1. Introduction

Low-level programs are sometimes hand-written to facilitate efficient computing. Another situation where low-level programs are used is extensible, performance-conscious systems. Such systems exploit low-level portable programs. However, the safety of most low-level programs is not guaranteed since most low-level languages provide only inferior safety mechanisms and don't have their own type systems.

Typed assembly languages are introduced in a paper "From System F to Typed Assembly Language" (Morrisett et al., 1998).

In this article, we define a general-purpose typed assembly language which targets abstract machines. Its syntax is given in Figure 1.

r ::=		registers:
	r1 | r2 | ... | rk	general-purpose registers
ν ::=		operands:
	r	register
	i	integer
ι ::=		instructions:
	mov r ν	move
	add r ν ν	add
	sub r ν ν	subtract
	and r ν ν	logical and
	or r ν ν	logical or
	not r ν	logical not
	shl r ν ν	logical shift left
	shr r ν ν	logical shift right

I ::=	instruction sequences:
ι ; … ; ι	instructions

Figure 1: Instructions and operands

Its evaluation rules are given in Figure 2.