Dart Kernel Semantics
(draft)

June 8, 2017

The small-step operational semantics of Dart Kernel is given by an abstract machine in the style of the CESK machine. The machine is defined by a single step transition function where each step of the machine starts in a configuration and deterministically gives a next configuration.

1 Definitions

1.1 Conventions

1.2 Domains

1.3 Meta-functions

1.4 Notations

1.5 Configurations for the CESK machine

The state space of the CESK machine contains various kinds of configurations, each containing components for applying the appropriate continuation in order to transition to the next configuration.

$\langle E,\, \rho,\, st, H,\ cex,\, cst,\, \kappa_E \rangle_{\text{eval}}$	:	EvalConfiguration
$\langle Es,\, \rho,\, st, H,\ cex,\, cst,\, \kappa_E \rangle_{\text{evalList}}$	:	EvalListConfiguration
$\langle S,\, \rho,\, lbls,\, clbls,\, st,\, H,\, cex,\, cst,\, \kappa_E,\, \kappa_S \rangle_{\text{exec}}$	:	ExecConfiguration
$\langle \kappa_E,\, v \rangle_{\text{cont}}$	:	ValuePassingConfiguration
$\langle \kappa_A,\, vs \rangle_{\text{acont}}$	:	ApplicationConfiguration
$\langle \kappa_S,\, \rho \rangle_{\text{acont}}$	:	ForwardConfiguration
$\langle H,\, v,\, st \rangle_{\text{throw}}$	:	ThrowConfiguration
$\langle \kappa_B \rangle_{\text{breakCont}}$	:	BreakConfiguration
$\langle \kappa_{switch} \rangle_{\text{switchCont}}$	:	SwitchConfiguration