Induction and Co-induction

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

Hence...

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 | F(A) \subseteq A\}; the least F-closed set.
- gfp(F) = \bigcup\{A | A \subseteq F(A)\}; the greatest F-consistent set.

Inductive and co-inductive interpretation of rules

- If F(A) \subseteq A then F(lfp A) \subseteq A \quad \text{--- Induction proof principle}
- If A \subseteq F(A) then A \subseteq F(gfp A) \quad \text{--- Co-induction proof principle}

Ex. 1 Consider the strings over an alphabet \(\Sigma\)

- Consider the set S coinductively defined by the following rules (where \(\Sigma\) is an alphabet)
  
  The largest set S such that
  
  - Consider the relation on elements of S co-inductively defined by the rules (where S is the
    smallest closed set and the largest consistent set):

  \[ F(X) = \{(x, i) \mid (\sigma_1 \leq \sigma_2, s_1 \leq s_2) \} \]

  \[ F(X) = \{(x, i) \mid (\sigma_1 \leq \sigma_2, s_1 \leq s_2) \} \]

  Therefore, \(\{s1, s2\} \subseteq \text{gfp } F\)

EX Lists (coinductive method)

\[ F(S) = \{\text{nil}\} \cup \{\text{cons}(\alpha, \gamma) : \alpha \in A, \gamma \in S\} \]

Show that the infinite list \(\text{nil} \circ \text{b} \circ \text{e} \circ \text{b} \circ \ldots\) is in the set coinductively defined by the two rules above, assuming \(\text{nil} \circ \text{b} \circ \text{e} \circ \text{b} \circ \ldots \)

1. Let us try \(T = \{\text{nil}\}\) and check that \(T\) is consistent with the rules, ie \(T \subseteq F(T)\)
2. We strengthen the hypothesis. Take \(\text{nil} \circ \text{b} \circ \text{e} \circ \text{b} \circ \ldots\) and \(\text{nil} \circ \text{b} \circ \text{e} \circ \text{b} \circ \ldots\)

Therefore, \(\{\text{nil}\} \subseteq \text{gfp } F\)

for a given T.

If for all \(s \in T\) there is a rule \((S, s) \in R\) with \(S \subseteq T\),

then \(T \subseteq \text{gfp } F\)
Induction and co-induction principle

A set $\mathcal{R}$ of rules on yields a monotone operator

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

If $\Phi_\mathcal{R}(T) \subseteq T$ then $\text{fp}(\Phi_\mathcal{R}) \subseteq T$,
if $T \subseteq \Phi_\mathcal{R}(T)$ then $T \subseteq \text{gfp}(\Phi_\mathcal{R})$.

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

More examples

Convergence (inductive)

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

$$\lambda x. \downarrow_n \lambda x. c \quad \text{if } e = \text{\textbf{true}} \rightarrow \lambda x. e \quad \text{if } n \neq 0$$

$$\Phi_\downarrow(T) \equiv \{ (e, e') \mid e = \lambda x. e' \text{ for some } e' \in \Lambda \text{ and variable } x \}$$

There are $e_1, e_2 \in \Lambda$, $e_0 \in \Lambda$, and a variable $x$ with $e = e_1 \beta e_2$ and $(e_1, \lambda x. e_0) \in T$ and $(e_0 \beta e', e') \in T$.

For a given $T$,
if for all rules $(S, x) \in R, S \subseteq T$ implies $x \in T$
then $\downarrow_n(\Phi_\downarrow) \subseteq T$.

Convergence (inductive)

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

$$\lambda x. \downarrow_n \lambda x. c \quad \text{if } e = \text{\textbf{true}} \rightarrow \lambda x. e \quad \text{if } n \neq 0$$

$\downarrow_n$ is the smallest relation on (closed) $\lambda$-terms that is closed under the rules, i.e.,
the smallest relation $S \subseteq \Lambda \times \Lambda$ such that:

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

For a given $T$,
if for all rules $(S, x) \in R, S \subseteq T$ implies $x \in T$
then $\downarrow_n(\Phi_\downarrow) \subseteq T$.

Divergence (co-inductive)

Consider the definition of $\uparrow^*$ (divergence) in CBN $\lambda$-calculus:

$$\lambda e. \uparrow^* \quad \text{if } e \in \text{\textbf{false}} \rightarrow \lambda e. e \quad \text{if } e \in \text{\textbf{true}}$$

$$\Phi_\uparrow(T) \equiv \{ e_1, e_2 \mid e_1 \in T \}$$

There is $e_0 \in \Lambda$ and a variable $x$ with $e_1 \uparrow \lambda x. e_0$ and $e_0 \beta e' \in T$.

$\uparrow^*$ is the largest predicate on $\lambda$-terms that is consistent with $T \subseteq F(T)$
these rules; i.e., the largest subset $D$ of $\Lambda$ s.t. if $e \in D$ then
- either $e = e_1 e_2$ and $e_1 \in D$,
- or $e = e_1 e_2$, $e_1 \uparrow \lambda x. e_0$ and $e_0 \beta e'$ $\in D$.

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

Ex: What is the smallest predicate consistent with the rules?

$T \subseteq F(T)$
Let $e_1 = \lambda x.x x$

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 $(S, x) \in R$ with $S \subseteq T$, then $T \subseteq \uparrow (\Phi_T)$

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

In fact:

We deduce that $T \subseteq \uparrow$

---

**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).

---

**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, i.e. contexts $C$ such that $C[M]$ and $C[N]$ are closed terms.

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

Contextual equivalence is derived by defining:

$M \equiv C N \text{ if } M \preceq C N \text{ and } N \preceq C M$

---

**Motivating example**

\[
\begin{align*}
\text{one} & \overset{\text{def}}{=} \lambda x. \lambda y. x y \\
\text{two} & \overset{\text{def}}{=} \lambda x. \lambda y. x (x y) \\
\text{suc} & \overset{\text{def}}{=} \lambda n. \lambda x. \lambda y. x (n x y)
\end{align*}
\]

Is it the case that suc one $\Downarrow$, two holds?
CbN Simulation

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

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

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

\textbf{CO-INDUCTIVE DEFINITION}

\hline
\textbf{Ex. 2 Simulation}

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

\begin{enumerate}
  \item \textit{i.} Show that $\leq_S$ is a preorder over $\Lambda$ is a reflexive and transitive binary relation
  \item \textit{ii.} Is the union of two simulations a simulation?
  \item \textit{iii.} If $M \Downarrow V$ and $N \Downarrow V$, $M, N$ closed, then $M \leq_S N$. Prove it.
\end{enumerate}