

Communication in Computer Science

**An invitation to reflect
on the notion of **definition****

Olivier Danvy

version of 02 Sep 2015 at 20:30

Definition(?) of a compiler

A compiler's primary function is to compile, organize the compilation, and go right back to compiling.

Definition(?) of a compiler

A compiler's primary function is to compile, organize the compilation, and go right back to compiling. It compiles basically only those things that require to be compiled, ignoring things that should not be compiled.

Definition(?) of a compiler

A compiler's primary function is to compile, organize the compilation, and go right back to compiling. It compiles basically only those things that require to be compiled, ignoring things that should not be compiled. **The main way a compiler compiles, is to compile the things to be compiled until the compilation is complete.** (Ha, ha, found on the web.)

Unsatisfying definition, huh?

Unsatisfying definition, huh?

(As usual, laugh as you want to be laughed at.)

Unsatisfying definition, huh?

Why are we dissatisfied with this definition?

Unsatisfying definition, huh?

Why are we dissatisfied with this definition?

- It is circular
 (“a compiler compiles”).

Unsatisfying definition, huh?

Why are we dissatisfied with this definition?

- It is circular
 (“a compiler compiles”).
- It emphasizes secondary issues
 (“only compiles what needs to be compiled”).

Unsatisfying definition, huh?

Why are we dissatisfied with this definition?

- It is circular
("a compiler compiles").
- It emphasizes secondary issues
("only compiles what needs to be compiled").
- It is redundant
(i.e., repetitive).

On the other hand

Suppose that you already know
the meaning of “to compile.”

Then the definition is not so bad, is it?

Definition(!) of a compiler

A compiler's primary function is to compile, organize the compilation, and go right back to compiling. It compiles basically only those things that require to be compiled, ignoring things that should not be compiled. The main way a compiler compiles, is to compile the things to be compiled until the compilation is complete. (Red is bad.)

Still unsatisfying

The definition says

- what a compiler does (namely: it compiles),
and
- how it works

but it does not say **what it is**.

So what is a compiler?

A compiler is **a program**.

And what does a compiler do?

A compiler **translates programs**
from one programming language
to another programming language.

How is a compiler organized?

...see any compiler course or book...

So what is a definition?

It is an informative answer
to the question “what is this?”

We express a reality by saying what “this” is:
that is the goal of a definition.

To say what “this” is

We use words.

So be lucid about words:

- Do they have a **single** meaning?
- Do they have **multiple** meanings?
- Do they support analogies?

Basically

Proceed

from what is known

to what is (yet) unknown.

The variety of definitions

- Synonymous definitions
- Lexical definitions
- Definitions by genus and species
- Definitions by enumeration
- Non-definitions
- Self-referential definitions
- Aristotle's 4 causes

Synonymous definitions

A concept is already known.

A new word is introduced.

Lexical definitions

Actual or normative meanings of a word
(typically in a dictionary).

Definitions by **genus and species**

1. Specifies the type of the thing being defined.
2. Lists distinguishing properties.

Used by botanists, geologists, etc.

Definitions by **genus and species**

1. Specifies the type of the thing being defined.
2. Lists distinguishing properties.

Used by botanists, geologists, etc.

Examples: *theobroma cacao* (chocolate)

Definitions by **genus and species**

1. Specifies the type of the thing being defined.
2. Lists distinguishing properties.

Used by botanists, geologists, etc.

Examples: *theobroma cacao* (chocolate)

homo sapiens lycanthropus (werewolf)

Definitions by enumeration

Listing the things that are meant.

Major French car makers are
Renault, Peugeot, Citroën, etc.

Example of non-definition: **by extension**

Pointing at particular examples
of the things denoted by the word.

A parent teaches a child the meaning of 'chair.'

Example of non-definition: **by osmosis**

- “The well-known elephant test.
It is difficult to describe,
but **you know it when you see it.**”
– Lord Justice Stuart-Smith

Example of non-definition: **by osmosis**

- “The well-known elephant test.
It is difficult to describe,
but **you know it when you see it.**”
– Lord Justice Stuart-Smith
- What is jazz?
“**Man, if you gotta ask, you’ll never know.**”
– Louis Armstrong

Example of non-definition: **by osmosis**

- “The well-known elephant test.
It is difficult to describe,
but **you know it when you see it.**”
– Lord Justice Stuart-Smith
- What is jazz?
“**Man, if you gotta ask, you’ll never know.**”
– Louis Armstrong
- “**Aw, you know what I mean.**”

Example of non-definition: **by negation**

Knowing **what it is not**
does not imply
knowing **what it is**.

Example of non-definition: **by negation**

Knowing **what it is not**
does not imply
knowing **what it is.**

- **“It’s not bad.”**

Example of non-definition: **by negation**

Knowing **what it is not**
does not imply
knowing **what it is.**

- “**It’s not bad.**” (Or worse: “**It’s not half bad.**”)

Example of non-definition: **by negation**

Knowing **what it is not**
does not imply
knowing **what it is.**

- “It’s not bad.” (Or worse: “It’s not half bad.”)
- “Je ne sais quoi.”

Example of non-definition: **by assertion**

- “The best thing since sliced bread.”
- “You will like it.”
- “Trust me here.”

Circular definitions

For example:

I am right because I am right.

Circular definitions

For example:

I am right because I am right.

Question: is the following definition circular?

Circular definitions

For example:

I am right because I am right.

Question: is the following definition circular?

“Fashion is what goes out of fashion.”

– Coco Chanel

Inductive definition

Base case:

With something atomic and pre-defined,
you define something.

Inductive case:

Using something you already have defined,
you define something composite (and bigger).

Example of inductive definition

The graph of the factorial function:

- Here is a pair of numbers: $(0, 1)$.
- Given a pair of numbers (n, f) ,
here is a new one: $(n + 1, (n + 1) \times f)$.

Recursive definition

Base case:

Given something atomic,
you define something.

Inductive case:

Given something compound,
and using your definition on the components,
you define something.

Example of recursive definition

- The factorial of 0 is 1 .
- The factorial of $n + 1$ is $n + 1$ times the factorial of n .

Programming

Your recursive programs
typically operate
over inductively constructed data:
they are **structurally recursive**.

Well-founded self-referential definitions

Watch for **well-founded orderings**
when defining something
inductively or recursively.

No “turtles all the way down.” (Russell)

Example of problematic definition

An onion is

- either nothing
- or a layer with an onion inside.

Example of problematic definition

An onion is

- either nothing
- or a layer with an onion inside.

→ But can we define something out of nothing?

Talking about nothing

- an empty list
- an empty string
- an empty array
- a parameterless procedure
- etc.

Talking about nothing

- an empty list
- an empty string
- an empty array
- a parameterless procedure
- etc.

Are zero things nothing?

Aristotle's 4 causes

Material cause: what is the object of discourse?
what are we talking about?

Efficient cause: what is done to it / who does it?

Formal cause: which methods are used to do that?

Final cause: towards what use / what goal?
why is this done?
what for / pourquoi / hvorfor?

Example #1

To make a dish,
a cook operates
over the ingredients
according to the recipe.

Example #1

To make a dish,
a cook operates
over the ingredients
according to the recipe.

← final cause

Example #1

To make a dish, ← final cause
a cook operates ← efficient cause
over the ingredients
according to the recipe.

Example #1

To make a dish, ← final cause
a cook operates ← efficient cause
over the ingredients ← material cause
according to the recipe.

Example #1

To make a dish,	← final cause
a cook operates	← efficient cause
over the ingredients	← material cause
according to the recipe.	← formal cause

Example #2

Using a toothbrush,
Michael Scofield unhinges
the prison door
to escape.

Example #2

Using a toothbrush,

← formal cause

Michael Scofield unhinges

the prison door

to escape.

Example #2

Using a toothbrush, ← formal cause

Michael Scofield unhinges ← efficient cause

the prison door

to escape.

Example #2

Using a toothbrush, ← formal cause

Michael Scofield unhinges ← efficient cause

the prison door ← material cause

to escape.

Example #2

Using a toothbrush, ← formal cause

Michael Scofield unhinges ← efficient cause

the prison door ← material cause

to escape. ← final cause

Example #3

My program computes

a function

over discrete data

to predict the weather.

Example #3

My program computes ← efficient cause
a function
over discrete data
to predict the weather.

Example #3

My program computes

← efficient cause

a function

← formal cause

over discrete data

to predict the weather.

Example #3

My program computes

← efficient cause

a function

← formal cause

over discrete data

← material cause

to predict the weather.

Example #3

My program computes ← efficient cause
a function ← formal cause
over discrete data ← material cause
to predict the weather. ← final cause

Example #4

My recognition

algorithm

for infinite strings

operates in linear time.

Example #4

My recognition

← final cause

algorithm

for infinite strings

operates in linear time.

Example #4

My recognition

← final cause

algorithm

← formal cause

for infinite strings

operates in linear time.

Example #4

My recognition ← final cause

algorithm ← formal cause

for infinite strings ← material cause

operates in linear time.

Example #4

My recognition ← final cause
algorithm ← formal cause
for infinite strings ← material cause
operates in linear time. ← efficient cause

Example #4

My recognition ← final cause
algorithm ← formal cause
for infinite strings ← material cause
operates in linear time. ← efficient cause

Linear time! Nice! And is it efficient?

Exercise #1

Write a few more sentences
where the 4 causes are visible.

Aristotle's 4 causes

Sometimes, one cause is enough.

Material cause

Q. What is this rock?

A. Granite.

Formal cause

Q. What is this mathematical figure?

A. A triangle.

Efficient cause

Q. What is this painting?

A. A Picasso.

Final cause

Q. What is a pair of compasses?

A. It is used to draw circles.

Summary

- A definition is an informative answer to the question “what is this?”
- It proceeds from the known to the unknown.

Exercise #2

Compose a definition with your eyes open:

- by genus and species
(i.e., by type and differences),
- using (some or all of) the 4 causes.

Exercise #3

Write down the four causes of a compiler.