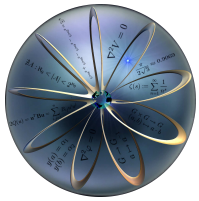

planetmath.org

Math for the people, by the people.

straight-line program

Canonical name	StraightlineProgram
Date of creation	2013-03-22 16:16:02
Last modified on	2013-03-22 16:16:02
Owner	Algeboy (12884)
Last modified by	Algeboy (12884)
Numerical id	10
Author	Algeboy (12884)
Entry type	Definition
Classification	msc 08A99
Classification	msc 20A05
Classification	msc 20-00
Related topic	word
Related topic	Word
Defines	straight-line program
Defines	SLP
Defines	SLP representation

Definition 1. *Given a set S, a* straight-line program *(SLP) is a family of functions $\mathcal{F} = \{f_i : 1 \leq i \leq m\}$*

$$f_i : S^{n+i} \to S$$

for some fixed $n \in \mathbb{N}$. An SLP is evaluated *on a tuple $(s_1, \ldots, s_n)$ by recursion in the following way: f_1 is evaluated on $(s_1, \ldots, s_n)$ as a function. The remaining evaluations are recursive. So $f_2(s_1, \ldots, s_n)$ denotes*

$$f_2(s_1, \ldots, s_n) := f_2(s_1, \ldots, s_n, f_1(s_1, \ldots, s_n)).$$

and in general

$$f_{i+1}(s_1, \ldots, s_n) := f_{i+1}(s_1, \ldots, s_n, f_1(s_1, \ldots, s_n), \ldots, f_i(s_1, \ldots, s_n)).$$

The final output $f_m(s_1, \ldots, s_n)$ is denoted $\mathcal{F}(s_1, \ldots, s_n)$. In this way we treat $\mathcal{F}$ as function from $S^n \to S$.

SLPs arrise from the multiple meansings of expressions of the sort a^n in a some algebraic structure S. First of all, one can formally treat a^n as the word $a \cdots a$ in S. Secondly this can be interpreted as the actual result of this mulitplication.

In the former meaning, actually storing a word of the form a^n as $a \cdots a$ is difficult hence it is abreviated. Other examples include words such as $a^{10}(bc)^6$ where the values of a, b, c are continually changing or even unknown. Here an SLP can encode this word in such a way that if we replace a by bc, then the resulting new word would result simply by evaluation the SLP at the input (bc, b, c) instead of (a, b, c).

In the second treatment where we whish to actually evaluate a^n and the like, we find the problem of understanding what a^n means as a program. Certainly we may have $a^4 = (aa)(aa) = a(a(aa))$ etc. However this equivalence neglects the problem of selecting a method of computing the result. Usually an efficient method is desired. An SLP developed from simple functions such as $f(x) = x^2$ and $f(x, y) = x + y$ formally address this problem.

The term *straight-line* reflects the fact that evaluating an SLP can be achieved by a program which does not branch or loop so its execution is a straight-line. It is common for SLPs to be built entirely from simple functions such as $f(x, y) = x + y$ or $f(x) = x \times x$.

Because each element f_i of an SLP is evaluated externally only on $s_1, \ldots, s_n$ and the remaining inputs come internally from previous f_j, $1 \leq j \leq i$, it is

convient to write definitions for f_i as taking inputs only from $(s_1, \ldots, s_n)$ and implicitly allowing for the use of the outputs of previous f_j's.

SLPs can be defined in contexts other than semigroups including rings, modules, and polynomials. Although they arise naturally to compress computations, they are also useful in describing smaller bounds for many combinatorial theorems related to algebraic objects.

It is possible for a function $f : S^n \to S$ to be defined equivalently by multiple SLPs and so a notion of equality of SLPs is stronger than equivalence of final outputs.

Definition 2. *Two SLPs $\mathcal{F} = \{f_i : 1 \leq i \leq m\}$ and $\mathcal{G} = \{g_i : 1 \leq i \leq k\}$ are* equivalent *if $\mathcal{F}$ and $\mathcal{G}$ can be evaluated on the same inputs and for every input $(s_1, \ldots, s_n)$, $\mathcal{F}(s_1, \ldots, s_n) = \mathcal{G}(s_1, \ldots, s_n)$. When and SLP $\mathcal{F}$ is equivalent to a trivial SLP $\{f : S^n \to S\}$ we say that $\mathcal{F}$ is an* SLP representation *of f.*

Every function $f : S^n \to S$ can be expressed as an SLP trivially by $\mathcal{F} = \{f\}$. However, this SLP is typically the least optimal for the actual evaluation of the output for a given input. This leads to a hierarchy imposed on equivalent SLPs based on their associated computational length.

Definition 3. *If an SLP represents an algebraic expression (alternatively a word in the generators) in the semigroup S then the* computational length *or simply* length *of the SLP is the maximum number of multiplications in the semigroup performed to evaluate the expression using the SLP.*

It is evident that the trivial SLP of an algebraic expression has length equal to the length of the word.