The Definition of λ_Q

April 9, 2021

Contents

1 Syntax

t	$::=$		**terms**:
		x	variable
		$unit$	constant unit
		$true$	constant true
		$false$	constant false
		$\lambda\ x : T.t$	function abstraction
		$t\ t$	function application
		(t, t)	pair
		$t.1$	first projection
		$t.2$	second projection
		$if\ t\ then\ t\ else\ t$	conditional
		$run\ C$	static lifting
		$\zeta\ p : W.C$	circuit abstraction
v	$::=$		**values**:
		$\lambda\ x : T.t$	abstraction value
		(v, v)	pair value
		$unit$	unit value
		$true$	true value
		$false$	false value
		$\zeta\ p : W.C$	circuit value
T	$::=$		**types**:
		$Unit$	unit type

		$Bool$	boolean type
		$T \times T$	product type
		$T \rightarrow T$	function type
		$T \rightsquigarrow T$	circuit type
Γ	$::=$		**contexts**:
		$\varnothing$	empty context
		$\Gamma, x : T$	term variable binding
W	$::=$		**wire types**:
		One	wire unit type
		Bit	bit type
		$Qubit$	qubit type
		$W \otimes W$	wire product type
p	$::=$		**wire patterns**:
		$()$	empty
		w	wire variable
		(p, p)	wire pair
C	$::=$		**circuits**:
		$output\ p$	output a pattern
		$p_2 \leftarrow gate\ g\ p_1; C$	gate application
		$p \leftarrow C; C$	circuit composition
		$x \Leftarrow lift\ p; C$	dynamic lifting
		$unbox\ t\ p$	circuit application
Ω	$::=$		**wire contexts**:
		$\varnothing$	empty context
		$\Omega, w : W$	wire variable binding

2 Type Checking Rules

First, we need to define what is a well-formed wire context.

Definition 1 (Well-formed Wire Contexts)**.** *A wire context Ω is well-formed, if there are no duplicate wire variables in it. For simplicity, we always assume the wire contexts are well-formed in the following contexts. And when we write Ω_1, Ω_2, we require Ω_1 and Ω_2 to be disjoint to preserve the well-formedness.*

Since there are three different kinds of terms: (λ-)terms, wire patterns and circuits, we have three different relations for each of them to check the correctness.

- $\Omega \Rightarrow p : W$ is used to check the well-formedness of patterns.
- $\Gamma; \Omega \vdash C : W$ is the typing relation for circuits;
- $\Gamma \vdash t : T$ is the typing relation for (λ-)terms;

Well-formed patterns: $\boxed{\Omega \Rightarrow p : W}$

$$\frac{}{\varnothing \Rightarrow () : \mathit{One}}$$

$$\frac{}{w : W \Rightarrow w : W z}$$

$$\frac{\Omega_1 \Rightarrow p_1 : W_1 \quad \Omega_2 \Rightarrow p_2 : W_2}{\Omega_1, \Omega_2 \Rightarrow (p_1, p_2) : W_1 \otimes W_2}$$

Typing rules for circuits: $\boxed{\Gamma; \Omega \vdash C : W}$

TODO: add typing rules for circuits.

Typing rules for (λ-)terms: $\boxed{\Gamma \vdash t : T}$

TODO: add typing rules for terms.

3 Operational Semantics

TODO: add semantics.