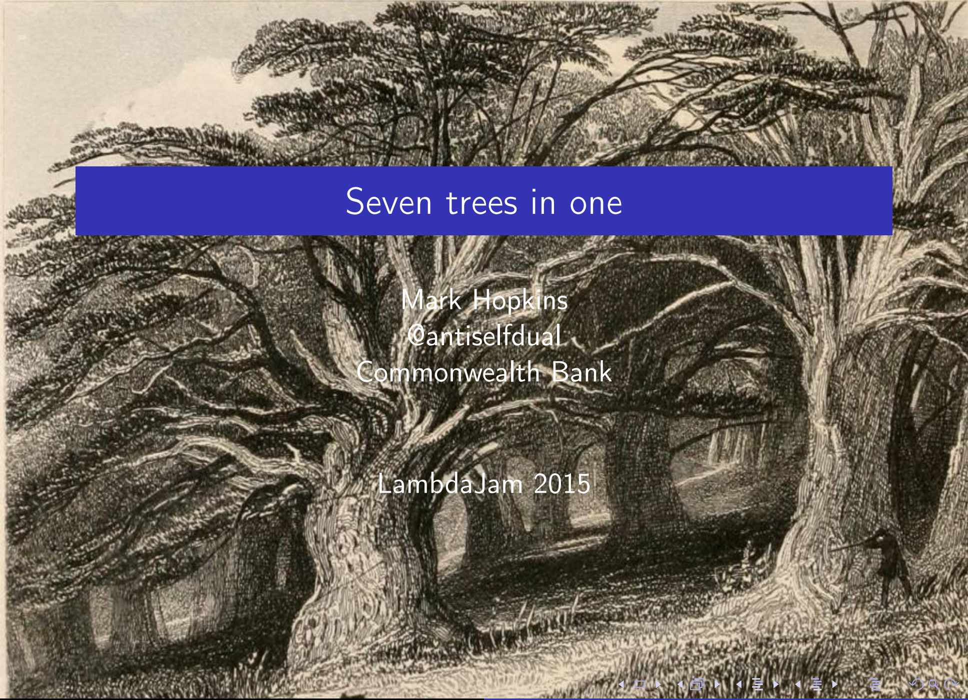


Seven trees in one

Mark Hopkins
@antiselfdual
Commonwealth Bank

LambdaJam 2015

Unlabelled binary trees

```
data Tree = Leaf | Node Tree Tree
```

Unlabelled binary trees

```
data Tree = Leaf | Node Tree Tree
```

$$T = 1 + T^2$$

$$T = 1 + T^2$$

Suspend disbelief, and **solve for** T.

$$T = 1 + T^2$$

Suspend disbelief, and **solve for** T.

$$T^2 - T + 1 = 0$$

$$T = 1 + T^2$$

Suspend disbelief, and **solve for** T.

$$T^2 - T + 1 = 0$$

$$T = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$T = 1 + T^2$$

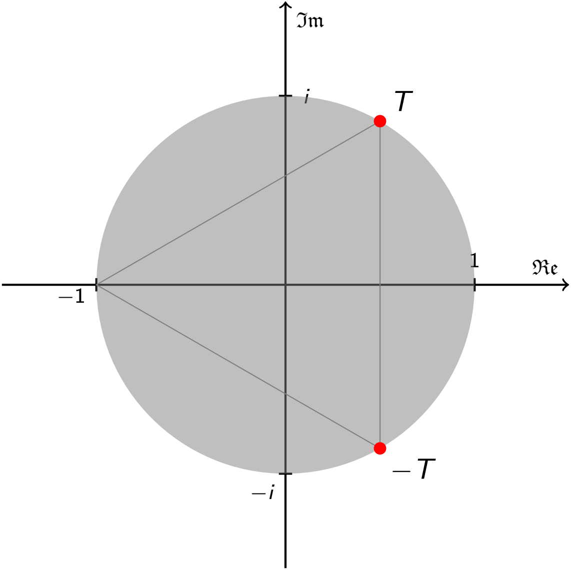
Suspend disbelief, and **solve for** T.

$$\begin{aligned} T^2 - T + 1 &= 0 \\ T &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\ &= \tfrac{1}{2} \pm \tfrac{\sqrt{3}}{2} i \end{aligned}$$

$$T = 1 + T^2$$

Suspend disbelief, and **solve for T**.

$$\begin{aligned} T^2 - T + 1 &= 0 \\ T &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\ &= \tfrac{1}{2} \pm \tfrac{\sqrt{3}}{2} i \\ &= e^{\pm \pi i/3} \end{aligned}$$



ℑm
i
T
1
ℜe
−1
−T
−i

So $T^6 = 1$.

So $T^6 = 1$. No, **obviously wrong**.

So $T^6 = 1$. No, **obviously wrong**.
What about

$$T^7 = T?$$

So $T^6 = 1$. No, **obviously wrong**.
What about

$$T^7 = T?$$

Not *obviously wrong*...

So $T^6 = 1$. No, **obviously wrong**.
What about

$$T^7 = T?$$

Not *obviously wrong*...

$$\Rightarrow \quad \text{true!}$$

Theorem

There exists an $O(1)$ bijective function from T to T^7.

Theorem

There exists an $O(1)$ bijective function from T to T^7.

i.e.

- we can pattern match on any 7-tuple of trees and put them together into one tree.

Theorem

There exists an $O(1)$ *bijective function from* T *to* T^7*.*

i.e.

- we can pattern match on any 7-tuple of trees and put them together into one tree.
- we can decompose any tree into the same seven trees it came from.

Theorem

There exists an $O(1)$ bijective function from T to T^7.

i.e.

- we can pattern match on any 7-tuple of trees and put them together into one tree.
- we can decompose any tree into the same seven trees it came from.

Actually holds for any $k = 1 \mod 6$.

Theorem

There exists an $O(1)$ bijective function from T to T^7.

i.e.

- we can pattern match on any 7-tuple of trees and put them together into one tree.
- we can decompose any tree into the same seven trees it came from.

Actually holds for any $k = 1 \mod 6$.
Not true for other values.

$T^2 \to T$

```
f :: (Tree, Tree) → Tree
t t1 t2 = Node t1 t2
```

$T^2 \to T$

```
f :: (Tree, Tree) → Tree
t t1 t2 = Node t1 t2
```

Not surjective, since we can never reach `Leaf`.

$T \to T^2$

```
f :: Tree → (Tree, Tree)
f t = Node t Leaf
```

$T \to T^2$

```
f :: Tree → (Tree, Tree)
f t = Node t Leaf
```

Not surjective either.

$T^2 \to T$, but cleverer

```
f :: (Tree, Tree) → Tree
f (t1, t2) = go (Node t1 t2)
  where
        go t        = if leftOnly t then left t else t
        leftOnly t = t == Leaf
                   || right t == Leaf && leftOnly (left t)
```

$T^2 \to T$, but cleverer

```
f :: (Tree, Tree) → Tree
f (t1, t2) = go (Node t1 t2)
  where
        go t         = if leftOnly t then left t else t
        leftOnly t = t == Leaf
                   || right t == Leaf && leftOnly (left t)
```

Bijective!

$T^2 \to T$, but cleverer

```haskell
f :: (Tree, Tree) → Tree
f (t1, t2) = go (Node t1 t2)
  where
        go t        = if leftOnly t then left t else t
        leftOnly t = t == Leaf
                   || right t == Leaf && leftOnly (left t)
```

Bijective! but not $O(1)$.

A solution

```
f :: T → (T, T, T, T, T, T, T)
f L                                                         = (L,L,L,L,L,L,L)
f (N t1 L)                                                  = (t1,N L L,L,L,L,L,L)
f (N t1 (N t2 L))                                           = (N t1 t2,L,L,L,L,L,L)
f (N t1 (N t2 (N t3 L)))                                    = (t1,N (N t2 t3) L,L,L,L,L,L)
f (N t1 (N t2 (N t3 (N t4 L))))                             = (t1,N t2 (N t3 t4),L,L,L,L,L)
f (N t1 (N t2 (N t3 (N t4 (N L L)))))                       = (t1,t2,N t3 t4,L,L,L,L)
f (N t1 (N t2 (N t3 (N t4 (N (N t5 L) L)))))                = (t1,t2,t3,N t4 t5,L,L,L)
f (N t1 (N t2 (N t3 (N t4 (N (N t5 (N t6 L)) L)))))         = (t1,t2,t3,t4,N t5 t6,L,L)
f (N t1 (N t2 (N t3 (N t4 (N (N t5 (N t6 (N t7 t8))) L))))) = (t1,t2,t3,t4,t5,t6,N t7 t8)
f (N t1 (N t2 (N t3 (N t4 (N t5 (N t6 t7))))))              = (t1,t2,t3,t4,t5,N t6 t7,L)
```

Where did this come from

$$T = 1 + T^2$$

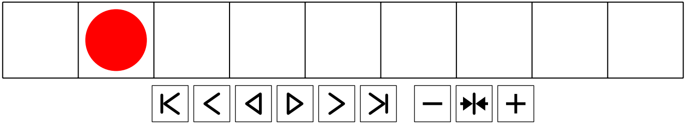
Where did this come from

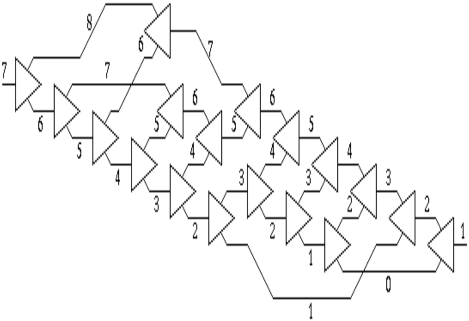
$$T = 1 + T^2$$
$$T^k = T^{k-1} + T^{k+1}$$

Penny game

T^0	T^1	T^2	T^3	T^4	T^5	T^6	T^7	T^8

- start with a penny in position 1.
- aim is to move it to position 7 by splitting and combining




8
7
6
7
7
6
6
6
5
5
5
5
5
4
4
4
4
4
3
3
3
3
3
2
2
2
2
1
1
0
1

Why did this work?
If we have a type isomorphism $T \cong p(T)$ then

Why did this work?
If we have a type isomorphism $T \cong p(T)$ then

$q_1(T) \cong q_2(T)$ as types

$\iff q_1(x) \cong q_2(x)$ in the rig $\mathbb{N}[x]/(p(x) = x)$

Why did this work?
If we have a type isomorphism $T \cong p(T)$ then

$q_1(T) \cong q_2(T)$ as types

$\iff q_1(x) \cong q_2(x)$ in the rig $\mathbb{N}[x]/(p(x) = x)$

$\Rightarrow q_1(x) \cong q_2(x)$ in the ring $\mathbb{Z}[x]/(p(x) = x)$

Why did this work?
If we have a type isomorphism $T \cong p(T)$ then

$q_1(T) \cong q_2(T)$ as types

$\iff q_1(x) \cong q_2(x)$ in the rig $\mathbb{N}[x]/(p(x) = x)$

$\Rightarrow q_1(x) \cong q_2(x)$ in the ring $\mathbb{Z}[x]/(p(x) = x)$

$\Rightarrow q_1(z) \cong q_2(z)$ for all $z \in \mathbb{C}$ such that $p(z) = z$.

Why did this work?
If we have a type isomorphism $T \cong p(T)$ then

$q_1(T) \cong q_2(T)$ as types

$\iff q_1(x) \cong q_2(x)$ in the rig $\mathbb{N}[x]/(p(x) = x)$

$\Rightarrow q_1(x) \cong q_2(x)$ in the ring $\mathbb{Z}[x]/(p(x) = x)$

$\Rightarrow q_1(z) \cong q_2(z)$ for all $z \in \mathbb{C}$ such that $p(z) = z$.

And, under some conditions, the reverse implications hold.

Summary

- Simple arithmetic helps us find non-obvious type isomorphisms

Extensions

- Are there extensions to datatypes of decorated trees? (multivariate polynomials)
- What applications are there?

Extensions

- Are there extensions to datatypes of decorated trees? (multivariate polynomials)
- What applications are there?
    - important when writing a compiler to know when two types are isomorphic

Extensions

- Are there extensions to datatypes of decorated trees? (multivariate polynomials)
- What applications are there?
  - important when writing a compiler to know when two types are isomorphic
  - It could interesting to split up a tree-shaped stream into seven parts

- Rich theory behind isomorphisms of polynomial types
- brings together a number of fields
  - distributive categories
  - theory of rigs (semirings)
  - combinatorial species
  - type theory

Further reading

- *Seven Trees in one*, Andreas Blass, Journal of Pure and Applied Algebra
- *On the generic solution to $P(X) = X$ in distributive categories*, Robbie Gates
- *Objects of Categories as Complex Numbers*, Marcelo Fiore and Tom Leinster
- *An Objective Representation of the Gaussian Integers*, Marcelo Fiore and Tom Leinster
- http://rfcwalters.blogspot.com.au/2010/06/robbie-gates-on-seven-trees-in-one.html
- http://blog.sigfpe.com/2007/09/arboreal-isomorphisms-from-nuclear.html

Challenge

Consider this datatype (Motzkin trees):

```
data Tree = Zero | One Tree | Two Tree Tree
```

Challenge

Consider this datatype (Motzkin trees):

```
data Tree = Zero | One Tree | Two Tree Tree
```

$$T = 1 + T + T^2$$

Challenge

Consider this datatype (Motzkin trees):

```
data Tree = Zero | One Tree | Two Tree Tree
```

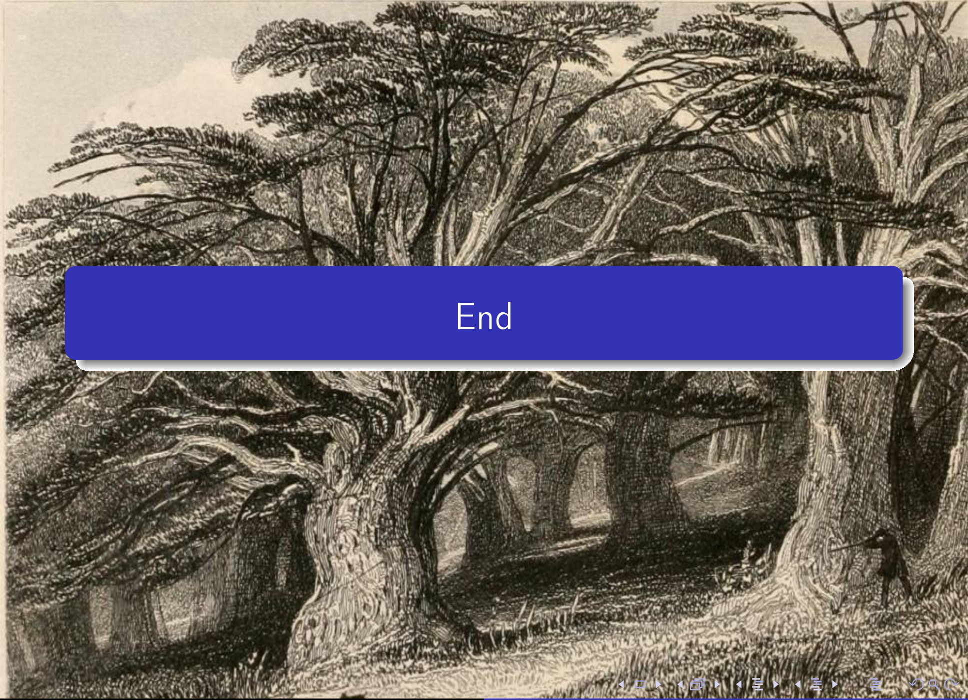
$$T = 1 + T + T^2$$

Show that $T^5 \cong T$

- by a nonsense argument using complex numbers
- by composing bijections (the penny game)
- implement the function and its inverse in a language of your choice

Photo credits

- *The Druid's Grove, Norbury Park: Ancient Yew Trees* by Thomas Allom 1804-1872 http://www.victorianweb.org/art/illustration/allom/1.html



End