Turing's Dots

Lance Fortnow

Northwestern University

Abstract

Discussion of Turing's paper "The Use of Dots as Brackets in Church's System," *J. Symbolic Logic* 7, pp 146-156 (1942)

Alan Turing's rarely-cited paper "The Use of Dots as Brackets in Church's System"defines a new notation for Church's λ-calculus using what Turing calls dots, the symbols "." and ":".

Turing states that he intended to make use of this notation in forthcoming papers entitled "Some theorems about Church's systems" and "The theory of virtual types". I can find no record of those later papers. Likely Turing's activities during World War II curtailed his scientific research and his interests shifted after the war.

Even though this paper had little to no direct influence to logic and computer science, it shows once again Turing's ability to reason about important issues in computer science before there were digital computers to reason about. In this case, Turing essentially studies an important aspect of programming languages, a syntax for trees.

To understand the paper consider precedence operations on formulas such as

$$4x - 3y^2 + 7$$

To parse this equation we need to know that exponentiation has precedence over multiplication which has precedence over addition and subtraction. Operations with the same precedence occur left to right. The expression above can be written with parenthesis as

$$(((4x) - (3(y^2))) + 7)$$

Turing creates virtual precedence operations he calls dots and shows how it can replace balance parentheses used by Church ("A Formulation of the Simple Theory of Types," *J. Symbolic Logic* 5, pp 56-68 (1940)).

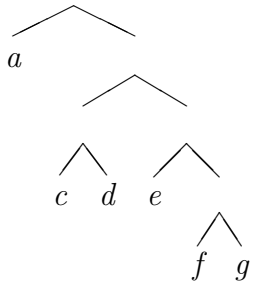
Zero dots has highest precedence, then one dot (.), two dots (:), three dots (:.) etc. At the same precedence, application is done left to right. So the expression

$$a : .cd : e.fg$$

is evaluated as

$$(a((cd)(e(fg)))).$$

Though Turing doesn't discuss trees, both notations describe binary trees. In this case



Every binary tree can be expressed through dots or through parentheses.Turing acknowledges that for simplicity sometimes dot can be mixed with parenthesis or other precedence operators.

Today we have a common method for creating trees known as Extensible Markup Language (XML). The tree above can be described by

```
<t1>
a
<t2>
<t3> c d</t3>
<t4> e
<t5> f g </t5>
</t4>
</t2>
</t1>
```

The tags (`<t1>, <t2>...`) act like parentheses. According to Northwestern Professor Robby Findler, an expert in programming languages, no major system uses virtual precedence operations akin to Turing's dots.

The reason is modularity. Suppose we wanted to replace g with a subtree consisting of r and s. With parentheses we can do a simple replacement

$$(a((cd)(e(fg))))$$

becomes

$$(a((cd)(e(f(rs))))).$$

With dots though we have to readjust the whole formula,

$$a : .cd : e.fg$$

becomes

$$a :: cd : .e : f.rs$$

In short, Turing's dots gave him a way to think about the order of operations in a structure that was more intuitive to him to prepare him for planned future work on Church's λ-calculus. Unlike the Turing machine, the dot notation did not catch on for reasons Turing did not appreciate: that someone might want to modify the code.

Acknowledgements

Thanks to my student Arefin Huq for helping me with "breaking the code" of Turing's dot notation and to Robby Findler for helpful discussions.