Domain Theory

Part 6: Non-determinism

Dr. Liam O'Connor

based on material from

Glynn Winskel, Gordon Plotkin, Kai Engelhardt, Dexter Kozen, Andrew Myers, Carl Gunter

March 27, 2024

1 Introduction

This lecture concerns semantics for non-deterministic computations. When modelling programming languages we sometimes require a way to be imprecise about some aspects of a program — usually, this takes the form of non-determinism. Non-determinism can be hard for beginners to grasp, but it typically has to be employed when modelling real programs. For example, suppose we had a greeting program that differed depending on the physical location of the computer[1]:

```
if $currentLocation() = \text{Korea}$ then
    say "안녕하세요"
else
    say "Hello!"
fi
```

If we wanted to mathematically model the behaviour of this program, it would be frightfully inconvenient to include the geography of Earth and the computer's physical location in our model. That's where non-determinism comes in. If we abstract away from the geographical details, and instead regard the program as choosing between the two options based on some unspecified criteria, then we can get away with modelling less, at the cost of some detail:

```
if ??? then
    say "안녕하세요"
else
    say "Hello!"
fi
```

Such underspecified conditionals are usually called non-deterministic choice, where our conditional **if** ??? **then** P **else** Q **fi** is written simply as $P + Q$.

While we tend to view our physical machines as deterministic automata, the state upon which each decision is deterministically made includes a number of external things which are tedious to model mathematically. We can also use non-determinism to ignore details that we don't care about for our particular domain – a common example is memory allocation, where it is very convenient (for some programs) to think of memory as infinite, and allocation as an operation that can potentially fail, without specifying exactly when and how this failure can occur. This is normally modelled as a choice between successful allocation and a failure state.

[1] Further internationalisation is left as an exercise

In a deterministic setting, such as the imperative language $\mathcal{C}$ we saw in Lecture 1, we usually model semantics of a program as a function – given an initial state, there will be one final state determined entirely by the initial state (or $\perp$ in the case of non-termination). But, with non-determinism, each use of the choice operator potentially doubles the number of final states[2]. So, with non-determinism in our language, there are multiple possible final states for a given initial state.

Example from Lecture 1 (A Simple Imperative Language)

Recall the simple imperative language $\mathcal{C}$ from Lecture 1:

$$\begin{array}{lcl}
\mathcal{E} & ::= & n \mid x \mid \mathcal{E}_1 + \mathcal{E}_2 \mid \mathcal{E}_1 - \mathcal{E}_2 \\
\mathcal{B} & ::= & \mathsf{false} \mid \mathsf{true} \mid \neg\mathcal{B} \mid \mathcal{E}_1 = \mathcal{E}_2 \\
\mathcal{C} & ::= & \mathsf{skip} \mid x := \mathcal{E} \mid \mathcal{C}_1 ; \mathcal{C}_2 \mid \mathsf{if}\ \mathcal{B}\ \mathsf{then}\ \mathcal{C}_1\ \mathsf{else}\ \mathcal{C}_2 \mid \mathsf{while}\ \mathcal{B}\ \mathsf{do}\ \mathcal{C}\ \mathsf{od} \\
x & \in & \mathcal{V} \quad \textit{(variables)} \\
n & \in & \mathbb{Z}
\end{array}$$

In Lecture 1, we gave a semantics to imperative programs (without while) in terms of functions on states $\Sigma \to \Sigma$. Later, we showed that, by using a cpo such as $\Sigma \rightharpoonup \Sigma_\perp$ for our semantic domain, we have least fixed points and can therefore assign semantics to while loops. Since non-deterministic programs can have more than one outcome, a natural choice of semantic domain would be $\Sigma \rightharpoonup \mathcal{P}(\Sigma_\perp) \setminus \emptyset$, i.e., functions from an initial state to a non-empty set of outcomes.

2 Generalising Semantics

Definition

A monad on the category $\mathbf{C}$ consists of an endofunctor $\mathcal{M} : \mathbf{C} \to \mathbf{C}$ and two operations[a] for all $\mathbf{C}$-objects X: $\mu_X : \mathcal{M}(\mathcal{M}(X)) \to \mathcal{M}(X)$ and $\eta_X : X \to \mathcal{M}(X)$, such that the following diagrams commute:

$$\begin{tikzcd}
\mathcal{M}(\mathcal{M}(\mathcal{M}(X))) \arrow[r, "\mathcal{M}(\mu_X)"] \arrow[d, "\mu_{\mathcal{M}(X)}"'] & \mathcal{M}(\mathcal{M}(X)) \arrow[d, "\mu_X"] & \mathcal{M}(X) \arrow[r, "\eta_{\mathcal{M}(X)}"] \arrow[d, "\mathcal{M}(\eta_X)"'] \arrow[dr, equal] & \mathcal{M}(\mathcal{M}(X)) \arrow[d, "\mu_X"] \\
\mathcal{M}(\mathcal{M}(X)) \arrow[r, "\mu_X"] & \mathcal{M}(X) & \mathcal{M}(\mathcal{M}(X)) \arrow[r, "\mu_X"] & \mathcal{M}(X)
\end{tikzcd}$$

In equational form, these laws are

1. $\mu_X \circ \mathcal{M}(\mu_X) = \mu_X \circ \mu_{\mathcal{M}(X)}$
2. $\mu_X \circ \mathcal{M}(\eta_X) = \mu_X \circ \eta_{\mathcal{M}(X)} = \mathsf{id}_{\mathcal{M}(X)}$

In computer science, η_X is sometimes called *unit, pure,* or *return*, and μ_X is usually called *join*.

It is often convenient to instead work in the Kleisli category of the monad $\mathcal{M}$. For a monad $(\mathcal{M}, \mu, \eta)$ over the category $\mathbf{C}$, the Kleisli category has the same objects as $\mathbf{C}$, but the morphisms $A \xrightarrow{h} B$ are those morphisms h of $\mathbf{C}$ that have the form $A \xrightarrow{h} \mathcal{M}(B)$.

[2] This is why deterministically simulating a non-deterministic program is exponential complexity in the worst-case.

Composition is defined as follows, for morphisms $A \xrightarrow{f} \mathcal{M}(B)$ and $B \xrightarrow{g} \mathcal{M}(C)$:

$$g \circ_K f = \mu_C \circ \mathcal{M}(g) \circ f$$

Identity morphisms for this category are just η.

[a] These operations are *natural transformations* – morphisms in the category of functors, but we omit this detail here.

Example

Our familiar flat domain lifting operation $(\cdot)_\perp$ forms a monad, where the endofunctor $\mathcal{M}(X) = X_\perp$, the operation $\mu_X : (X_\perp)_\perp \to X_\perp$ collapses the two $\perp$ values, and $\eta_X : X \to X_\perp$ is simply $\eta_X(x) = x$.

Instead of committing to a particular semantic domain, let's define our semantics generically in terms of a monad $(\mathcal{M}, \mu, \eta)$:

$$\begin{aligned}
&\mathit{ite} : (\Sigma \to \mathbb{B}) \times (\Sigma \to \mathcal{M}(\Sigma)) \times (\Sigma \to \mathcal{M}(\Sigma)) \to (\Sigma \to \mathcal{M}(\Sigma)) \\
&\mathit{ite}(b, t, e)(\sigma) = \begin{cases} t(\sigma) & \text{if } b(\sigma) = \mathrm{T} \\ e(\sigma) & \text{otherwise} \end{cases}
\end{aligned}$$

$$\begin{aligned}
&[\![\cdot]\!]_{\mathcal{C}} : \mathcal{C} \to \Sigma \to \mathcal{M}(\Sigma) \\
&[\![\mathsf{skip}]\!]_{\mathcal{C}} &&= \eta_\Sigma \\
&[\![x \coloneqq e]\!]_{\mathcal{C}} &&= \lambda\sigma.\, \sigma\,(x \mapsto [\![e]\!]_{\mathcal{E}}) \\
&[\![c_1; c_2]\!]_{\mathcal{C}} &&= [\![c_2]\!]_{\mathcal{C}} \circ_K [\![c_1]\!]_{\mathcal{C}} \\
&[\![\mathsf{if}\ b\ \mathsf{then}\ c_1\ \mathsf{else}\ c_2]\!]_{\mathcal{C}} &&= \mathit{ite}([\![b]\!]_{\mathcal{B}}, [\![c_1]\!]_{\mathcal{C}}, [\![c_2]\!]_{\mathcal{C}}) \\
&[\![\mathsf{while}\ b\ \mathsf{do}\ c\ \mathsf{od}]\!]_{\mathcal{C}} &&= \mathsf{fix}(\lambda f.\, \mathit{ite}([\![b]\!]_{\mathcal{B}}, f \circ_K [\![c]\!]_{\mathcal{C}}, \eta_\Sigma))
\end{aligned}$$

We can provide η and μ operations for our proposed monad for non-determinism $\mathcal{M}(X) = \mathcal{P}(X_\perp) \setminus \emptyset$:

$$\begin{aligned}
&\eta : X \to \mathcal{P}(X_\perp) \setminus \emptyset \\
&\eta(x) = \{x\} \\
&\mu : \mathcal{P}((\mathcal{P}(X_\perp) \setminus \emptyset)_\perp) \to \mathcal{P}(X_\perp) \setminus \emptyset \\
&\mu(S) = (\textstyle\bigcup_{X \neq \perp \in S} X) \cup (S \cap \{\perp\})
\end{aligned}$$

And, with this semantics, we can easily add a denotation for non-deterministic choice, in terms of union:

$$[\![c_1 + c_2]\!]_{\mathcal{C}} = \lambda\sigma. [\![c_1]\!]_{\mathcal{C}}\sigma \cup [\![c_2]\!]_{\mathcal{C}}\sigma$$

Unfortunately this is only the beginning of our troubles. While we have given a valuation function above, its definition uses **fix**, and therefore our domain $\Sigma \to \mathcal{M}(\Sigma)$ must be a cpo, and all our operations continuous. It is not clear how to order elements of $\mathcal{P}(X_\perp) \setminus \emptyset$.

3 Powerdomains

If we consider our domain $\mathcal{P}(X_\perp) \setminus \emptyset$, a natural choice of ordering would be subset inclusion $\subseteq$. However, as we lack $\emptyset$, there is no least element for this ordering. Furthermore, simply using $\subseteq$ as our ordering would mean that the deterministic, terminating program that always returns a single state $\{\sigma\}$ would be considered *less* informative than the program that may diverge $\{\perp, \sigma\}$. This would make most of our operations non-continuous.

Instead, we shall examine three separate orderings on this set, each of which carries a different view of non-determinism. More generally, we wish to come up with a construction analogous to the powerset operator $\mathcal{P}$, but for our semantic domains (i.e. Scott domains). It is convenient at this point to switch from cpos to Scott domains, as the algebraicity of Scott domains is useful here. As Scott domains are ω-algebraic, and all algebraic domains D are isomorphic to the ideal completion of their compact elements $D \simeq \mathsf{Id}(K(D))$, we lose nothing by working only with compact elements.

Reminder: Ideals

An ideal is a downwards-closed directed set. The ideal completion of a set A, written sometimes as $\mathsf{Id}(A)$, is the set of all ideal subsets of A, i.e.:

$$\mathsf{Id}(A) = \{X \subseteq A \mid X \text{ is ideal}\}$$

3.1 The Hoare Powerdomain

We write $\mathcal{P}^*_f(S)$ to denote finite non-empty subsets of S. Let $X, Y \in \mathcal{P}^*_f(K(A))$ be finite, non-empty sets of compact elements of a Scott domain A. The Hoare powerdomain construction is based on the following ordering:

$$X \preceq_H Y \quad \text{iff} \quad (\forall x \in X.\, \exists y \in Y.\, x \sqsubseteq_A y)$$

The intuition behind this is that "Anything X can do, Y can do better". This ordering has some desirable properties: The least element is $\{\bot\}$, as we might expect. It also satisfies the equation $X \preceq_H X \cup Y$, which intuitively means that a non-deterministic choice is considered to have *more* information than any of its components. When specialised to the case where A is a flat domain like $\Sigma_\bot$, it simplifies to:

$$X \preceq_H Y \quad \text{iff} \quad X \setminus \{\bot\} \subseteq Y$$

This is, however, not a proper order, but a preorder, as it fails antisymmetry. Even in the flat domain case, $\{x, \bot\} \preceq_H \{x\}$ and $\{x\} \preceq_H \{x, \bot\}$ but they are not equal. However, any preorder *does* induce an equivalence relation, where $X \approx_H Y$ iff $X \preceq_H Y$ and $Y \preceq_H X$. Furthermore, if we take the *ideal completion* of $\mathcal{P}^*_f(K(A))$ with respect to the preorder $\preceq_H$ (i.e., the set of all $\preceq_H$-downwards-closed directed subsets of $\mathcal{P}^*_f(K(A))$), we can order them by inclusion to obtain an algebraic domain with compact elements which correspond to equivalence classes of elements under $\approx_H$. We have a lub operation for these ideals and thus our cpo is directed:

$$A \sqcup_H B = \{X \cup Y \mid X \in A \wedge Y \in B\}$$

Example

Let us find the Hoare powerdomain of $\mathbb{N}_\bot$. Note that $\mathbb{N}_\bot$ is a flat domain so all elements are compact, i.e. $K(\mathbb{N}_\bot) = \mathbb{N}_\bot$. Let us find examine the $\preceq_H$-ideal subsets of $\mathcal{P}^*_f(\mathbb{N}_\bot)$. If v and u both contain $\bot$, then $v \preceq_H u$ iff $v \subseteq u$. As ideal subsets are downwards closed, we can identify an ideal $I \in \mathsf{Id}(\mathcal{P}^*_f(\mathbb{N}_\bot))$ with its union $\bigcup I$. The ideals of $\mathcal{P}^*_f(\mathbb{N}_\bot)$ are then isomorphic to domain $\mathcal{P}(\mathbb{N})$ of all subsets of $\mathbb{N}$ under subset inclusion.

Consider the following three programs:

1. $x := 1$
2. while true do skip
3. while true do skip od $+\ x := 1$

Given a state σ, let $\sigma' = \sigma(x \mapsto 1)$. Then, the possible results of these programs are $\{\sigma'\}$, $\{\bot\}$, and $\{\sigma', \bot\}$ respectively. Because our construction identifies $\approx_H$-equivalent sets, this semantics will consider programs 1 and 3 to be the same. That is, the non-determinism here is *angelic*. The bad outcome ($\bot$) is only said to happen if it is *inevitable*.

3.2 The Smyth Powerdomain

As before, let $X, Y \in \mathcal{P}^*_f(K(A))$ be finite, non-empty sets of compact elements of a Scott domain A. The Smyth powerdomain construction is based on the following preorder:

$$X \preceq_S Y \quad \text{iff} \quad (\forall y \in Y.\, \exists x \in X.\, x \sqsubseteq_A y)$$

The intuition here is "Everything Y can do, X can do worse". It satisfies the equation $X \cup Y \preceq_S X$, which intuitively means that a non-deterministic choice is considered to have *less* information than any of its components. When applied to a flat domain, it simplifies to:

$$X \preceq_S Y \quad \text{iff} \quad \bot \in X \vee Y \subseteq X$$

Once again, this fails to be a partial order, but we can use the same ideal completion trick to induce an algebraic domain where compact elements correspond to equivalence classes of elements under $\approx_S$ (where $X \approx_S Y$ is, as with $\approx_H$, just defined as $X \preceq_S Y \wedge Y \preceq_S X$).

Example

Let us find the Smyth powerdomain of $\mathbb{N}_\bot$. Firstly, note that, as ideals are downwards-closed, any $\preceq_S$-ideal subsets of $\mathcal{P}^*_f(\mathbb{N}_\bot)$ will contain *all* finite subsets of $\mathbb{N}_\bot$ that contain $\bot$. Let us call those sets *trivial*. So, a set in $\mathcal{P}^*_f(\mathbb{N}_\bot)$ is *non-trivial* if it does not contain $\bot$ and an ideal subset of $\mathcal{P}^*_f(\mathbb{N}_\bot)$ is *non-trivial* if it contains a non-trivial set. Observe that for non-trivial sets X and Y, $X \preceq_S Y$ iff $X \supseteq Y$. Thus, we can identify an ideal $I \in \mathsf{Id}(\mathcal{P}^*_f(\mathbb{N}_\bot))$ with the down-closure of the *intersection* of its non-trivial elements. The smaller this set is, the larger the ideal I. Hence, and confusingly, the non-trivial ideals in the powerdomain (ordered by subset inclusion) correspond to finite subsets of $\mathbb{N}$ ordered by *superset* inclusion. As all *trivial* sets are identified, we just need to throw the unique trivial ideal, and we can see that the Smyth powerdomain of $\mathbb{N}$ is isomorphic to the domain of sets $\{\mathbb{N}\} \cup \mathcal{P}^*_f(\mathbb{N})$ ordered by subset inclusion.

Note that all sets that contain with $\bot$ are considered equivalent by $\approx_S$. Thus, this models *demonic* non-determinism: Of the three programs in the previous section, this semantics considers 2 and 3 to be the same. Thus, the *possibility* of divergence $\bot$ is no different from *inevitable* divergence.

3.3 Plotkin Powerdomain

As before, let $X, Y \in \mathcal{P}^*_f(K(A))$ be finite, non-empty sets of compact elements of a Scott domain A. The Plotkin powerdomain construction is based on the following preorder, simply combining the previous two:

$$X \preceq_P Y \quad \text{iff} \quad X \preceq_H Y \wedge X \preceq_S Y$$

This ordering is also called the *Egli-Milner* ordering. On a flat domain, it simplifies to:

$$A \sqsubseteq B \quad \text{iff} \quad A = B \vee (\bot \in A \wedge (A \setminus \{\bot\}) \subseteq B)$$

Note that this *is* antisymmetric in the case of a flat domain, and the lub of a chain of sets S can be found by taking the union of all elements of the chain if all elements contain $\bot$. If not, then the element without $\bot$ will be the upper bound.

The failures of antisymmetry can only be observed in domains of height higher than one. Consider a set $\{x, y\}$ containing two elements. According to the induced equivalence from this preorder $\approx_P$, this set would be considered equal to the set that contains those two elements *plus* any elements that lie between them on the information ordering:

$$\{x, y\} \approx_P \{z \mid x \sqsubseteq z \sqsubseteq y\}$$

The Plotkin powerdomain is sometimes called the *convex powerdomain*, due to the similarity of this to the geometric definition of convexity. Thus, we use the ideal-completion trick here to arrive at a definition of powerdomain that distinguishes between all three programs outlined in Section 3.1.

Aside

There is a beautiful algebraic characterisation of Plotkin Powerdomains in:
Matthew Hennessy and Gordon D. Plotkin. 1979. *Full Abstraction for a Simple Parallel Programming Language*. In Mathematical Foundations of Computer Science 1979, Proceedings, Olomouc, Czechoslovakia, September 3-7, 1979 (LNCS, Vol. 74), Springer, 108–120

Glossary

choice An operator, $P + Q$ that represents non-deterministic execution of P or of Q, but it is not known which . 1–5

Hoare powerdomain The power domain of A induced by the ideal completion of the finite non-empty subsets of $K(A)$ ordered by the Hoare preorder:

$$X \preceq_H Y \quad \text{iff} \quad (\forall x \in X.\, \exists y \in Y.\, x \sqsubseteq_A y)$$

. 4, 6

Kleisli category A category induced from a monad $\mathcal{M}$ on a category $\mathbf{C}$, whose objects are the same as $\mathbf{C}$ but whose morphisms $X \xrightarrow{m} Y$ are those of the form $X \xrightarrow{m} \mathcal{M}(Y)$ in $\mathbf{C}$. 2

monad A monoid in a category of endofunctors. Specifically, a *monad* on the category $\mathbf{C}$ consists of an endofunctor $\mathcal{M} : \mathbf{C} \to \mathbf{C}$ and two natural transformations $\mu_X : \mathcal{M}(\mathcal{M}(X)) \to \mathcal{M}(X)$ and $\eta_X : X \to \mathcal{M}(X)$, such that $\mu_X \circ \mathcal{M}(\mu_X) = \mu_X \circ \mu_{\mathcal{M}(X)}$ and $\mu_X \circ \mathcal{M}(\eta_X) = \mu_X \circ \eta_{\mathcal{M}(X)} = \mathrm{id}_{\mathcal{M}(X)}$. 2, 3, 6

non-deterministic A program is *non-deterministic* if it can (as specified) have more than one final state for a given initial state . 1–6

Plotkin powerdomain The power domain of A induced by the ideal completion of the finite non-empty subsets of $K(A)$ ordered by the Egli-Milner preorder:

$$X \preceq_P Y \quad \text{iff} \quad X \preceq_H Y \wedge X \preceq_S Y$$

where $\preceq_S$ and $\preceq_H$ are the preorders from the Smyth powerdomain and Hoare powerdomain respectively . 5, 6

preorder A relation that is reflexive and transitive, but not necessarily antisymmetric. 4–6

Smyth powerdomain The power domain of A induced by the ideal completion of the finite non-empty subsets of $K(A)$ ordered by the Smyth preorder:

$$X \preceq_S Y \quad \text{iff} \quad (\forall y \in Y.\, \exists x \in X.\, x \sqsubseteq_A y)$$

. 5, 6