\lambda x.\, e \Downarrow_n \lambda x.\, e} \qquad \frac{e_1 \Downarrow_n \lambda x.\, e_0 \qquad e_0\{e_2/x\} \Downarrow_n e'}{e_1(e_2) \Downarrow_n e'}$$

$\Downarrow$ is the *smallest* relation on (closed) λ-terms that is *closed* under the rules; i.e., the smallest relation $\mathcal{S} \subseteq \Lambda^0 \times \Lambda^0$ such that

- $\lambda x.e\ \mathcal{S}\ \lambda x.e$ for all abstractions,
- if $e_1\ \mathcal{S}\ \lambda x.e_0$ and $e_0\{e_2/x\}\ \mathcal{S}\ e'$ then also $e_1\ e_2\ \mathcal{S}\ e'$.

for a given T,
if for all rules $(S, x) \in \mathcal{R}, S \subseteq T$ implies $x \in T$
then $\mathtt{lfp}(\Phi_{\mathcal{R}}) \subseteq T$.

10

Divergence (co-inductive)

Consider the definition of $\Uparrow^n$ (divergence) in CbN λ-calculus :

$$\frac{e_1 \Uparrow^n}{e_1(e_2) \Uparrow^n} \qquad \frac{e_1 \Downarrow_n \lambda x.\, e_0 \qquad e_0\{e_2/x\} \Uparrow^n}{e_1(e_2) \Uparrow^n}$$

$$\begin{aligned} \Phi_{\Uparrow}(T) \stackrel{\text{def}}{=} & \{e_1\ e_2 \mid e_1 \in T, \} \\ & \bigcup \{e_1\ e_2 \mid \text{there is } e_0 \in \Lambda \text{ and a variable } x \text{ with} \\ & \qquad e_1 \Downarrow \lambda x.e_0 \text{ and } e_0\{e_2/x\} \in T\}. \end{aligned}$$

11

Divergence (co-inductive)

Consider the definition of $\Uparrow^n$ (divergence) in CbN λ-calculus :

$$\frac{e_1 \Uparrow^n}{e_1(e_2) \Uparrow^n} \qquad \frac{e_1 \Downarrow_n \lambda x.\, e_0 \qquad e_0\{e_2/x\} \Uparrow^n}{e_1(e_2) \Uparrow^n}$$

$\Uparrow^n$ is the *largest* predicate on λ-terms that is consistent with $T \subseteq F(T)$ these rules; i.e., the largest subset D of Λ s.t. if $e \in D$ then

- either $e = e_1(e_2)$ and $e_1 \in D$,
- or $e = e_1(e_2)$, $e_1 \Downarrow_n \lambda x.\, e_0$ and $e_0\{e_2/x\} \in D$.

Hence: to prove e is divergent it suffices to find $E \subseteq \Lambda$ that is consistent and with $e \in E$ (co-induction proof technique).

Ex: What is the smallest predicate consistent with the rules?
$T \subseteq F(T)$

12

EX Let $e_1 = \lambda x.xx$ Show that the term $e_1 e_1$ is divergent, using the coinduction proof method.

for a given T,
if for all $x \in T$ there is a rule $(S, x) \in \mathcal{R}$ with $S \subseteq T$,
then $T \subseteq \mathtt{gfp}(\Phi_{\mathcal{R}})$

Take the singleton set $T = \{ e_1 e_1 \}$.
We check that T is closed backward under the rules for $\Uparrow$.
In fact:

$$\frac{e_1 \Downarrow e_1 \qquad e_1\, e_1 \in T}{e_1\, e_1 \in T}$$

We deduce that $T \subseteq \Uparrow$

13

EX. use the coinduction proof method to show that if a closed term e does not converge (that is, there is no e' with $e \Downarrow e'$) then $e \Uparrow$.

Let T be the set of non-converging terms; we show that it is consistent with the rules defining $\Uparrow$.

Take a term $e \in T$.
This term cannot be an abstraction, otherwise $e \Downarrow e$ would hold.
Therefore $e = e1\ e2$.

- Case $e1 \in T$. We can match e against the first of the rules defining $\Uparrow$.
- *Case* $e1$ converges to $\lambda x.e0$. Consider thus $e0\{e2/x\}$.
If this term is in T, we match e against the second of the rules defining $\Uparrow$;
otherwise $e0\{e2/x\}$ converges and so also e converges (absurd).

$$\frac{}{\lambda x.e \Downarrow \lambda x.e} \qquad \frac{e_1 \Downarrow \lambda x.e_0 \qquad e_0\{e_2/x\} \Downarrow e'}{e_1\, e_2 \Downarrow e'}$$

$$\frac{e_1 \Uparrow}{e_1\, e_2 \Uparrow} \qquad \frac{e_1 \Downarrow \lambda x.e_0 \qquad e_0\{e_2/x\} \Uparrow}{e_1\, e_2 \Uparrow}$$

14

Recap

Reasoning on equivalence of programs

We will follow notes (available online) by :
- Luke Ong (Oxford)
- Roberto Amadio (IRIF)

•Lecture notes by L. Ong: Section 5 (and 6)
•Operational methods in semantics by R. Amadio:
Chapter 8 (weak reduction strategies) and 9 (simulation).

15

Equivalence on programs

A notion of *equivalence* among programs should be *natural* and *usable*.

- *Contextual equivalence is natural*
(but difficult to use)
- It can be characterized as a certain *simulation* which is easier to reason about.

16

Contextual equivalence

write $M \Downarrow$ and say that M converges if $\exists V \;\; M \Downarrow V$

We observe the *termination* of the term placed in a closing context, ie:
contexts C such that $C[M]$ and $C[N]$ are *closed terms*

$M \leq_C N$ if for all closing C ($C[M] \Downarrow$ implies $C[N] \Downarrow$)

Contextual equivalence *is derived by defining:*

$M \approx_C N$ if $M \leq_C N$ and $N \leq_C M$

17

Motivating example

$$one \stackrel{\text{def}}{=} \lambda x.\, \lambda y.\, x\, y$$
$$two \stackrel{\text{def}}{=} \lambda x.\, \lambda y.\, x\, (x\, y)$$
$$succ \stackrel{\text{def}}{=} \lambda n.\, \lambda x.\, \lambda y.\, x\, (n\, x\, y)$$

Is it the case that $succ\ one \Downarrow_v two$ holds?

18

CbN Simulation

We consider weak call-by-name λ *calculus*. We write $\Downarrow$ for $\Downarrow^n$

Definition 185 (simulation) *We say that a binary relation on* closed terms S *is a simulation if whenever* $(M, N) \in S$ *we have: (1) if* $M \Downarrow$ *then* $N \Downarrow$ *and (2) for all* P *closed* $(MP, NP) \in S$. *We shall also use the infix notation* $M\ S\ N$ *for* $(M, N) \in S$. *We define* $\leq_S$ *as the largest simulation.*

(recall that the set of binary relations is a complete lattice under set inclusion.)

$\leq_S$ is the largest fixed point of the following function on binary relations

$$f(S) = \{(M, N) \mid M \Downarrow \textit{ implies } N \Downarrow \ , \ \forall P \textit{ closed } (MP, NP) \in S\}$$

CO-INDUCTIVE DEFINITION

19

Ex. 2 Simulation

To prove that $M \leq_S N$ (M, N closed) it suffices to find a relation S which is a simulation and such that $M\ S\ N$.

EX

i. Show that $\leq_S$ is a preorder over Λ ie a reflexive and transitive binary relation

ii. Is the union of two simulations a simulation ?

iii. *If* $M \Downarrow V$ *and* $N \Downarrow V$, M, N *closed, then* $M =_S N$. Prove it.

20