Recursion

You will learn what is recursion as well as how and when to use it in your programs.

James Tam

What Is Recursion?

"the determination of a succession of elements by operation on one or more preceding elements according to a rule or formula involving a finite number of steps" (Merriam-Webster online)

What This Really Means

Breaking a problem down into a series of steps. The final step is reached when some basic condition is satisfied. The solution for each step is used to solve the previous step.

Definition For Philosophy

"…state of mind of the wise man; practical wisdom…" [1]

See Metaphysics

1 The New Webster Encyclopedic Dictionary of the English Language

Metaphysics

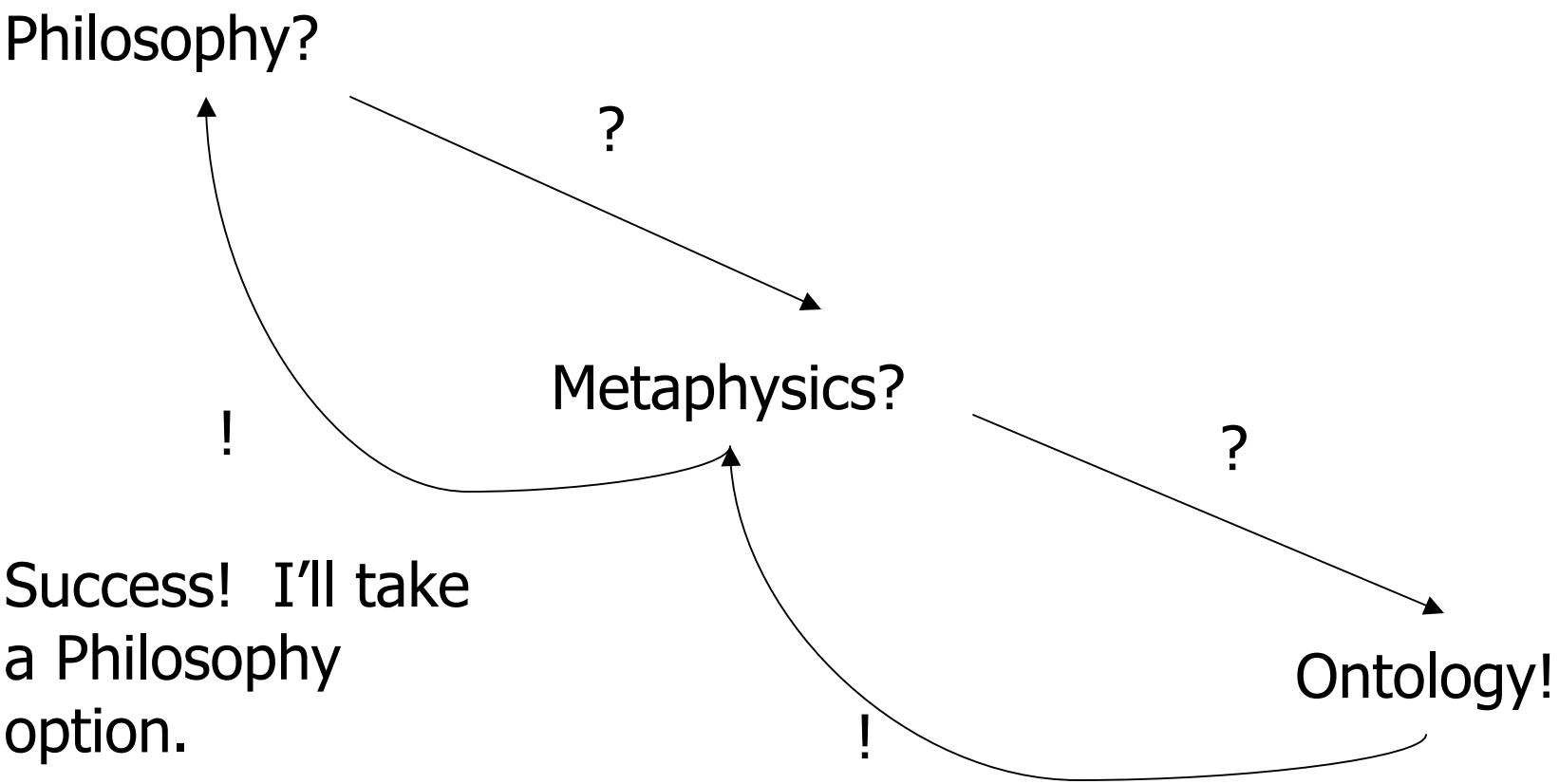
"…know the ultimate grounds of being or what it is that really exists, embracing both psychology and ***ontology***." [2]

2 The New Webster Encyclopedic Dictionary of the English Language

Result Of Lookup (Possibility One: Success)

I know what Ontology means!

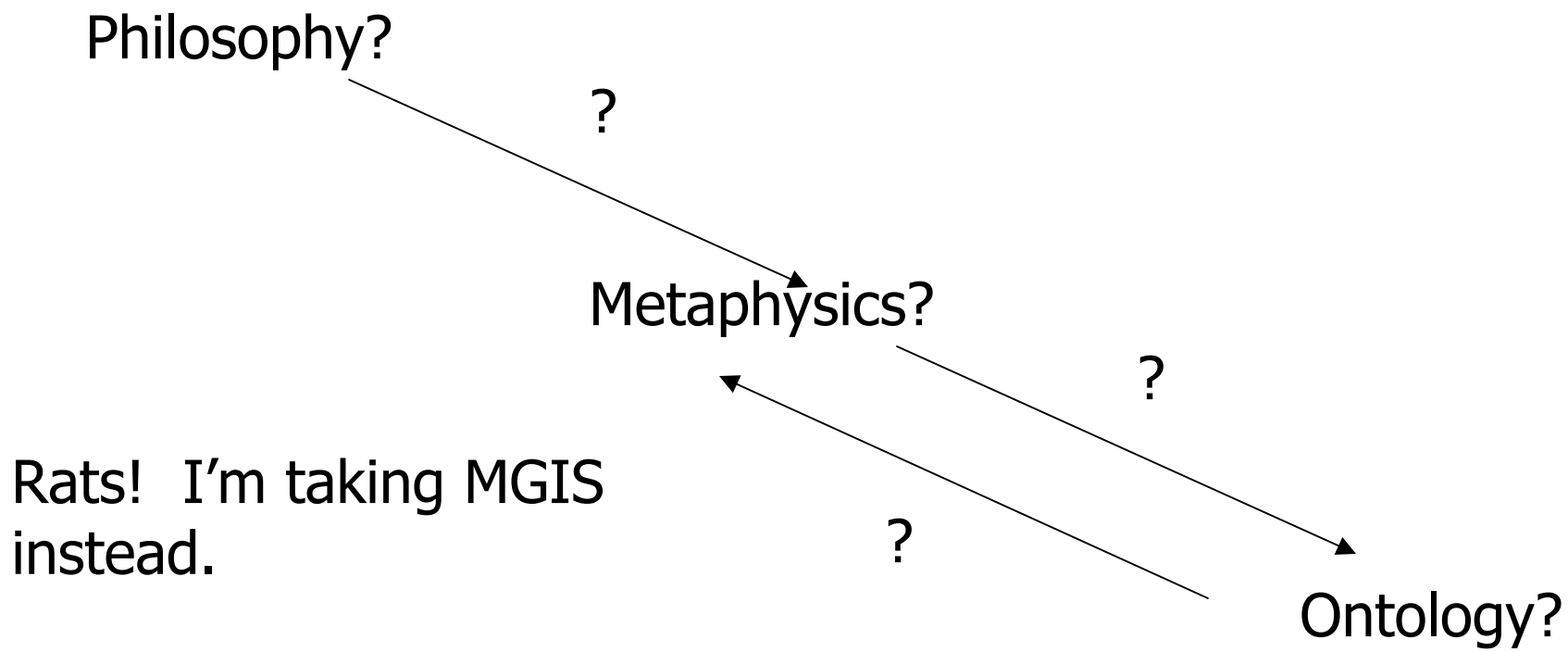
Result Of Lookup (Possibility One)



Result Of Lookup (Possibility Two: Failure)

I don’t have a clue.

Result Of Lookup (possibility two)



Rats! I'm taking MGIS instead.

Ontology

"…equivalent to metaphysics."[3]

3The New Webster Encyclopedic Dictionary of the English Language

Looking Up A Word

If (completely understand a definition)

Return to previous definition (using definition that's understood)

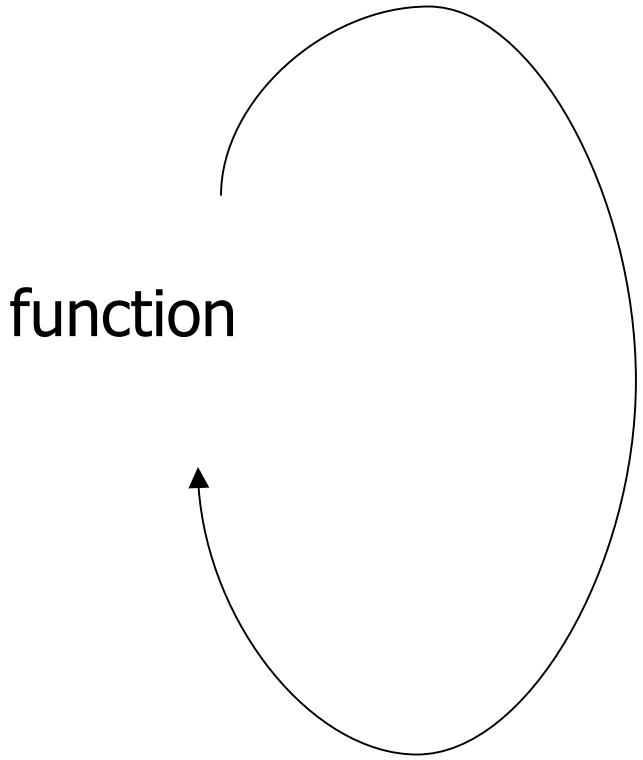
Else

lookup (unknown word(s))

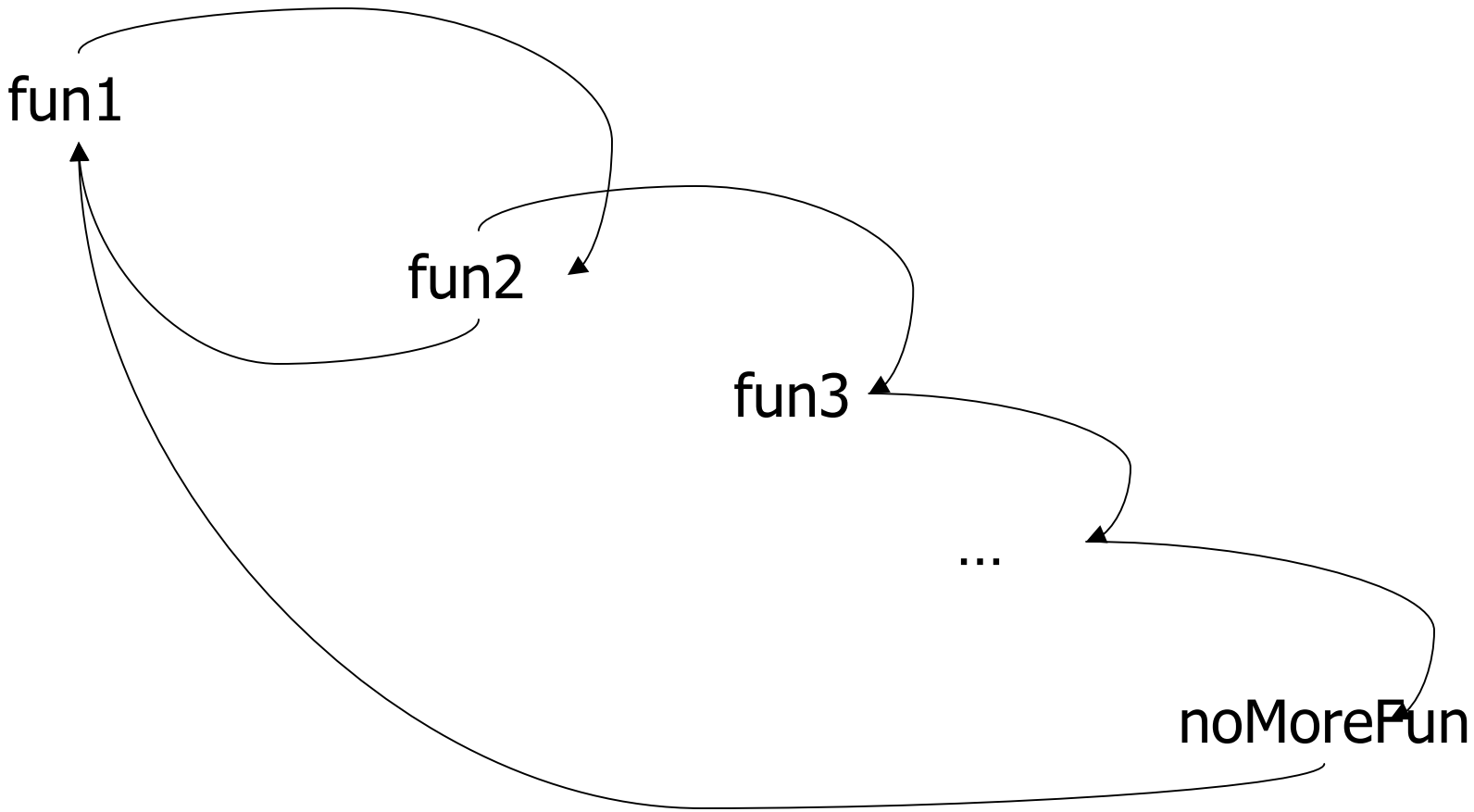
Recursion In Programming

"A programming technique whereby a function or procedure calls itself either directly or indirectly."

Direct Call



Indirect Call



Requirements For Recursion

1) Base case
2) Progress is made (towards the base case)

Counting Example

Write a program that will compute the sum of the first n positive integers.

e.g. n = 3, sum = 3 + 2 + 1 = 6

sum (3) = 3 + sum (2) = 3 + 3 = 6

sum (2) = 2 + sum (1) = 2 + 1 = 3

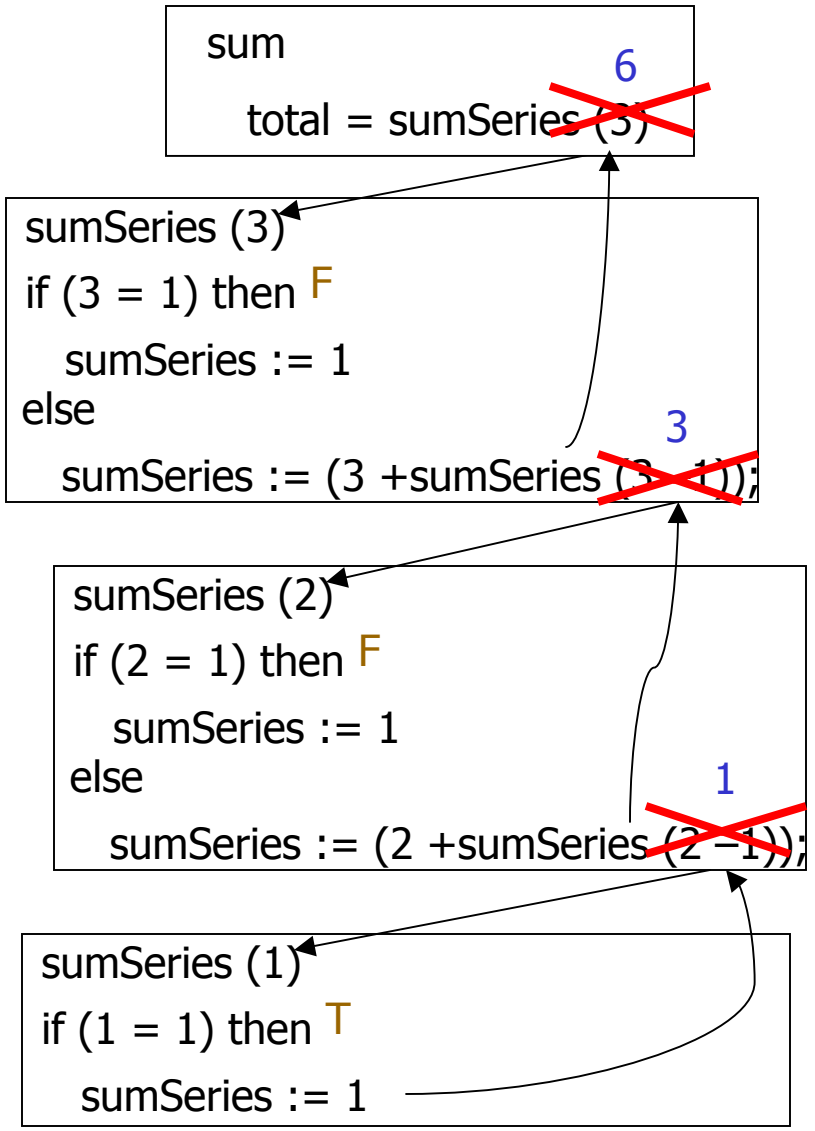
sum (1) = 1

Example Program

```
program sum (input, output);
var
   lastNumber : integer;
   total      : integer;

function sumSeries (no : integer): integer;
begin
   if (no = 1) then
      sumSeries := 1
   else
      sumSeries:= (no + sumSeries (no - 1));
end;

begin
 write('Enter the last number in the series :');
   readln(lastNumber);
   total := sumSeries(lastNumber);
   writeln('Sum of the series from 1 - ',
        lastNumber, ' is, ', total);
end.
```



When To Use Recursion

When a problem can be divided into steps

The result of one step can be used in a previous step

All of the results together solve the problem

When To Consider Alternatives To Recursion

When a loop will solve the problem just as well

Drawbacks Of Recursion

Function calls can be costly

- Uses up memory
- Uses up time

Benefits Of Using Recursion

Simpler solution that's more elegant (for some problems)

Easier to visualize solutions (for some people)

Common Pitfalls When Using Recursion

No base case

No progress towards the base case

Using up too many resources (variable declarations) for each function call

No Base Case

```
function sumSeries (no : integer): integer;
begin
    sumSeries:= (no + sumSeries (no - 1));
end;
```

No Progress Towards Base Case

```
function sumSeries (no : integer): integer;
begin
   if (no = 1) then
      sumSeries := 1
   else
      sumSeries:= (no + sumSeries (no));
end;
```

Undergraduate Definition Of Recursion

Word: **re·cur·sion**

Pronunciation: ri-'k&r-zh&n

Definition: See recursion

Summary

Description of recursion

Real world example

Trace of a recursive Pascal program

Benefits and drawbacks of using recursion

When to use recursion and when to consider alternatives

What are the potential pitfalls of using recursion

Alternative definition of recursion