Introduction

Main support: "Formalized, Effective Domain Theory in Coq" by Robert Dockins (left for Galois)

Objective: Define a Coq library providing the tools to define denotational semantics to languages
Hence formalizes domain theory:
-> In Coq
-> In an effective way

Effective: -> "Internal" constructions defined are effective
-> "External" the metalogic used is purely constructive

Consequence: * Can use Coq's native support for recursive functions
* No axiom added

This course: Using this paper as a pretext to discuss notions related to Type Theory, Coq and Constructiveness

Toward CPOS!

I) Preorders, Partial Orders and setoids

Domain theory is usually built upon Partial Orders: Reflexive, Transitive, Anti-symmetric.
Dockins takes a different route: to work with Preorders: Reflexive, Transitive.

① The "How"

Let $(A, \sqsubseteq)$ be a preorder.
One can derive a setoid, i.e. a set equipped with an equivalence relation, from the preorder:
$x \equiv y$ iff $x \sqsubseteq y$ and $y \sqsubseteq x$
Now by identifying elements up to $\equiv$, we actually recover the structure of a partial order:
$x \sqsubset y$ iff $x \sqsubseteq y$ and $\neg\, x \equiv y$
Dockins works along all the development with such preorders.
More specifically, effective preorders: A is an enumerable set, and we have a decision procedure for its order.

② The "why"

Dockins is not very explicit about his motivations, but here's an attempt.
In mathematics, the notion of quotient is extremely common.
Given a set, and an equivalence relation, you define equivalences classes.
Ex1: $x \equiv y$ if $x \bmod 3 = y \bmod 3$ $\quad \mathbb{Z}_3 \cong \mathbb{Z}/_\equiv$
Ex2: $(n,m) \equiv (p,q)$ if $nq = mp$ $\quad \mathbb{Q} \cong (\mathbb{Z} \times \mathbb{Z})/_\equiv$

Note that you actually define a new set, $|\mathbb{Z}_3| = 3$

How to transfer this in type theory?

-> Define a notion of quotient type

"Q is a quotient type of base type T if:

$$\left.\begin{array}{l}\pi : T \to Q \\ repr : Q \to T\end{array}\right\} \forall x,\ \pi(repr\ x) = x$$ "

Not natively supported in Coq, see "Pragmatic Quotient Types in Coq" by Cyril Cohen

-> Pretend you work over the quotient by carrying the setoid with you.

This has decent support in Coq. What do we mean?

Rewriting in Coq

* Leibniz equality

Native equality in Coq ≡ Leibniz.

↳ Extremely strong: convenient, but restrictive

"Inductive eq {A: Type} (x : A) : A → ℙ := eq_refl : x = x."

$x = y$ iff any property on A which holds true of x also holds true of y.

↳ i.e. we can replace x by y in any context, rewriting is easy!

* Beyond Leibniz

But what if that's not what we want?

Sol 1: Wait for HIT

Sol 2: Setoids

↳ See setoids.v

↳ Then preorders.v

Finite Sets

Next comes the notion of finite set.
Let us explore that this innocent notion is non trivial in constructive mathematics

"Constructively Finite?" by Coquand and Spiwack

For Set Theory, ZFC for instance, finite sets is a non-ambiguous notion, admitting numerous equivalent characterization. Even worse, a set is simply either finite or infinite.

We explore here the fact that they are not all finite for the same reason!
To this end, we visit four notions of finiteness.

I) Enumerated Sets

"A set A is enumerated if there is a list of all its elements"
$A \in \mathfrak{F}$

* $\forall A \in \mathfrak{F}$, $A = \emptyset$ is decidable

* $\forall A \in \mathfrak{F}$, $P \in (A \to \mathbb{B}ool)$, $\{x \mid P x\} \in \mathfrak{F}$ (map, then filter)
  However, this is not true for $P \in (A \to Prop)$
  Proof: Let $Unit = \{0\}$ the one element set, enumerated by $[0]$
  $\forall A$ proposition, if $\{x \in Unit \mid P x\}$ where $P = \lambda x. A \in Unit \to Prop$.
  Then if the set was enumerated, checking if it is inhabited would decide A.
  This notion is closed under computable subsets, but not general subsets

* $\forall A \in \mathfrak{F}$, $f : A \to B$ ~~then~~ surjective, then $B \in \mathfrak{F}$
  Proof: ℓ enumeration of A then $(f\ \ell)$ enumeration of B

* $\forall A, B \in \mathfrak{F}$, $A \times B \in \mathfrak{F}$ and $A + B \in \mathfrak{F}$
  ℓ, v enums of A, B.
  $\{(s, t) \mid s \in \ell, t \in v\}$ enum of $A \times B$
  $(\ell ++ v)$ enum of $A + B$

II) Bounded Sets

"A set A is bounded if $\exists N \in \mathbb{N}$, $\forall l \in \text{List } A$ s.t. $|l| > N$, l has duplicates."
$A \in \mathfrak{B}$

Rq: Gives a bound on the size but not the size
Does not give a way to choose an element in A.

* $\forall A \in \mathfrak{B}$ $f : B \to A$ injective, then $B \in \mathfrak{B}$.
  i.e. $\mathfrak{B}$ is closed under arbitrary subsets.
  Proof: $l \in \text{List} B$. If $f\, l$ has duplicates, then so does l by injectivity.
  Hence the bound on A is a bound on B.

* $\mathfrak{F} \subsetneq \mathfrak{B}$
  $\subseteq$: $A \in \mathfrak{F}$, l an enum. $\forall v \in \text{List} A$ s.t. $|v| > |l|$, then v has duplicates by the pigeon hole principle.
  $\mathfrak{F} \neq \mathfrak{B}$: $\mathfrak{B}$ is stable for arbitrary subsets not $\mathfrak{F}$.

* $\forall A \in \mathfrak{B}$, $f \in A \to B$ surjective, then $B \in \mathfrak{B}$

* $\mathfrak{B}$ stable under disjoint sum
  Proof: N, M bounds on A, B.
  $l \in \text{List } A+B$ with $|l| > N+M$
  $l_A = [a \mid a \in A \wedge a \in l]$, $l_B = \ldots$
  $|l_A| + |l_B| > N+M$ hence $|l_A| > N$ or $|l_B| > M$
  hence duplicate.

* $\mathfrak{B}$ stable under cartesian product
  Proof: Say $p =_A q$ for $(p, q) \in A \times B$ if $fst\ p = fst\ q$
  Define A-duplicates.
  F | Let $L \in \text{List } (A \times B \times \text{List } \mathbb{N})$, define the following operation if $|L| > N$
  Find $x(s_1, t_1, l_1)$ and $y(s_2, t_2, l_2)$ A-duplicates
  $L' = L \setminus \{x, y\} \cup \{(s_1, t_1, l_1 ++ l_2)\}$
  Now let $l \in \text{List } A \times B$ s.t. $|l| > NM$
  Let $L_0 = [(a, b, [i]) \mid (a, b) \in A \times B ; \text{index}]$
  Here are two invariants of L_0 and F:
  * fold $(\lambda\ acc\ (a, b, l) \Rightarrow acc ++ l)\ [\,]\ L$ is a permutation of $[1, NM]$
  * if $(s, t, l) \in L$, $i \in l$, then $l[i] =_A (s, t)$

  From L_0, we can iterate until $|L| \leq N$
  By extracting the lists of positions, we have $\leq N$ lists whose total size is $> N \times M$
  Let l' be one such sublist of size $> M$
  It has a pair of B-duplicates, and only A-equal elements
  Hence the conclusion.

III) Noetherian sets (After Emmy Noether)

Intuition: "Any ascending chain of enumerated subsets has two consecutive equal terms"

Definition: Define $\mathcal{N}_l$ inductively, for $l \in$ List A:

- If l has duplicates, $A \in \mathcal{N}_l$
- If, $\forall a \in A$, $A \in \mathcal{N}_{a::l}$, then $A \in \mathcal{N}_l$

$A \in \mathcal{N}$ iff $A \in \mathcal{N}_{[]}$ (equivalently, $\forall l$, $A \in \mathcal{N}_l$)

* $\mathcal{B} \notin \mathcal{N}$
* Same properties of stability

IV) Streamless sets

Intuition: "A set is finite if one cannot inject $\mathbb{N}$ into it"

Definition: "A streamless if any stream $s: \mathbb{N} \to A$ admits two indices i, j s.t. $s\,i = s\,j$", $i < j$

* $\mathcal{N} \subseteq \mathcal{S}$
* $\mathcal{N} \neq \mathcal{S}$: conjectured, not proved
* $\forall A, B \in \mathcal{S}$, $A + B \in \mathcal{S}$

Proof:

Lemma: $A + \text{Unit} \in \mathcal{S}$

Proof: if $s\,0 = s\,1 = \text{inr}\,()$: ok

otherwise, one of them is some inl a_0.

let $s' = \lambda n.$ match $s\,n$ with $|\text{inl}\,a \Rightarrow a\;|\;\text{inr}\,() \Rightarrow a_0$ end

s' stream of a, hence $\exists i, j$, $s'\,i = s'\,j$

let $s'' = s'|_{j+1}$, $\exists\, i' < j'$ s.t. $s''\,i' = s''\,j'$

We now have 4 indices $i < j < k < l$ with $k = j+1+i'$ and $l = j+1+j'$

s.t. $s'_i = s'_j$ and $s'_k = s'_l$.

Transporting to s, we can conclude no matter what $\square$

Proof:

$s: \mathbb{N} \to A+B$, define $s^A: \mathbb{N} \to A + \text{Unit}$ and $s^B: \mathbb{N} \to B + \text{Unit}$.

Let $i < j$ s.t. $s^A\,i = s^A\,j$, define $s': \mathbb{N} \to A+B$

hd $s' = s\,i$

tl s': keep acting corecursively on $s^A|_{j+1}$

s' is s.t. for all k, $s'\,k$ corresponds to Lk and Rk in s s.t.:

- $Rk < L(k+1)$
- $s'_k \in B \Rightarrow s_{Lk} = s'_k$
- $s'_k \in A \Rightarrow s_{Lk} = s_{Rk}$

Then consider s'^B. It has a collision $i < j$, $s'^B\,i = s'^B\,j$. Either both are in B or both are $()$.

In both case, we can conclude. $\square$

* Finally, the cartesian product is an open problem!