Rebecca Douglas

Ms. Carter

Information Management

7 October 2016

<center>Quines: Useless Programs Within Programs</center>

A "quine" (or "selfrep") is a computer program which prints its own listing. This may sound either impossible, trivial or completely uninteresting, depending on your temperament and your knowledge of computer science. Actually, it is possible, and there are some interesting ideas involved (in particular, writing a quine is not a hack that only works because the programming language has certain nice properties — it is a consequence of the general so-called "fixed-point" theorem, itself an instance of Cantor's ubiquitous diagonal argument).

Quines are named after the American mathematician and logician Willard van Orman Quine (lived 1908 – 2000) who introduced the concept. This page is dedicated to his memory (madore.org, 2005).

Example on Basic:

10 LIST

Example on Fortran:

SOURCE TYPE

<center>Do they really exist?</center>

Despite the simple task formulation spending some time at its decision, the question arises: Do they really exist? The answer is: Yes!

Moreover, Quines exists in any language that can output an arbitrary computed line! This idea was first described by Paul Bratlie. At first, a Quine was considered a program written in the language of Atlas AutoCode by Hamish Dewar.

And here is the example in the modern language Javascript:

function f(){alert(f.toString()+"f();");}f();

Pascal:

program ritchie; const ma = 12; a: array[1..ma] of string = (

'var j : integer;',

'procedure p (s : string; c : boolean); var i : integer; begin',

' if c then write ("''''");',

' for i := 1 to length (s) do',

' if c and (s[i]="''''") then write ("''''''") else write (s[i]);',

' if c then begin',

'   if j < ma then writeln ("''",",") else writeln ("''");")',

' end else writeln; end ;',

'begin writeln',

' ("program ritchie; const ma=", ma, "; a: array[1..ma] of string = (");',

' for j := 1 to ma do p (a[j], true);',

' for j := 1 to ma do p (a[j], false); end.');

var j : integer;

```
procedure p (s : string; c : boolean); var i : integer; begin

  if c then write ("'");

  for i := 1 to length (s) do

  if c and (s[i]="'") then write ("''") else write (s[i]);

  if c then begin

    if j < ma then writeln ("',') else writeln ("');')

  end else writeln; end ;

begin writeln

  ('program ritchie; const ma=', ma, '; a: array[1..ma] of string = (');

  for j := 1 to ma do p (a[j], true);

  for j := 1 to ma do p (a[j], false); end.
```

(nyx.net, 1999)

So we are convinced that such programs exist, and now for some theory about how to make them.

Intuitively, it is necessary to output the value of the variable in which the partial program code is stored. Why partial? Because the assignment itself is variable, too, and should be in the value of the variable. In other words, the value of the variable is to be copied itself, which is why there is an infinite recursion. Unpleasant moment.

To remedy the situation, we will not make the fact of variable assignment:

```
char c[]="char c[]=;";
```

Now there is a problem inserting the line in the output with string. Here we recall the printf and its power:

```
printf("%.2s", c+1);
```

And finally, here's a couple of exotic examples:

PHP:

```
<?php $a="$"; $e='printf(\'<?php %sa="$"; %se=%s; eval(%se);\', $a, $a, var_export($e, 1), $a);'; eval($e);
```

SQL:

```
SELECT REPLACE(REPLACE('SELECT REPLACE(REPLACE("$",CHAR(34),CHAR(39)),CHAR(36),"$") AS Quine',CHAR(34),CHAR(39)),CHAR(36),'SELECT REPLACE(REPLACE("$",CHAR(34),CHAR(39)),CHAR(36),"$") AS Quine') AS Quine
```

Python:

```
q='"';s='q=%s;s=;print(s[:3]+q+s[3:7]+q+s+q+s[7:])';print(s[:3]+q+s[3:7]+q+s+q+s[7:])
```

Perl:

```
$_=q{$_=q{x};s/x/$_/;print};s/x/$_/;print
```

Thanks for your attention!

References

"Quines (self-replicating Programs)." Quines (self-replicating Programs). N.p., 18 Oct. 2005.

Web. 10 Aug. 2016.

"The Quine Page." (self-reproducing Code). N.p., 7 Oct. 1999. Web. 10 Aug. 2016.