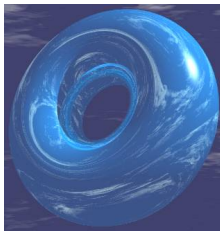


NON-WELLFOUNDED SETS

THIS IS **DRAFT**: CHECK BACK TONIGHT FOR MORE

Larry Moss
Indiana University, Bloomington

ESLLI 2012, Opole



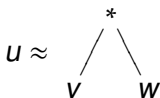
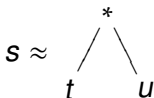
We want to move from streams to a more complicated example, **infinite trees**.

Some of the points that we make will be closely related to what we have seen for streams, and some will raise new issues.

Here is a class of objects which we shall call **trees**.

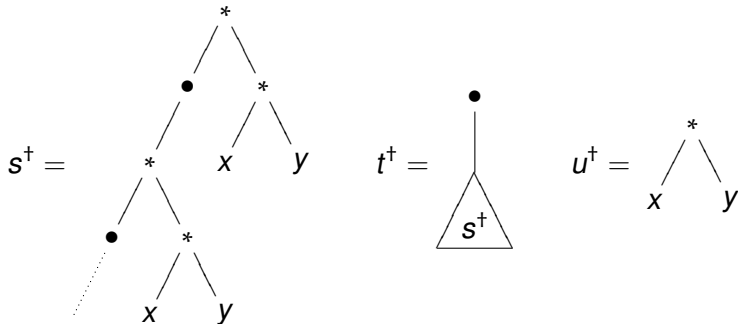
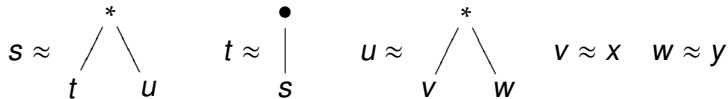
- 1 The variables x and y alone are trees.
- 2 If t is a tree, then adding a single node labeled \bullet as a new root with t as its only subtree gives a tree.
- 3 If s and t are trees, then adding a single node labeled $*$ as a new root, with s as the left subtree and t as the right, is a tree.
- 4 Trees may go on forever.

Trees may be specified by **tree systems**.
Here is one such system:



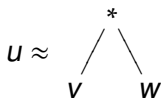
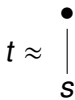
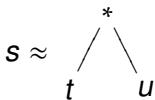
$$v \approx x \quad w \approx y$$

Trees may be specified by **tree systems**.
Here is one such system:



- ① The symbols x and y alone are trees.
- ② If t is a tree, then $\langle \bullet, t \rangle$ is a tree.
- ③ If s and t are trees, then $\langle *, s, t \rangle$ is a tree.
- ④ Trees may be “infinitely deep”.

TWO NOTATIONS FOR THE SAME THING



$$v \approx x \quad w \approx y$$

$$s \approx \langle *, t, u \rangle$$

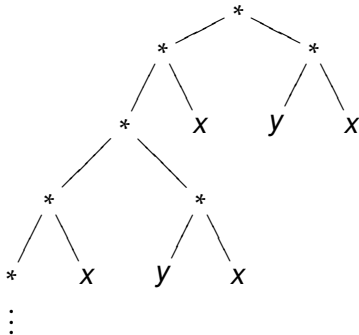
$$t \approx \langle \bullet, s \rangle$$

$$u \approx \langle *, v, w \rangle$$

$$v \approx x$$

$$w \approx y$$

Here is an infinite tree



Find a system of tree equations that has this tree as the solution to one of the variables. Find your system in terms of pictures, and then translate it to a system in terms of tuples.

With streams we had an **unraveled form**,
a function from Nat to Nat .

What is the unraveled form in the case of trees?

Let Tr be the set of trees that we have been discussing.
Then our definition in terms of Tr would have

$$Tr = \{x, y\} \cup (\{\bullet\} \times Tr) \cup (\{*\} \times Tr \times Tr).$$

The standard modeling in set theory gives us a problem:

one can prove in ZF set theory that Tr only contains the finite trees;

but this runs afoul of our pictures and intuition.

The standard way out is to change the equals sign $=$ above to something else.

But again, [how can we understand what is going on here?](#)

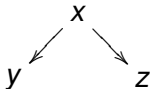
Let us turn from streams and trees to **sets**.

Before presenting some analogs to what we have just seen, at **pictures of sets**.

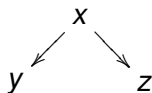
To make the discussion concrete, consider the set

$$x = \{\emptyset, \{\{\emptyset\}, \emptyset\}\}$$

Let us call this set x . We want to draw a picture of this set, so we start with a point which we think of as x itself. Since x has two elements, we draw add two children:



$$x = \{\emptyset, \{\{\emptyset\}, \emptyset\}\}$$

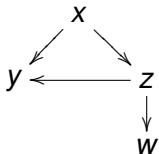


We take y to be \emptyset and z to be $\{\{\emptyset\}, \emptyset\}$.

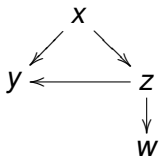
We do not add any children of y because it is empty.

But we want to add two children to z ,
one for $w = \{\emptyset\}$ and one for \emptyset .

So we have



$$x = \{\emptyset, \{\{\emptyset\}, \emptyset\}\}$$



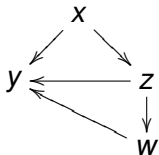
x for $\{\emptyset, \{\{\emptyset\}, \emptyset\}\}$

y for \emptyset

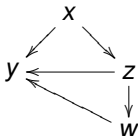
z for $\{\{\emptyset\}, \emptyset\}$

w for $\{\emptyset\}$

We conclude by putting an arrow from w to y, since $\emptyset \in \{\emptyset\}$.



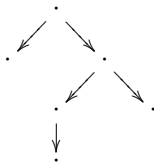
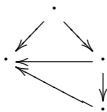
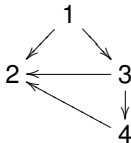
WHAT ARE THE NODES?



Now we want to forget the identity of the nodes.

We could either trade in the four sets that we used for numbers (to mention just one way), or else finesse the issue entirely.

We would get one of the pictures below:



A **graph** is a pair (G, \rightarrow) , where \rightarrow is a relation on G (a set of ordered pairs from G).

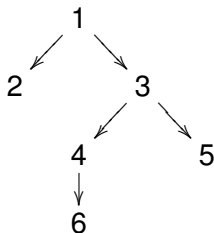
The idea is that we want to think of a graphs as **notations for sets**, just as systems of equations were notation for streams.

This is explained by the concept of a **decoration**:

A decoration d of a graph G is a function whose domain is G and with the property that

$$d(g) = \{d(h) : g \rightarrow h\}.$$

For example, let us introduce names for the nodes in the tree-like graph and then find its decoration:



Since 6 has no children, $d(6)$ must be \emptyset .
Similarly, $d(5)$ and $d(2)$ are also \emptyset .

$$\begin{aligned}
 d(4) &= \{d(6)\} &= \{\emptyset\} \\
 d(3) &= \{d(4), d(5)\} &= \{\{\emptyset\}, \emptyset\} \\
 d(1) &= \{d(2), d(3)\} &= \{\emptyset, \{\{\emptyset\}, \emptyset\}\}
 \end{aligned}$$

Note this is the set x with which we started.

Why is this true?

However, things get more interesting with an example like the loop graph



Let d be a decoration of this graph.

Then we would have $d(x) = \{d(x)\}$.

So writing Ω for $d(x)$, we have

$$\Omega = \{\Omega\}$$

This set Ω is the most conspicuous example of **object circularity**:

a set that is a member of itself.

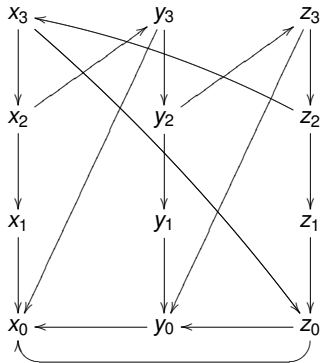
(Indeed, Ω is its own only member.)

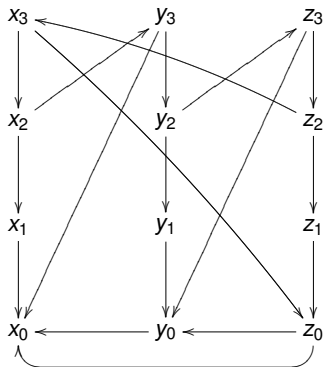
Finally, we want to consider an example that harks back to

$$x \approx \langle 0, y \rangle$$

$$y \approx \langle 1, z \rangle$$

$$z \approx \langle 2, x \rangle$$

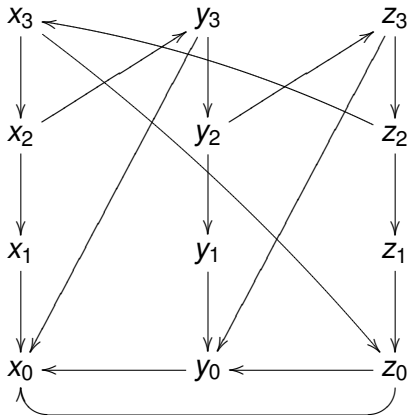




Let us try to understand what a decoration d of this graph would be. Remember from set theory that the standard rendering of the first few natural numbers is by

$$0 = \emptyset, \quad 1 = \{\emptyset\}, \quad 2 = \{0, 1\} = \{\emptyset, \{\emptyset\}\}$$

and also that the standard definition of the ordered pair $\langle x, y \rangle$ is as $\{\{x\}, \{x, y\}\}$.



Since x_0 has no children, $d(x_0)$ must be \emptyset .

Then it follows that $d(y_0) = \{d(x_0)\} = \{\emptyset\} = 1$.

And now $d(z_0) = \{d(x_0), d(y_0)\} = \{0, 1\} = 2$.

Furthermore, $d(z_1) = \{2\}$. It follows now that

$$d(x_1) = \{0\}, \quad d(y_1) = \{1\}, \quad d(z_1) = \{2\}.$$

And then

$$\begin{aligned} d(x_2) &= \{d(y_3), d(x_1)\} = \{\{0, d(y_2)\}, \{0\}\} = \langle 0, d(y_2) \rangle \\ d(y_2) &= \{d(z_3), d(y_1)\} = \{\{1, d(z_2)\}, \{1\}\} = \langle 1, d(z_2) \rangle \\ d(z_2) &= \{d(x_3), d(z_1)\} = \{\{2, d(x_2)\}, \{2\}\} = \langle 2, d(x_2) \rangle \end{aligned}$$

The upshot is that we can go back to our original stream system and then “solve it” by putting down our big graph and decorating it.

The solution would be

$$x^\dagger = d(x_2) \quad y^\dagger = d(y_2) \quad z^\dagger = d(z_2).$$

How do we know that our big graph actually has a decoration?

A **hyperset** or **non-wellfounded set** is a set that is obtained by decorating an arbitrary graph.

By such a system we mean a pair (X, e) , where X is a set which we think of as “variables” (any set will do), and $e : X \rightarrow \mathcal{P}X$.

That is, the value of e on each variable is again a set of variables.

Set systems and related concepts correspond to ones for graphs in the following way:

the graph (G, \rightarrow)	the system of set equations (X, e)
the nodes of G	the set X of variables
the relation \rightarrow on the nodes	the function $e : X \rightarrow \mathcal{P}X$
the children of x in G	the set $e(x) \in \mathcal{P}X$
a decoration d of the graph	a solution e^\dagger of the system

Every graph corresponds to a system of set equations,
and vice-versa.

An example, corresponding to the big picture from earlier:

$$X = \{x_0, y_0, z_0, x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3\}$$

$$e(x_0) = \emptyset \quad e(x_1) = \{x_0\} \quad e(x_2) = \{x_1, y_3\} \quad e(x_3) = \{z_0, x_2\}$$

$$e(y_0) = \{x_0\} \quad e(y_1) = \{y_0\} \quad e(y_2) = \{y_1, z_3\} \quad e(y_3) = \{x_0, y_2\}$$

$$e(z_0) = \{x_0, y_0\} \quad e(z_1) = \{z_0\} \quad e(z_2) = \{z_1, x_3\} \quad e(z_3) = \{y_0, z_2\}$$

So the way to go from the picture to the function is that
for all variables v , $e(v)$ is the set of children of v .

Our preferred notation elides e :

$$x_0 \approx \emptyset \quad x_1 \approx \{x_0\} \quad x_2 \approx \{x_1, y_3\} \quad x_3 \approx \{z_0, x_2\}$$

$$y_0 \approx \{x_0\} \quad y_1 \approx \{y_0\} \quad y_2 \approx \{y_1, z_3\} \quad y_3 \approx \{x_0, y_2\}$$

$$z_0 \approx \{x_0, y_0\} \quad z_1 \approx \{z_0\} \quad z_2 \approx \{z_1, x_3\} \quad z_3 \approx \{y_0, z_2\}$$

stream system

$$\begin{aligned}x &\approx \langle 0, y \rangle \\y &\approx \langle 1, z \rangle \\z &\approx \langle 2, x \rangle\end{aligned}$$

tree system

$$\begin{aligned}s &\approx \langle *, t, u \rangle \\t &\approx \langle \bullet, s \rangle \\u &\approx \langle *, v, w \rangle \\v &\approx x \\w &\approx y\end{aligned}$$

set system

$$\begin{aligned}a &\approx \{b, c\} \\b &\approx \emptyset \\c &\approx \{c, a\}\end{aligned}$$

stream system

$$e : X \rightarrow \text{Nat} \times X$$

tree system

$$e : X \rightarrow \{x, y\} \cup (\{\bullet\} \times X) \cup (\{*\} \times X \times X)$$

set system

$$e : X \rightarrow \mathcal{P}X$$

stream system

$$e : X \rightarrow \text{Nat} \times X$$

solutions are maps from X to $\text{in Nat}^\infty = \text{Nat} \times \text{Nat}^\infty$

tree system

$$e : X \rightarrow \{x, y\} \cup (\{\bullet\} \times X) \cup (\{*\} \times X \times X)$$

solutions map to $Tr = \{x, y\} \cup (\{\bullet\} \times Tr) \cup (\{*\} \times Tr \times Tr)$

set system

$$e : X \rightarrow \mathcal{P}X$$

solutions map to $V = \mathcal{P}V$

All of the equalities are problematic.

stream system

$$e : X \rightarrow \text{Nat} \times X$$

solutions e^\dagger in $\text{Nat}^\infty = \text{Nat} \times \text{Nat}^\infty$

$$e^\dagger(x) = \langle \text{fst}(e(x)), e^\dagger(\text{snd}(e(x))) \rangle$$

tree system

$$e : X \rightarrow \{x, y\} \cup (\{\bullet\} \times X) \cup (\{*\} \times X \times X)$$

solutions e^\dagger in $\text{Tr} = \{x, y\} \cup (\{\bullet\} \times \text{Tr}) \cup (\{*\} \times \text{Tr} \times \text{Tr})$

set system

$$e : X \rightarrow \mathcal{P}X$$

solutions (decorations) in $V = \mathcal{P}V$

$$e^\dagger(x) = \{e^\dagger(y) : y \in e(x)\}$$

MORE DETAILS ON THE STREAM SETTING, FROM OUR EARLIER EXAMPLE

$$\begin{array}{l|l} x \approx \langle 0, y \rangle & x^\dagger = \langle 0, y^\dagger \rangle \\ y \approx \langle 1, z \rangle & y^\dagger = \langle 1, z^\dagger \rangle \\ z \approx \langle 2, x \rangle & z^\dagger = \langle 2, x^\dagger \rangle \end{array}$$

Re-packaged in terms of $X = \{x, y, z\}$ and $e : X \rightarrow \text{Nat} \times X$ as

$$\begin{array}{l|l} e(x) = \langle 0, y \rangle & e^\dagger(x) = \langle 0, e^\dagger(y) \rangle \\ e(y) = \langle 1, z \rangle & e^\dagger(y) = \langle 1, e^\dagger(z) \rangle \\ e(z) = \langle 2, x \rangle & e^\dagger(z) = \langle 2, e^\dagger(x) \rangle \end{array}$$

$$e^\dagger : X \rightarrow \text{Nat}^\infty = \text{Nat} \times \text{Nat}^\infty$$
$$e^\dagger(x) = \langle \text{fst}(e(x)), e^\dagger(\text{snd}(e(x))) \rangle$$

AN EXERCISE ON THE BASIC NOTION OF TREE SYSTEMS

For a stream system $e : X \rightarrow \text{Nat} \times X$, we have

$$e^\dagger(x) = \langle \text{fst}(e(x)), e^\dagger(\text{snd}(e(x))) \rangle$$

What is the analog for a tree system?

MORE ON THE SET SETTING, FROM A NEW EXAMPLE

$$x \approx \{y, z\}$$

$$y \approx \emptyset$$

$$z \approx \{x\}$$

$$e(x) = \{y, z\}$$

$$e(y) = \emptyset$$

$$e(z) = \{x\}$$

$$d(x) = \{d(y), d(z)\}$$

$$d(y) = \emptyset$$

$$d(z) = \{d(x)\}$$

$$e^+(x) = \{e^+(y), e^+(z)\}$$

$$e^+(y) = \emptyset$$

$$e^+(z) = \{e^+(x)\}$$

MORE ON SET SYSTEMS, FROM A NEW EXAMPLE

The point here is that it makes sense to think of a set system as a function of the form $e : X \rightarrow \mathcal{P}X$ and as its solution as one of the form $e^+ : X \rightarrow V$ (that is, as a function defined on X with some sets or others as its values).

However, two more points deserve comment.

First, unlike the case of streams and trees, it is not clear that every system “should” have a unique solution.

Second, we wrote $V = \mathcal{P}V$ for the solution space.

This was mainly in analogy to what we saw with streams and trees.

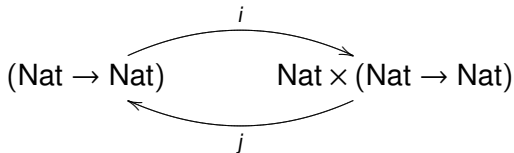
But **what is the analogy?**

For streams we also may use the unraveled form, working via isomorphism:

$$\begin{array}{ccc}
 & i & \\
 & \curvearrowright & \\
 (\text{Nat} \rightarrow \text{Nat}) & & \text{Nat} \times (\text{Nat} \rightarrow \text{Nat}) \\
 & \curvearrowleft & \\
 & j &
 \end{array}$$

Here $i(f) = \langle f(0), \lambda m.f(m+1) \rangle$
 and $j(n, f) = \lambda m.\text{if } m = 0, \text{ then } n, \text{ otherwise } f(n-1).$

For streams we also may use the unraveled form, working via isomorphism:



And then the solution condition for a stream system would be

$$e^\dagger(x) = j(\text{fst}(e(x)), e^\dagger(\text{snd}(e(x))))$$

Can isomorphism help with the other examples?

At this point, we have seen three examples of

- ① Objects presented as the solution of systems of various kinds.
- ② The solution spaces themselves giving rise to problematic forms of circularity.

The goal of the course is to make sense of this phenomenon in a general, and mathematically insightful manner.

At this point, we have seen three examples of

- ① Objects presented as the solution of systems of various kinds.
- ② The solution spaces themselves giving rise to problematic forms of circularity.

The goal of the course is to make sense of this phenomenon in a general, and mathematically insightful manner.

The way it will work:

system of some sort	coalgebra of some functor F
solution space for those systems	final coalgebra for F

I want to broaden the scope of our discussion by presenting examples/questions which at first glance might be unrelated to what we have seen.

- ① A different kind of example of a circularly-defined object (function)
- ② A case where the solution space was originally an open question.

The Cantor set $c \subseteq [0, 1]$ has several equivalent definitions/characterizations:

- ① Take the unit interval $[0, 1]$, then remove the open middle third $(\frac{1}{3}, \frac{2}{3})$, leaving two disconnected pieces. For each of those, remove the open middle third. Keep going for infinitely many steps. Then c is what remains “at the end”.
- ② c is the set of numbers possessing a ternary (base 3) decimal expansion with no 1’s.
- ③ c is the unique non-empty compact subset of the unit interval $[0, 1]$ such that

$$c = \frac{1}{3}c \cup \left(\frac{2}{3} + \frac{1}{3}c \right),$$

where $\frac{1}{3}c$ denotes the set $\{\frac{1}{3}x \mid x \in c\}$, and the second set is interpreted similarly, by also adding $\frac{2}{3}$ to each point.

The equation **defines c in terms of itself**.

Let C be the set of non-empty compact subsets of $[0, 1]$.

Let $g : C \times C \rightarrow C$ be

$$g(x, y) = \frac{1}{3}x \cup \left(\frac{2}{3} + \frac{1}{3}y\right)$$

Then, it turns out that there is a unique function $f : C \rightarrow C$ so that

$$f(x) = g(f(x), x).$$

Compare this with a function defined by [recursion on Nat](#), such as

$$\mathit{factorial}(n) = \mathit{ifzero}(n, \mathit{one}, \mathit{factorial}(\mathit{pred}(n)) * n)$$

Here we define the factorial function $n \mapsto n!$ in terms of ifzero , one , pred and $*$.

Every set of equations like

$$\begin{aligned}x_0 &\approx \frac{1}{2}x_3 + \frac{1}{2} \\x_1 &\approx \frac{1}{2}x_2 + \frac{1}{2} \\x_2 &\approx \frac{1}{2}x_{15} \\x_3 &\approx \frac{1}{2}x_{19} + \frac{1}{2} \\x_4 &\approx \frac{1}{2}x_2 \\&\vdots \qquad \qquad \qquad \vdots\end{aligned}$$

has a unique solution in $[0, 1]$.

My presentation here omits details of importance in the economics/game theory literature.

If M is a measurable space, then $\Delta(M)$ is the set of **probability measures on M** , made into a measurable space in a canonical way (next lecture).

Can we solve

$$M \cong \Delta([0, 1] \times M)$$

and similar equations? Can we get the “biggest” solutions?

Let S be a measurable space intended as a model of “the state of the world.”

A **two player beliefs space over S** is a tuple (M, σ, N, τ) , where M and N are measurable spaces, and

$$\sigma : M \rightarrow \Delta(S \times N)$$

$$\tau : N \rightarrow \Delta(S \times M)$$

Can we find a single beliefs space $(M^*, \sigma^*, N^*, \tau^*)$ such that every belief space mapped uniquely into it via a map that respects “beliefs”?

The question about $M \cong \Delta([0, 1] \times M)$ contains most of the mathematical details but is missing some features coming from having two players.

SOMETHING TO LEAVE YOU WITH