

DANIELE PROCIDA

MINDS & MACHINES & PYTHON

DANIELE PROCIDA

- ▶ Community manager, Divio
- ▶ django CMS developer
- ▶ Django core developer
- ▶ Board member, Django Software Foundation
- ▶ daniele.procida@divio.com
- ▶ EvilDMP (IRC, GitHub, Twitter)



DANIELE PROCIDA

- ▶ Community manager, Divio
- ▶ django CMS developer
- ▶ Django core developer
- ▶ Board member, Django Software Foundation
- ▶ daniele.procida@divio.com
- ▶ EvilDMP (IRC, GitHub, Twitter)



DANIELE PROCIDA

- ▶ Community manager, [Divio](#)
- ▶ django CMS developer
- ▶ Django core developer
- ▶ Board member, [Django Software Foundation](#)
- ▶ daniele.procida@divio.com
- ▶ [EvilDMP](#) (IRC, GitHub, Twitter)

The Django logo, consisting of the word "django" in a lowercase, white, sans-serif font.

MINDS & MACHINES & PYTHON



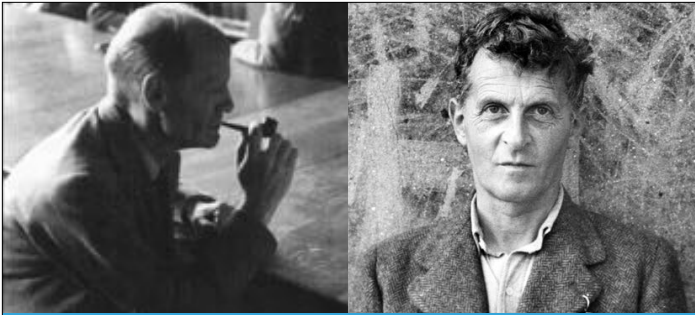
THE [...] QUESTION, "CAN MACHINES THINK?" [IS] TOO MEANINGLESS TO DESERVE DISCUSSION.

Alan Turing

In 1950, in his paper "Computing Machinery and Intelligence" in *Mind*, Alan Turing brilliantly turned the question of whether machines could think on its head, arguing that "the [...] question, *Can machines think?* [is] too meaningless to deserve discussion."

Instead of asking whether thought can occur inside a digital computer, he invited us to consider whether a machine could, in principle, be indistinguishable from a human, in those respects that allow us to say that a human thinks or has intelligence.

He was applying insights from the philosophy of mind that emerged in the first half of the 20th century, which also moved the questions of mind and intelligence away from a concern with the metaphysics of thought and the intrinsic nature of consciousness, towards questions like *what constitutes an interaction with intelligence?* or *how do we recognise other minds?*



GILBERT RYLE

LUDWIG WITTGENSTEIN

In other words, Turing, like philosophers such as Gilbert Ryle and Ludwig Wittgenstein, argued that the search for mind was not a search into inner mysteries, but a matter of recognising what was in front of us all along.

THE QUEST FOR ARTIFICIAL INTELLIGENCE



We can say that this represents a fork in the study of artificial intelligence. Turing's argument in a way freed researchers from having to clamber into those very opaque mysteries, and instead concentrate on producing, for example, interactions that seem like encounters with intelligence.

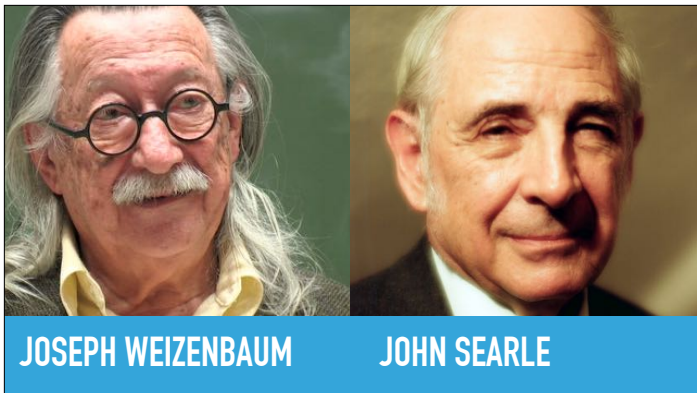
This has not just been the dominant fork since then, it has also been by far the most successful.

Its fruits are all around us, literally in our pockets. The power of machines to recognise - language, faces, text, road traffic - and to respond appropriately, has taken us by surprise. I think that by Turing's account we are indeed living at the beginning of an age of machine intelligence.

But, what about that other fork? What about that lonelier furrow, in which people *have* tried to unpick some of the mysteries of thought itself. Turing was a very intelligent thinker, but just because he dismissed a question as "too meaningless to deserve discussion" doesn't mean that it is.

THE QUEST FOR ARTIFICIAL INTELLIGENCE





JOSEPH WEIZENBAUM

JOHN SEARLE

Some fine minds have in fact disagreed with him enough to expend their efforts there, including researchers in computing, like Joseph Weizenbaum, and philosophers such as John Searle.

Weizenbaum's book *Computer power and human reason* was as important as Turing's paper. He was also the author of Eliza - his attempt to demonstrate in a concrete way what was unsatisfactory about Turing's analysis.

THE QUEST FOR ARTIFICIAL INTELLIGENCE

- ▶ WHAT GOES ON INSIDE
- ▶ CONSCIOUSNESS
- ▶ THOUGHT

- ▶ WHAT GOES ON OUTSIDE
- ▶ BEHAVIOUR
- ▶ INTERACTIONS

In this talk, I want to go back to that other fork, because I too think that Turing's analysis is flawed and inadequate. I think that the incredible advances, the Siris and the chat-bots and the self-driving cars, are, as far as we are actually concerned with intelligence, a *dead-end*.

Like Turing, these efforts begin *from the outside*. They're not really concerned with the nature of intelligence itself, but with the challenge of creating an appearance of it: *what does intelligence look like?* In narrow, limited spheres, the appearance can be very successful, and the more successful the appearance, the more easily we fall into using the language of intelligence around the behaviour - and into forgetting that we are dealing merely with a simulacrum of intelligence, that is no closer to consciousness than a stone.

I think that the more interesting question is the one that has become neglected, that begins *from the inside*, that asks *what lies at the heart of intelligence? Where does consciousness arise?*

THE QUEST FOR ARTIFICIAL INTELLIGENCE



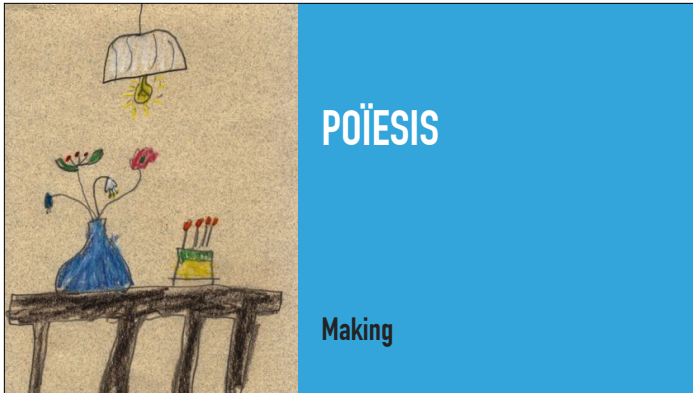
THE QUEST FOR ARTIFICIAL INTELLIGENCE

- ▶ We can find some of the most interesting insights into intelligence in the most basic programming concepts.
- ▶ We can find their counterparts in the work of poets, writers and artists.

As programmers, I think we have the concepts and the tools to investigate this in useful, interesting ways. In this talk I want to argue two things:

- * that the programming concepts that give us the most interesting insights into this are in fact the most basic ones
- * that we can find their counterparts in the work of poets, writers and artists, and use this work to help understand and inspire the quest into intelligence and consciousness

For this, let's turn to the concept of *poiësis*.



The word *poetry* comes from the Greek *poïesis*, meaning *making*. Its roots are in the ancient Greek verb *to make* or *to produce*.

It was a verb, an activity, before it was a noun.

Poïesis doesn't mean making or production in the sense of *manufacture*. Think of the verbs *faire* or *fare*, in French and Italian, which mean both *make* and *do*. Or think of the way we use *make* as in *making friends* or *making love*.

It's not concerned with material or technical construction, but with a transformation in the world; an act or process of bringing forth. In *poïesis*, something becomes another kind of thing altogether - a new thing emerges.

Poetry can be considered a kind of *poïesis*. And I think that programming also represents *poïesis*, that poets and programmers, because their work is *poïesis*, making, can help us understand or at least usefully explore some quite deep questions about ourselves; in particular, about the nature of human consciousness, thought, the mind.

So let's dive right into the work of programmers and poets and artists, and see what characterises a certain kind of *poïesis*.

RULE-GOVERNED PLAY

- ▶ rules
- ▶ processes
- ▶ play

Programmers seem to be particularly fascinated by *rule-governed play*, and to respond to it strongly when they find it, in poetry, music and other art.

There are some notable writers and artists whose work and ideas speak to programmers.

I believe it's the way that programmers think that makes them especially ready to understand and appreciate the **intersection of rules, processes and play** that characterises these work and ideas.

Quite often, it's really obvious that something, even if it has nothing to do with programmers, is going to appeal to programmers: they will *get it*.

We could spend a lot of time discussing exactly why this is so, but I think that part of the answer anyway is that *rule-governed play* takes place in *systems*, and that systems appeal very much to programmers.

SYSTEMS

- ▶ loops
- ▶ self-reference
- ▶ hierarchies

In fact three of the things they love best in systems are also things that are very, very interesting outside programming.

What's more, they are the things that make rule-governed play particularly interesting.

They also represent some of the most basic concepts or structures in programming, ones that determine the way programming itself works, and they are:

loops

self-reference

hierarchies

LOOPS

Let's start with loops.

```
10 PRINT "HELLO"  
20 GOTO 10
```

Even the simplest possible loop represents power.

It doesn't matter how trivial it is, a loop can still unleash an infinite sequence, and the computer will try to make it real.

There are many constructs in programming, but the one that I love the best, the one that seems most beautiful and powerful, is the loop.

Loops are perfect and simple.

But still - the fascination of GOTO 10 is rather limited. GOTO 10 is only interesting for being infinite, rather than for anything new that comes out of it. It's not **about** anything, and it's always the same.

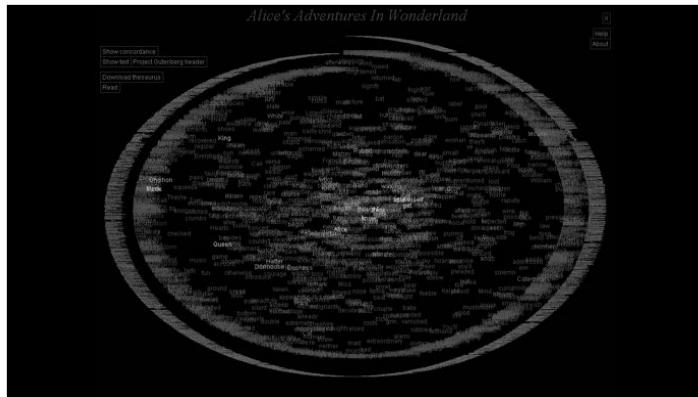
So this particular loop isn't very interesting, but when we apply loops to other things, then they get more intriguing.

TextArc

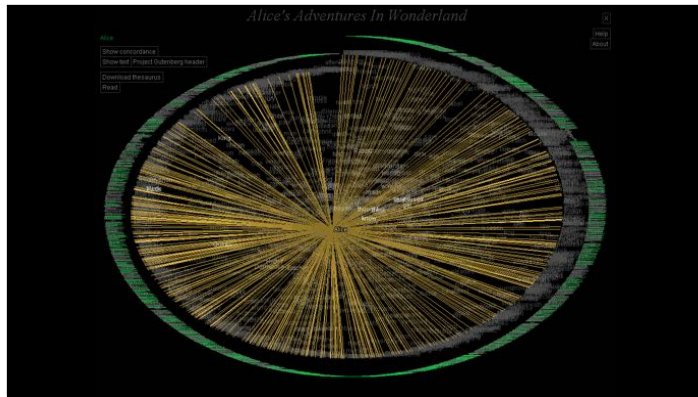
Many years ago, W. Bradford Paley produced TextArc, an application running at textarc.org. TextArc is beautiful and magical.

It loops over a text, and represents it, visually.

(You can try running TextArc on a modern system, but although some have reported success on using OS X/Firefox, I have not been able to run it except in a virtual machine running an older version of Windows.)

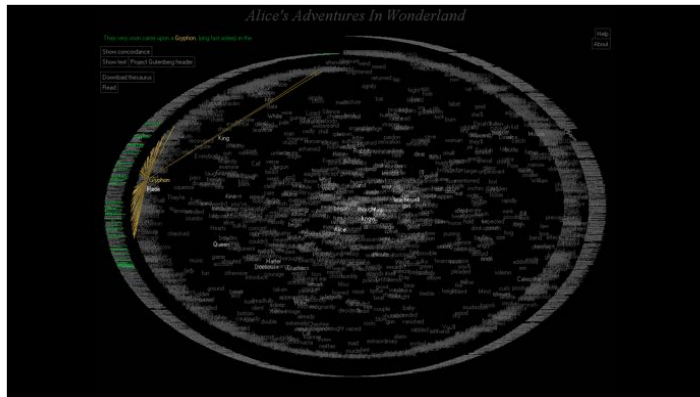


Here's the text of *Alice's Adventures in Wonderland*, represented in TextArc.

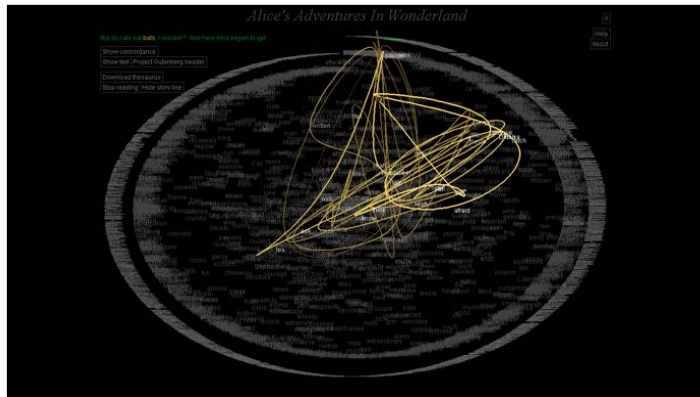


Here's the word - and the character - *Alice*, in relation to the entire text.

We can see that Alice is at the centre of the story - literally.



Here's the Gryphon - literally, a peripheral character.



And here is a linear sequence through the actual text.

STRUCTURE AND METADATA

In TextArc, we don't really get a sense anymore of the *content* of the work. Instead, we get its *structure*, exposed by a loop, presented as *metadata*.

Why do structure and metadata matter?

We do lose something, when we lose the content, but we also *gain* something, because it lets us see new meaning that perhaps was previously obscured.

We see who is associated with whom, or what. We can see who appears when, with whom. We can see who dominates the story. In the case of TextArc, we can see it, literally.

STRUCTURE & METADATA

WHO'S SPEAKING TO WHOM?

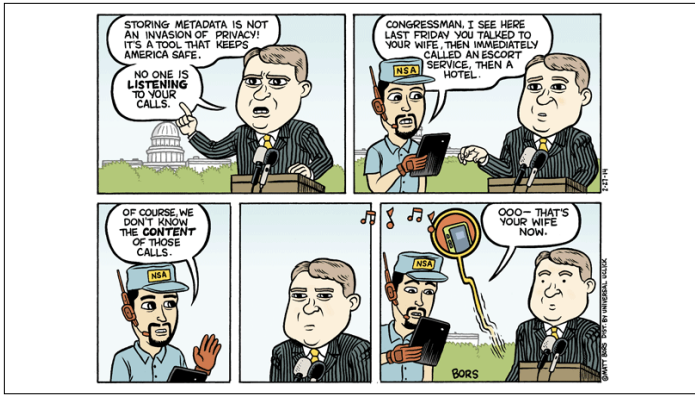
Don't be distracted by the fact that what TextArc does is amusing and literary; it's also powerful and significant.

This is not lost on large corporations and government agencies, who understand its power and significance very well.

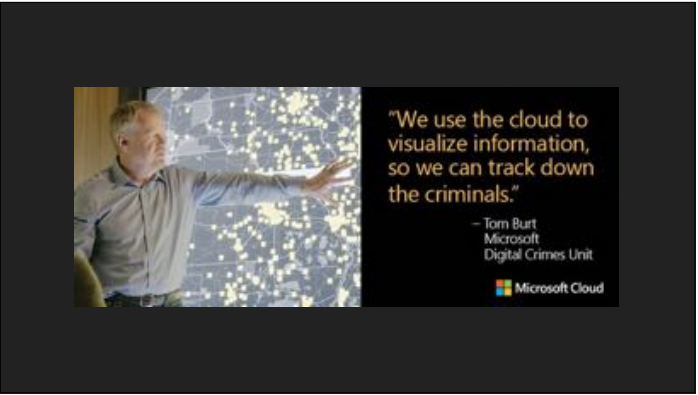
Google and Facebook know what structure and metadata mean; they take it very seriously.

In various countries in the world, right now or recently or perhaps simply coming soon, governments are anxious to have legislation enacted that will give their agencies the right to similar information about our communications and activities.

Of course, they would *never* do anything like eavesdrop on their citizens, recording their conversations. That would be a terrible invasion of privacy that only an oppressive regime would do. But they would like to know who is talking to whom...



We're programmers; we know how much information can be found in metadata, and how much can be done with it.



ALICE AND HAMLET

Alice is the first text offered on TextArc. You can run TextArc on any text, but the next text listed there is Shakespeare's *Hamlet*.

For some reason, *Alice* and *Hamlet* are the two texts that keep appearing in the work of artists, and programmers, who want to explore texts in new ways.

BERNARDO
Who's there?
FRANCISCO
Nay, answer me: stand, and unfold yourself.

BERNARDO
Long live the king!

FRANCISCO
Bernardo?

BERNARDO
'Tis

FRANCISCO
You come most carefully upon your hour.

BERNARDO
'Tis now struck twelve; get thee to bed, Francisco.

FRANCISCO
For this relief much thanks: 'tis bitter cold,

And I am sick at heart.

BERNARDO
Have you had quiet guard?

FRANCISCO
Not a mouse stirring.

BERNARDO
Well, good night.

If you do meet Horatio and Marcellus,

The rivals of my watch, bid them make haste.

FRANCISCO
I think I hear them. Stand, ho! Who's there?

Enter **HORATIO** and **MARCELLUS**

HORATIO
Friends to this ground.

MARCELLUS

And Iagmen to the Dane.

FRANCISCO
Give you good night.

MARCELLUS
O, farewell, honest soldier:

Who hath relieved you?

FRANCISCO
Bernardo has my place.

Give you good night.

Exit

MARCELLUS
Holla! Bernardo!

BERNARDO
Say

What, is Horatio there?

HORATIO
A piece of him.

BERNARDO
Welcome, Horatio: welcome, good Marcellus.

MARCELLUS
What, has this thing appear'd again to-night?

BERNARDO
I have seen nothing.

MARCELLUS
Horatio says 'tis but our fantasy.

And will not let belief take hold of him.

Touching this dreaded sight, twice seen of us:

Therefore I have entreated him along

With us to watch the minutes of this night;

That if again this apparition come,

He may approve our eyes and speak to it.

HORATIO

Tush, tush, 'twill not appear.

BERNARDO

Sit down awhile;

And let us once again assail your ears,

That are so fortified against our story

What we have two nights seen.

HORATIO

Well, sit we down,

And let us hear Bernardo speak of this.

BERNARDO

Last night of all,

When yond same star that's westward from the

pole

Had made his course to illumine that part of heaven

Where now it burns, Marcellus and myself,

The bell then beating one,—

Enter Ghost

MARCELLUS
Peace, break thee off; look, where it comes again!

BERNARDO

In the same figure, like the king that's dead.

MARCELLUS

Thou art a scholar; speak to it, Horatio.

BERNARDO

Looks it not like the king? mark it, Horatio.

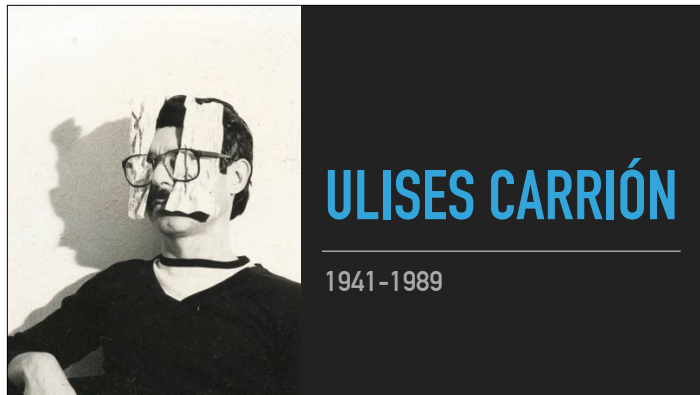
HORATIO

Most like: it harrows me with fear and wonder.

MARCELLUS

Question it, Horatio.

Here's *Hamlet*.



ULISES CARRIÓN

1941-1989

And here is Ulises Carrión, a Mexican artist.



I'm going to play you part of his *Hamlet for two voices*.

BERNARDO Who's there?	And flagmen to the Dane.	HORATIO Tush, tush, 'twill not appear.
FRANCISCO Nay, answer me: stand, and unfold yourself.	FRANCISCO Give you good night.	BERNARDO Sit down awhile;
BERNARDO Long live the king!	MARCELLUS O, farewell, honest soldier: Who hath relieved you?	And let us once again assail your ears, That are so fortified against our story What we have two nights seen.
FRANCISCO Bernardo?	FRANCISCO Bernardo has my place.	HORATIO Well, sit we down, And let us hear Bernardo speak of this.
BERNARDO 'Is	MARCELLUS Holla! Bernardo!	BERNARDO Last night of all, When yond same star that's westward from the pole
FRANCISCO You come most carefully upon your hour.	BERNARDO 'Tis now struck twelve; get thee to bed, Francisco.	Had made his course to illumine that part of heaven Where now it burns, Marcellus and myself, The bell then beating one.- Enter Ghost
FRANCISCO For this relief much thanks: 'tis bitter cold, And I am sick at heart.	HORATIO What, is Horatio there?	MARCELLUS Peace, break thee off; look, where it comes again!
BERNARDO Have you had quiet guard?	BERNARDO A piece of him.	BERNARDO In the same figure, like the king that's dead.
FRANCISCO Not a mouse stirring.	BERNARDO Welcome, Horatio: welcome, good Marcellus.	MARCELLUS Thou art a scholar; speak to it, Horatio.
BERNARDO Well, good night.	MARCELLUS What, has this thing appear'd again to-night?	BERNARDO Looks it not like the king? mark it, Horatio.
If you do meet Horatio and Marcellus, The rivals of my watch, bid them make haste.	BERNARDO I have seen nothing.	HORATIO Most like: it harrows me with fear and wonder.
FRANCISCO I think I hear them. Stand, ho! Who's there?	MARCELLUS Horatio says 'tis but our fantasy, And will not let belief take hold of him Touching this dreaded sight, twice seen of us: Therefore I have entreated him along With us to watch the minutes of this night; That if again this apparition come, He may approve our eyes and speak to it.	MARCELLUS Question it, Horatio.
Enter HORATIO and MARCELLUS		
HORATIO Friends to this ground,		
MARCELLUS		

BERNARDO Who's there?	And flagmen to the Dane. FRANCISCO Give you good night.	HORATIO Tush, tush, 'twill not appear.
FRANCISCO Nay, answer me: stand, and unfold yourself.	MARCELLUS O, farewell, honest soldier: Who hath relieved you?	BERNARDO Sit down awhile; And let us once again assail your ears, That are so fortified against our story What we have two nights seen.
BERNARDO Long live the king!	FRANCISCO Bernardo has my place. Give you good night. Exit	HORATIO Well, sit we down, And let us hear Bernardo speak of this.
FRANCISCO Bernardo?	MARCELLUS Holla! Bernardo!	BERNARDO Last night of all, When yond same star that's westward from the pole Had made his course to illumine that part of heaven Where now it burns, Marcellus and myself, The bell then beating one,— Enter Ghost
BERNARDO 'Is	BERNARDO Say What, is Horatio there?	MARCELLUS Peace, break thee off; look, where it comes again!
FRANCISCO You come most carefully upon your hour.	HORATIO A piece of him.	BERNARDO In the same figure, like the king that's dead.
BERNARDO 'Tis now struck twelve; get thee to bed, Francisco.	BERNARDO Welcome, Horatio: welcome, good Marcellus.	MARCELLUS Thou art a scholar; speak to it, Horatio.
FRANCISCO For this relief much thanks: 'tis bitter cold, And I am sick at heart.	MARCELLUS What, has this thing appear'd again to-night?	BERNARDO Looks it not like the king? mark it, Horatio.
BERNARDO Have you had quiet guard?	BERNARDO I have seen nothing.	HORATIO Most like: it harrows me with fear and wonder.
FRANCISCO Not a mouse stirring.	MARCELLUS Horatio says 'tis but our fantasy, And will not let belief take hold of him Touching this dreaded sight, twice seen of us: Therefore I have entreated him along With us to watch the minutes of this night; That if again this apparition come, He may approve our eyes and speak to it.	MARCELLUS Question it, Horatio.
BERNARDO Well, good night. If you do meet Horatio and Marcellus, The rivals of my watch, bid them make haste.	BERNARDO Exit	
FRANCISCO I think I hear them. Stand, ho! Who's there?		
Enter HORATIO and MARCELLUS		
HORATIO Friends to this ground, MARCELLUS		

BERNARDO Who's there?	And liegemen to the Dane.	HORATIO Tush, tush, 'twill not appear.
FRANCISCO Nay, answer me: stand, and unfold yourself.	FRANCISCO Give you good night.	BERNARDO Sit down awhile;
BERNARDO Long live the king!	MARCELLUS O, farewell, honest soldier: Who hath relieved you?	And let us once again assail your ears, That are so fortified against our story What we have two nights seen.
FRANCISCO Bernardo?	FRANCISCO Bernardo has my place.	HORATIO Well, sit we down, And let us hear Bernardo speak of this.
BERNARDO 'Tis now struck twelve; get thee to bed, Francisco.	Exit	BERNARDO Last night of all, When yond same star that's westward from the pole
FRANCISCO You come most carefully upon your hour.	MARCELLUS Holla! Bernardo!	Had made his course to illumine that part of heaven Where now it burns, Marcellus and myself, The bell then beating on:--
BERNARDO 'Tis now struck twelve; get thee to bed, Francisco.	BERNARDO Say	BERNARDO Welcome, Horatio: welcome, good Marcellus.
FRANCISCO For this relief much thanks: 'tis bitter cold, And I am sick at heart.	HORATIO A piece of him.	MARCELLUS Peace, break thee off; look, where it comes again!
BERNARDO Have you had quiet guard?	BERNARDO What, has this thing appear'd again to-night?	BERNARDO In the same figure, like the king that's dead.
FRANCISCO Not a mouse stirring.	MARCELLUS I have seen nothing.	MARCELLUS Thou art a scholar; speak to it, Horatio.
BERNARDO Well, good night. If you do meet Horatio and Marcellus, The rivals of my watch, bid them make haste.	MARCELLUS Horatio says 'tis but our fantasy, And will not let belief take hold of him Touching this dreaded sight, twice seen of us: Therefore I have entreated him along With us to watch the minutes of this night; That if again this apparition come, He may approve our eyes and speak to it.	BERNARDO Looks it not like the king? mark it, Horatio.
FRANCISCO I think I hear them. Stand, ho! Who's there? Enter HORATIO and MARCELLUS		HORATIO Most like: it harrows me with fear and wonder.
HORATIO Friends to this ground.		MARCELLUS Question it, Horatio.
MARCELLUS		

And here's Ulises Carrión's "Hamlet for Two Voices":

- audio file: <https://www.dropbox.com/s/vyi3psnt0da2hu0/Hamlet%20for%20two%20voices.mp3?dl=0> (complete version)
- available from <https://boomkat.com/products/the-poet-s-tongue>

"Hamlet for Two Voices" is funny. I think it's hilarious. But it's not just funny. What do we learn from this?

Ulises Carrión's performance is a new interpretation of Hamlet, a Hamlet for two voices, left and right. It's the whole of Hamlet - *without any content*. Only the structure remains.

So what are we left with? What have we lost?

It's true, you won't know from this that Hamlet is about revenge, murder or desire. You won't know that Hamlet is a prince, or that the story is set in Denmark.

But, there is a lot to learn.

We learn who is still around at the end (after all, if you're a character in a Shakespeare tragedy there's no guarantee you'll still be alive at the end of the story).

We can discover whom the story revolves around, who dominates conversations, who hangs around with whom. We can guess who is *conspiring* with whom (just like a government agency...).

We have lost something, we have thrown some information away, but maybe we expose some previously-hidden information in doing so.

THE PROGRAMMER'S SOUL

Most people react to this the same way when they hear it for the first time: *what on earth is this?*

But then it divides the world into programmers and non-programmers.

Non-programmers ask *What's the point?* They are baffled, at best, or worse, irritated. Some people are really annoyed by it.

But the reaction of programmers is different: they *get it*.

(Of course it's possible that a programmer won't like it, or that a non-programmer will get it instantly - what it really does is divide the world into the people who have the *soul* of a programmer, and those who don't.)

Programmers understand this kind of thing. Programmers understand loops, what's important about meta-data, and how to recognise a key-value pair when they see one.

KEY/VALUE PAIRS

Hamlet for two voices, in Python: <https://gist.github.com/evildmp/7f4702efac3a164629dd9c31bfff8ee4>

FUN WITH LANGUAGE

PLAY

Computers are the perfect tools for this sort of thing, for exploring, analysing, discovering texts and language.

If you're interested in this, it's very accessible and easy to explore further.

What could you discover with a simple program? Could you extract the punctuation from a text, and recognise the author by the punctuation? Or by the whitespace?

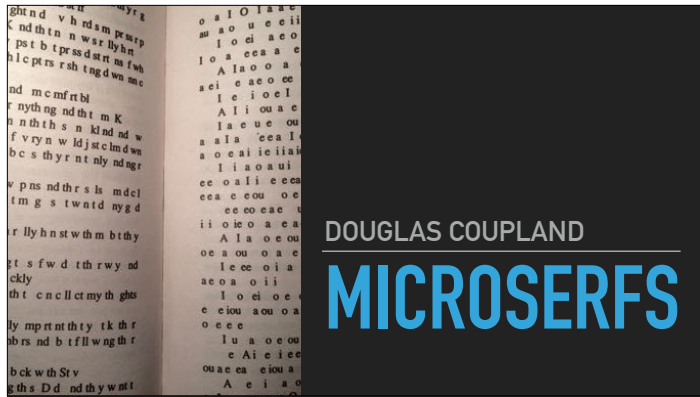
What else is there to learn by looping over text in this way?

OUVROIR DE LITTÉRATURE POTENTIELLE

OULIPO

In fact there is a whole field of experimental literature that does this, to analyse texts, synthesise them, and deconstruct them.

There's the Oulipo group of writers, mathematicians, artists, which includes such figures as Italo Calvino and Paul Fournel. Their interest in structure, meta-data and other "non-content" of texts makes them especially intriguing for programmers, and their work is ripe for exploration in programming.



And you don't have to go to experimental literature.

Here are a couple of pages from *Microserfs*, the 1994 novel by Douglas Coupland *about* programmers, with consonants and vowels lifted out from each other.

PLAY

- ▶ the looseness in a mechanical system
- ▶ *play* only exists within rules and constraints
- ▶ even very precise systems must have *play* in them
- ▶ the rules and connections of language leave room for *play*
- ▶ just like artists and writers, programmers like to play with and exploit *play*

We can play endless games with texts of this kind.

We're referring to *games*, and talk about *playing*, and some of it is a lot of fun, but it's also serious.

Play doesn't just refer to fun. Play is the looseness in a mechanical connection, the extent to which one side of a link is free and undetermined by the other.

It's also significant to note that play in this sense depends upon the linkage, the connection, the constraint - if there were no constraint at all, there wouldn't be play, there'd be nothing, just as if there were no looseness in the connection at all.

Play only exists in the context of constraint and rules, and we find it in the connections, where the joints permit movement.

If there were no play at all, the entire mechanical system would be locked solid.

So, the rules and connections of language are just open and loose enough to allow both for rigorous meaning, and play, *within the same system*.

Play, of this kind, is also something that's enjoyed and exploited by artists and writers and programmers.

We've seen some examples of looping, but this kind of *play* becomes even more interesting when the looping becomes *self-referential*.

SELF-REFERENTIAL LOOPS



Here is Ulises Carrión again, with his “First Spanish Lesson”.

FIRST SPANISH LESSON

- ▶ **Es español.**
It is Spanish.
- ▶ **¿Es español? Sí, es español.**
Is it Spanish? Yes, it is Spanish.
- ▶ **¿Es 'ese' español? Sí, 'ese' es español.**
Is 'that' Spanish? Yes, 'that' is Spanish.
- ▶ **¿Es 'ese español' español? Sí, 'ese español' es español.**
Is 'that Spaniard' Spanish? Yes, 'that Spaniard' is Spanish.
- ▶ **¿Es 'ese es español' español? Sí, 'ese es español' es español.**
Is "that is Spanish" Spanish? Yes, 'that is Spanish' is Spanish.
- ▶ **¿Es 'si es español' español? Sí, 'si es español' es español.**
Is 'if it is Spanish' Spanish? Yes, 'if it is Spanish' is Spanish.
- ▶ **¿Es 'si, es español' español? Sí, 'Si, es español' es español.**
Is 'yes, it is Spanish' Spanish? Yes, 'yes, it is Spanish' is Spanish.
- ▶ **¿Es 'si ese' español? No, 'si ese' no es español.**
Is 'if that' Spanish? No, 'if that' is not Spanish.
- ▶ **¿Es "ese no es español" español? Sí, "ese no es español" es español.**
Is "that is not Spanish" Spanish? Yes, "that is not Spanish" is Spanish.
- ▶ **¿Es 'españoles' español? Sí, 'españoles' es español.**
Is 'Spaniards' Spanish? Yes, 'Spaniards' is Spanish.
- ▶ **¿Es 'es españoles' español? No, 'es españoles' no es español.**
Is "Is Spaniards' Spanish? No, 'Is Spaniards' is not Spanish.

- audio file: <https://www.dropbox.com/s/zcvw2ouks4u1jbi/Poet%27s%20tongue%20-%20First%20Spanish%20Lesson.mp3?dl=0> (excerpt, 4 minutes 19 seconds)
- available from <https://boomkat.com/products/the-poet-s-tongue>

FIRST SPANISH LESSON

- ▶ **Es español.**
It is Spanish.
- ▶ **¿Es español? Sí, es español.**
Is it Spanish? Yes, it is Spanish.
- ▶ **¿Es 'ese' español? Sí, 'ese' es español.**
Is 'that' Spanish? Yes, 'that' is Spanish.
- ▶ **¿Es 'ese español' español? Sí, 'ese español' es español.**
Is 'that Spaniard' Spanish? Yes, 'that Spaniard' is Spanish.
- ▶ **¿Es 'ese es español' español? Sí, 'ese es español' es español.**
Is "that is Spanish" Spanish? Yes, 'that is Spanish' is Spanish.
- ▶ **¿Es 'si es español' español? Sí, 'si es español' es español.**
Is 'if it is Spanish' Spanish? Yes, 'if it is Spanish' is Spanish.
- ▶ **¿Es 'si, es español' español? Sí, 'Si, es español' es español.**
Is 'yes, it is Spanish' Spanish? Yes, 'yes, it is Spanish' is Spanish.
- ▶ **¿Es 'si ese' español? No, 'si ese' no es español.**
Is 'if that' Spanish? No, 'if that' is not Spanish.
- ▶ **¿Es "ese no es español" español? Sí, "ese no es español" es español.**
Is "that is not Spanish" Spanish? Yes, "that is not Spanish" is Spanish.
- ▶ **¿Es 'españoles' español? Sí, 'españoles' es español.**
Is 'Spaniards' Spanish? Yes, 'Spaniards' is Spanish.
- ▶ **¿Es 'es españoles' español? No, 'es españoles' no es español.**
Is "Is Spaniards' Spanish? No, 'Is Spaniards' is not Spanish.

FIRST SPANISH LESSON

- ▶ **Es español.**
It is Spanish.
- ▶ **¿Es español? Sí, es español.**
Is it Spanish? Yes, it is Spanish.
- ▶ **¿Es 'ese' español? Sí, 'ese' es español.**
Is 'that' Spanish? Yes, 'that' is Spanish.
- ▶ **¿Es 'ese español' español? Sí, 'ese español' es español.**
Is 'that Spaniard' Spanish? Yes, 'that Spaniard' is Spanish.
- ▶ **¿Es 'ese es español' español? Sí, 'ese es español' es español.**
Is "that is Spanish" Spanish? Yes, 'that is Spanish' is Spanish.
- ▶ **¿Es 'si es español' español? Sí, 'si es español' es español.**
Is 'if it is Spanish' Spanish? Yes, 'if it is Spanish' is Spanish.
- ▶ **¿Es 'si, es español' español? Sí, 'Si, es español' es español.**
Is 'yes, it is Spanish' Spanish? Yes, 'yes, it is Spanish' is Spanish.
- ▶ **¿Es 'si ese' español? No, 'si ese' no es español.**
Is 'if that' Spanish? No, 'if that' is not Spanish.
- ▶ **¿Es "ese no es español" español? Sí, "ese no es español" es español.**
Is "that is not Spanish" Spanish? Yes, "that is not Spanish" is Spanish.
- ▶ **¿Es 'españoles' español? Sí, 'españoles' es español.**
Is 'Spaniards' Spanish? Yes, 'Spaniards' is Spanish.
- ▶ **¿Es 'es españoles' español? No, 'es españoles' no es español.**
Is "Is Spaniards' Spanish? No, 'Is Spaniards' is not Spanish.

This is language about language. It's looping over its own structures, recursing on itself. It's language eating language.

Usually, when something consumes itself it reduces to nothingness: in this case, something new and magical comes out.

It's a perfect example of playing with rules and processes. It's the kind of thing programmers feel deep in their souls.

In a way, this *is* a program, following its own internal logic, obeying its own rules.

What it makes me think is:

FIRST PYTHON LESSON

Could we create a *First Python lesson*?

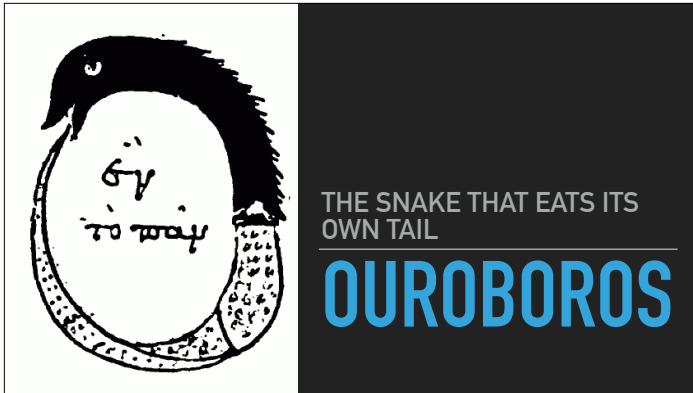
Could a Python programme consume and give birth *to itself*, regenerating and answering itself in the same way?

WE HAVE THE TECHNOLOGY

```
eval(compile(string, '<string>', 'exec'))
```

We have the technology.

It's actually fairly simple to get Python to consume itself, and decide whether it really is Python...



It's a kind of impossible magic, to make something that eats itself, but it's a very old concept.

Here is the ouroboros, the snake that eats its own tail.

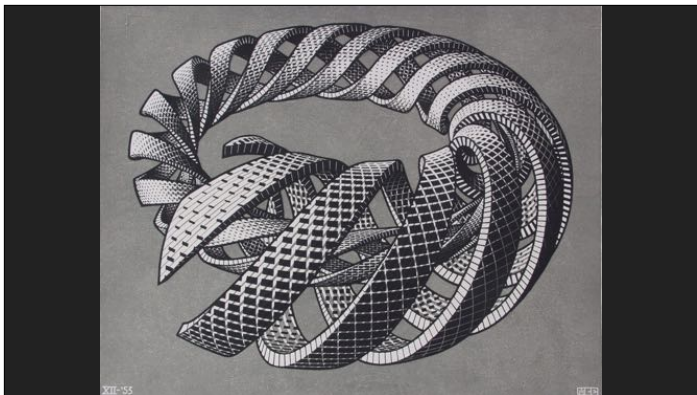
It visits scientists in their dreams, as you will know if you have studied chemistry: August Kekule's solution for the problem of the benzene molecule came to him in a reverie, after years of studying carbon-carbon bonds: a dream of snakes holding their own tails.

But the ouroboros is also an ancient symbol of renewal and regeneration in many cultures.

Programmers as much as artists are fascinated by what happens when a self-reflecting process loops, but I think that programmers are particularly lucky, because we have the perfect tools with which to explore this.

The impossible magic in this is not that the snake simply bites its own tail. Any half-witted dog could do that. The magic lies in something that eats itself and sustains itself, from which something emerges, something new, that wasn't there before.

It happens in Ulises Carrión. Something new comes out of the loop; not just something new, but a **new kind of thing altogether**. This is the *poiësis* we're in search of.



Here's another representation of the ouroboros, this time by M.C. Escher, who was also fascinated by loops and self-reference.

In this wood-block print, some interesting things are going on.

Where does this loop begin?

How many loops is it? Is it one loop? But then it's made out of four distinct bands, that spiral around each other. Does that make it four loops, or even five loops?

It's actually looping on two different levels, and you have to choose which loops you see, or hold in your mind when you look at it.

There's a *hierarchy* of looping here; the looping crosses the hierarchy.

HIERARCHIES OF SELF-REFERENTIAL LOOPS

And now we can bring in *hierarchy*, the third of our key programming concepts.



Here's Douglas Hofstadter, one of the most interesting thinkers you could ever hope to read a book by.

He's interested in hierarchies, multiple, tangled hierarchies, of self-referential loops.

His best-known book, *Gödel, Escher, Bach*, explores art, music, logic and consciousness. It's an utterly remarkable book, driven by a fascination with loops and self-reflecting processes.

In the book, he's in search of the ultimate in self-reflection or self-reference, our own human self-consciousness, subjectivity itself.

The book itself loops, recurses, reflects itself and its structure. Ultimately, Hofstadter is interested in what *emerges* from simple processes - loops.

By the way, it's a work whose humour and sense of aesthetics will appeal very naturally to programmers, because Hofstadter certainly has the soul of a programmer.

HOFSTADTER'S LAW

IT ALWAYS TAKES LONGER THAN YOU EXPECT, EVEN WHEN YOU TAKE INTO ACCOUNT HOFSTADTER'S LAW.

Douglas Hofstadter

Here's Hofstadter's Law.



Hofstadter is interested in the *emergent properties of systems*.

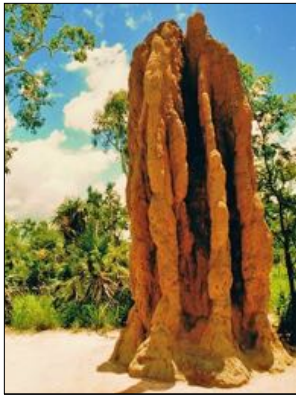
An emergent property is one that arises in a system, but can't be found anywhere in its components.

Sometimes, the property is *geometrical order*, as in the example above where water erosion has produced this remarkable effect.

The structure you see is not a structure in the rock itself. There's no hexagonal structural arrangement in the rock. You won't find hexagonal shapes anywhere in the rock.

What you see is a repeated effect of water on the rock; water that itself is affected by the rock that it affects... Together, the water and the rock form a system, and though the system is composed of its parts, the parts, lower in the system's hierarchy are themselves acted upon by the system that they are part of.

The hexagonal shapes *emerged* - unexpectedly - from the system.

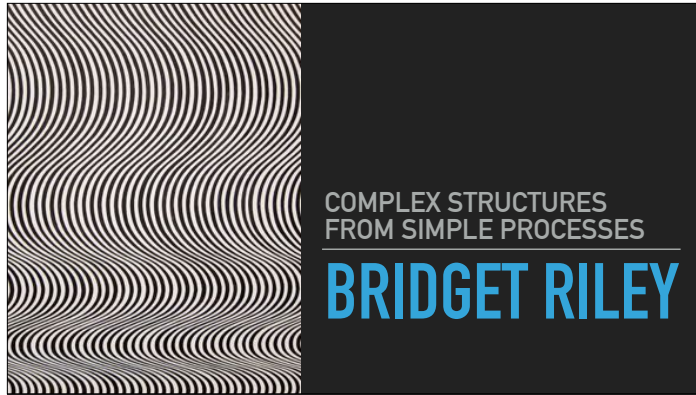


COMPLEX STRUCTURES
FROM SIMPLE PROCESSES

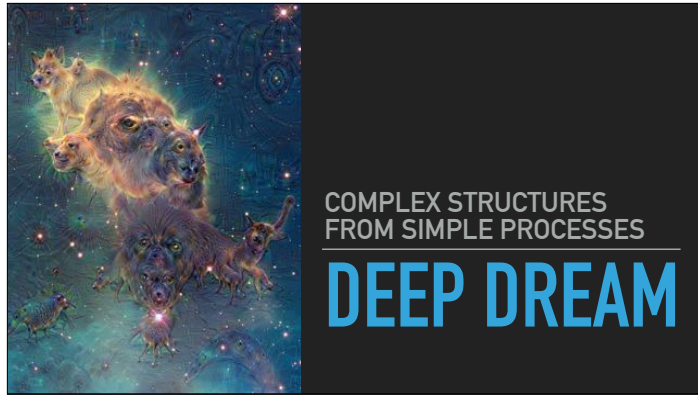
INSECT COLUMNS

Another example in nature might be an ant column.

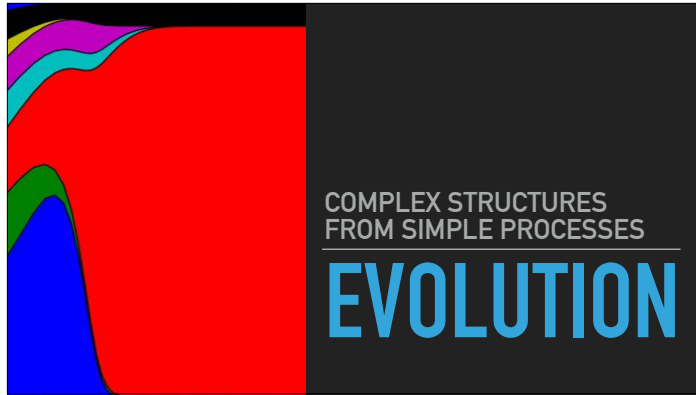
“Column organisation” is nowhere to be found in the behaviour of individual ants, and cannot even be predicted from it, but yet, in the system of ant interactions, a column, a new system that didn’t exist before, *emerges*.



Emergence can be found in art, as in the work of Bridget Riley, where simply repeatedly following a line produces, through the natural free play of the activity, something that is not in any of its components. The pattern that is the whole *emerges*.



Or Google's Deep Dream, that uses iterating pattern searching and generation algorithms, matching parts to wholes, crossing levels of hierarchy, and from which new levels of significance once again *emerge*.



Evolution - natural selection in nature - itself can be considered an *emergent property*, something that comes out of basic chemical or biological processes.

And importantly, because there are multiple levels of **hierarchy** in these systems, other properties in nature are in turn themselves emergent features of evolution.

There are numerous behaviours observed in nature whose explanations have been sought in emergence. *Co-operative* behaviour is one of those. Why is there co-operation within and even between species, when it would seem that at the level of individuals there is no benefit in co-operation?

The graph above is from a Python library, Axelrod, and it shows the success over time of successive generations of strategies in a tournament of the iterated prisoner's dilemma.

THE ITERATED PRISONER'S DILEMMA

- ▶ search for the secret of co-operation in evolution
- ▶ first tournament held in 1980
- ▶ selfish strategies did poorly
- ▶ implemented as a Python library in 2015 by Vincent Knight, Cardiff University
- ▶ <http://axelrod.readthedocs.org>
- ▶ <https://github.com/Axelrod-Python>

The iterated prisoner's dilemma is a search for the secret of co-operation in evolution.

The first tournament was held in 1980, by Robert Axelrod, a political scientist.

Perhaps surprisingly, selfish strategies in that tournament did poorly.

Axelrod was implemented as a Python library in 2015 by Vincent Knight at Cardiff University.

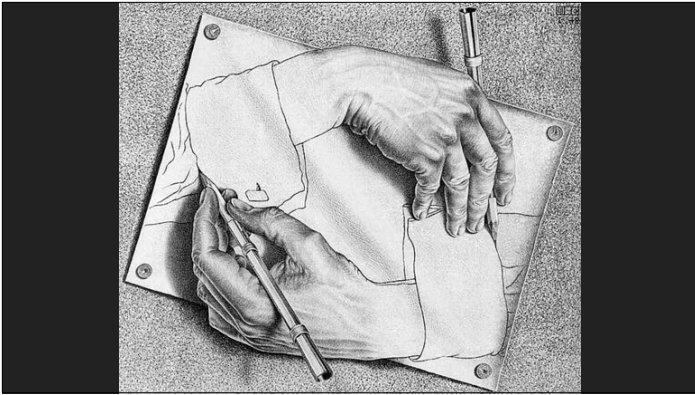
- <http://axelrod.readthedocs.org>
- <https://github.com/Axelrod-Python>

axelrod.readthedocs.org



Hofstadter argues that consciousness is an emergent property, that arises from the systems in the brain.

We discussed **loops**, and then the magic that seems to follow when loops become **self-referential**, and something new springs out of them; Hofstadter adds **hierarchy** to this, so that we have loops within loops, loops at different levels in the system.



Hofstadter's thesis is that hierarchies of self-reflecting or self-similar loops, that repeat themselves at different levels, lie at the heart of cognition and consciousness; that the human brain's neurological processes are themselves based on loops, self-reference, logic and *play*.

Consciousness, he says, is an emergent property - it doesn't exist in the neurons of the brain. We won't find it by looking at the components, we need instead to understand the system, what happens in its loops.

Is this the origin of consciousness? Are self-referring hierarchies of loops the secret of cognition?

Neurons and electrical signals in the brain are not consciousness, they are merely physical processes, and we will never find consciousness in them, however hard we look. Consciousness won't be found there, because it belongs to a different dimension from those things.

But can these loops at these lower neural, electrical, chemical levels in the system produce something that's beyond themselves, something that belongs to another dimension, just like the three dimensional hands that not only have appeared out of that lower two-dimensional level, but now, in turn, feed back both into that lower level, and into each other?

Can it be a system that nourishes itself, apparently impossibly, like the snake that eats its own tail?

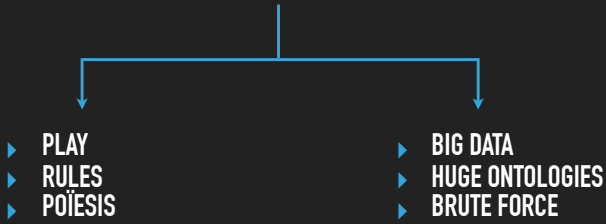
Hofstadter thinks so; in fact, one of his later books is actually called *I am a strange loop* - he argues that the *I* of consciousness is this *process* of looping and self-referring across hierarchies.

THE QUEST FOR ARTIFICIAL INTELLIGENCE



Now we are talking about intelligence and programming in a very different way from the one that has followed Turing; we are talking in ways that do, I think, make it possible to ask meaningful questions about whether machines could think.

THE QUEST FOR ARTIFICIAL INTELLIGENCE

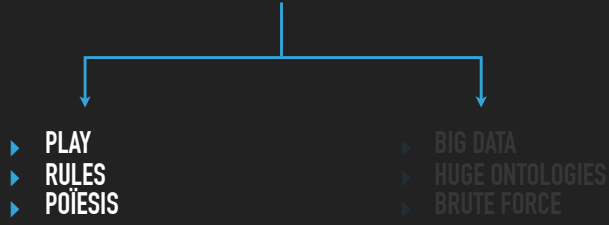


It raises the possibility of an approach into cognition research and artificial intelligence that's quite unlike some of the ones we see at the moment, that are premised on brute force, big data, huge ontologies.

Instead, it begins with the simplest of tools, the ones that programmers in fact enjoy playing with, and brings in ideas like play, analogy-making, processes, rules - *poïesis*.

It approaches the problem of consciousness as *poïesis*, as a process that's at the same time a transformative activity, that brings forth something new out of the world. It's intellectually compelling, and the opportunities for exploring it are within the reach of even the most novice Python programmer.

THE QUEST FOR ARTIFICIAL INTELLIGENCE



THE QUEST FOR ARTIFICIAL INTELLIGENCE

It gives us a way to grasp this quest that seems very rich and valuable and illuminating, and rich for exploration - especially by us, by programmers.

THE QUEST FOR INTELLIGENCE

It does something even more important: I think it gives us a way to look at and understand *our own* intelligence, our consciousness.

The question of whether Hofstadter is right, in the end, isn't that important.

REFERENCES

- ▶ [textarc.org](#)
- ▶ Ulises Carrión
"Hamlet for two voice"
"First Spanish lesson"
- ▶ Oulipo
- ▶ Ouroboros
- ▶ Axelrod, the Iterated Prisoner's Dilemma
- ▶ Douglas Hofstadter: *Gödel, Escher, Bach*



It doesn't matter for us whether Hofstadter's on the right track about the nature of consciousness, and cognition.

These ideas are so beautiful, powerful and compelling that we should wrestle with them.

Certainly, it seems more beautiful and elegant than the approach that has produced things like the chatbot Tay, by Microsoft, that after 24 hours on Twitter turned into a Hitler-loving sex-maniac.

The really important and interesting things is to have these ideas and exchanges, and to think about these things.

And something that delights me personally, that makes me very happy to be in the company of programmers, is that I find that the poets and writers and artists that fascinate me, and the things that fascinate me in programming, come back, somehow, in another pleasing circle, to the philosophical questions that have been with me for decades.

THANK YOU

ANY QUESTIONS?

DANIELE PROCIDA

- ▶ daniele.procida@divio.com
- ▶ EvilDMP on IRC, GitHub, Twitter etc
- ▶ "Documentation-driven development"
Thursday: 14.00

