Generative Recursion

19 November 2020
The recursive functions we’ve been writing so far have used *structural recursion*.

Each recursive call takes some sub-piece of the data, e.g.,

- going through a list, we keep taking the *rest* of the list
- processing a natural number $n$, we take $n-1$

The *structure* of the data determines the structure of the recursive function.

Today we’ll look at *generative recursion*.

This is more general – the recursive cases are *generated* based on the problem to be solved.

Generative recursion can be harder because neither the base nor recursive cases follow from the data definitions.
Template for generative recursion

;; Termination: Say why you’re guaranteed to reach the base case.
(define (problem-solver d)
  (if (trivial? d)
      ;; Base case: The computation is in some way trivial.
      (... d)
      ;; Recursive case: Transform d to generate new problems.
      (combiner (... d) (problem-solver (transform d)) ...))))
Termination

When we’ve written a function using structural recursion, the depth of recursion has been limited by the size of the input, e.g.,

\[
\begin{align*}
\text{(sum-list (list 3 6 5 4))} \\
\rightarrow (+ 3 \text{(sum-list (list 6 5 4))}) \\
\rightarrow (+ 3 (+ 6 \text{(sum-list (list 5 4))}))
\end{align*}
\]

For generative recursion, \textit{termination can’t be taken for granted}; we’ll need to argue why we know our computation will end.
Example: Fractals
“A fractal is a way of seeing infinity.”

Benoit Mandelbrot
Design a function that consumes a number and produces a *Sierpiński triangle* of that size:

Start with an equilateral triangle with side length $s$:

Inside that triangle are three more Sierpinski triangles:

And inside of each of those … and so on.

Producing something that looks like this:
The diagram shows a triangle with side lengths labeled as 's' and 's/2'. The area and perimeter of such a triangle can be calculated using these side lengths.

To find the area, you can use the formula for the area of an equilateral triangle, which is $A = \frac{\sqrt{3}}{4} \times (s^2)$.

To find the perimeter, simply add all the side lengths together: $P = s + s + \frac{s}{2} = \frac{3s}{2}$.
[sierpinski-triangle.rkt]
How do we know that this function won’t run forever?

Three-part termination argument:

**Base case:** \((\leq s \ \text{CUTOFF})\)

**Reduction step:** \((/ s 2)\)

**Argument that repeated application of reduction step will eventually reach the base case:**

As long as the cutoff is > 0 and s starts \(\geq 0\), repeated division by 2 will eventually be less than the cutoff.
Design a function \texttt{s-carpet} to produce a Sierpiński carpet of size \( s \):
Design a function $s$-carpet to produce a Sierpiński carpet of size $s$:

There are eight copies of the recursive call positioned around a blank square.
[sierpinski-carpet.rkt]
How do we know that this function won’t run forever?

Three-part termination argument:

*Base case:* \( (\leq s \text{ CUTOFF}) \)

*Reduction step:* \( (/ s 3) \)

*Argument that repeated application of reduction step will eventually reach the base case:*

As long as the cutoff is > 0 and s starts \( \geq 0 \), repeated division by 3 will eventually be less than the cutoff.
Acknowledgments

This lecture incorporates material from:

Matthias Felleisen
Robert Bruce Findler
Matthew Flatt
Gregor Kiczales
Shriram Krishnamurthi
Marc Smith
U. Waterloo CS135