

ネイティブスタック	Ruby スタック	ほしいスタック <要出典>				
Native: (no symbol information)	block (3 levels) in <main>	Native: (no symbol information)				
Native: vm_call_iseq_setup_normal_0start_3params	Kernel#foo	Native: vm_call_iseq_setup_normal_0start_3params_3locals				
Native: vm_exec_core	block (2 levels) in <main>	Native: vm_exec_core				
Native: rb_vm_exec	Kernel#foo	Ruby: block (3 levels) in <main>	← rb_yield() の2つ上に入れて見るだけ			
Native: rb_yield	block in <main>	Native: rb_vm_exec				
Native: rb_foo	Integer#times	Native: rb_yield				
Native: vm_call_cfunc_with_frame	<main>	Ruby: Kernel#foo	← rb_foo() が foo() であることを知らないと推定できない			
Native: vm_sendish.constprop.0		Native: rb_foo	↑			
Native: vm_exec_core		Native: vm_call_cfunc_with_frame	↑ vm_call_cfunc_with_frame() の2つ上に入れる的なヒューリスティックにはできる			
Native: rb_vm_exec		Native: vm_sendish.constprop.0				
Native: rb_yield		Native: vm_exec_core				
		Ruby: block (2 levels) in <main>	← rb_yield() の2つ上に入れて見るだけ			
Native: rb_foo		Native: rb_vm_exec				
Native: vm_call_cfunc_with_frame		Native: rb_yield				
Native: vm_sendish.constprop.0		Ruby: Kernel#foo				
Native: vm_exec_core		Native: rb_foo				
Native: rb_vm_exec		Native: vm_call_cfunc_with_frame				
Native: rb_ec_exec_node		Native: vm_sendish.constprop.0				
Native: ruby_run_node		Native: vm_exec_core				
Native: main		Ruby: block in <main>				
Native: (no symbol information)		Ruby: Integer#times	← 本当にここ？			
Native: __libc_start_main		Ruby: main				
Native: _start		Native: rb_vm_exec	← vm_exec_core のほうが良い			
Native: (no symbol information)		Native: rb_ec_exec_node				
		Native: ruby_run_node				
		Native: main				
ISEqを積み命令は4つだけ		Native: (no symbol information)				
send		Native: __libc_start_main				
opt_send_without_block		Native: _start				
invokesuper		Native: (no symbol information)				
invokeblock						
Code	foo()					
	VALUE					
	rb_foo(int argc, VALUE argv[], VALUE self)					
	{					
	rb_yield(self);					
	return Qnil;					
	}					
	RUBY_FUNC_EXPORTED void					
	Init_myecxt(void)					
	{					
	rb_define_method(rb_mKernel, "foo", rb_foo, -1);					
	}					
1000.times do						
some_cfunc do						
foo do						
do_something						
end						
end						
end						

Native: (no symbol information)							
Native: vm_call_iseq_setup_normal_0start_3params_3locals							
Native: vm_exec_core							
Ruby: block (3 levels) in <main>							
Native: rb_vm_exec							
Native: rb_yield							
Ruby: Kernel#foo							
Native: rb_foo							
Native: vm_call_cfunc_with_frame							
Native: vm_sendish.constprop.0							
Native: vm_exec_core							
Ruby: block (2 levels) in <main>							
Native: rb_vm_exec							
Native: rb_yield							
Ruby: Kernel#foo							
Native: rb_foo							
Native: vm_call_cfunc_with_frame							
Native: vm_sendish.constprop.0							
rb_hash_aref()	← rb_hash_aref() はあるけど、Ruby の方には Hash#[] がないケースが存在しうる (最適化)						
Native: vm_exec_core	opt_hash_aref 命令に対応していることは (VMのCスタックを見るだけでは) 分からないかな						
Ruby: block in <main>							
Ruby: Integer#times							
Ruby: main							
Native: rb_vm_exec							
Native: rb_ec_exec_node							
Native: ruby_run_node							
Native: main							
Native: (no symbol information)							
Native: __libc_start_main							
Native: _start							
Native: (no symbol information)							