Polish and Reverse Polish Notation

James T. Smith
San Francisco State University (SFSU)

A version of this article was published in the Spring 2016 issue of *Math News*, the newsletter of the SFSU Department of Mathematics and Statistics. The paragraph beginning *No bell yet* was omitted, and the figure as well. Here is the complete text; readers are invited to imagine the figure.

Reading about hand calculators or computer programming, you may have encountered these terms. What's this about?

During World War I Poland was emerging as an independent country. Polish mathematicians were determined to make it a research center of world rank. They emphasized a discipline where they were already strong: mathematical logic. During the 1920s, the logician Jan Łukasiewicz was investigating how to measure the complexity of formulas such as $(a+(b-1))\times d$ that you meet in algebra class. Does complexity have to do with the nesting of parentheses? Not directly: read on.

Łukasiewicz and his colleagues noticed that if we should always write the operation sign *before* the numbers it combines—for example, $-b1$ instead of $(b-1)$ —then we wouldn't need parentheses at all. Our formula $(a+(b-1))\times d$ would become first $(a+-b1)\times d$, then $(+a-b1)\times d$, then finally $\times+a-b1d$. Evidently, complexity stems from the number and arrangement of the operation signs, not the parentheses.

The Poles' achievements in several areas of logic quickly made them famous. They wrote many logic formulas this way in their research journal *Fundamenta Mathematicae*, probably to attract attention as much as to analyze complexity. That journal became tops in the world in its field, and this style widely known as *Polish notation*.

Not easy to read, is it! But someone noticed that something suddenly rings a bell if you always write the operation sign *after* the numbers it combines—for example, $b1-$ instead of $b-1$. With this *reverse Polish* notation, that trial formula $(a+(b-1))\times d$ would become first $(a+b1-)\times d$ then $(ab1-+)\times d$, and finally $ab1-+d\times$.

No bell yet? Find someone familiar with Hewlett–Packard hand calculators. Read this formula from left to right, say "enter" before each symbol for a number and "punch" before each operation sign. The HP user will envision the machine's main-storage display showing first the entered value of a, then b, then 1, then $b-1$ (the result of combining the previous two entries), then

Stack of plates, as in a cafeteria but staggered, with the top two plates overlaid by formulas to be combined.

$(a + (+b - 1))$ —same idea—then finally $(a + (b - 1)) \times d$. Ding: there's the answer!

This shows that reverse Polish notation is equivalent to the system you learned in algebra: whatever you do with one you can do with the other.

Here are some remarks that may intrigue you into inquiring further. Łukasiewicz was not just a mathematician: he served as a cabinet minister in the first independent Polish government, and as a professor of philosophy. Desktop calculators were introduced in the 1910 decade and this input technique developed for them soon after that; reverse Polish notation became familiar during the 1950s. The main storage unit of that sort of calculator is analogous to the pushdown stack of plates at the start of the serving lines in Polish cafeterias today; the display shows the top plate, and the operations always replace the top two plates with one containing the result. The idea of a pushdown stack is one of the most basic in computer architecture, and reverse Polish notation is the underlying idea of several very powerful computer languages, such as the *PostScript* language that a computer uses to describe this document to a printer.

James T. Smith