Syntax and Semantics of D

Roger Burtonpatel

November 24, 2023

1 Syntax

We present a grammar of D, the language of decision trees:

Programs	P	::=	$\{d\}$	definition
Definitions	d	::=	`val` x`=` e	bind name to expression
Expressions	e	::=	x	name
		|	$\mathcal{D}_\alpha$	decision trees
		|	$K\{e\}$	value constructor application
		|	e_1 e_2	function application
Decision Tree	$\mathcal{D}_\alpha$	::=	`case` x `of` $\{\|$ $K\{x\}$ `=>` $\mathcal{D}_\alpha\}[\|$ x `=>`$\mathcal{D}_\alpha]$	test node
		|	α	match node
		|	`if` x `then` $\mathcal{D}_\alpha$ `else` $\mathcal{D}_\alpha$	condition with two children
		|	`let` x `=` knf `in` $\mathcal{D}_\alpha$	let-bind a name
Value Constructors	K	::=	`::`	cons
		|	`[]`	empty list
		|	`#`x	name beginning with `#`
		|	`A-Z`x	name beggining with capital letter
		|	[`-`|`+`](`0` − `9`)+	signed integer literal

2 What is a decision tree?

Scott, Ramsey 2000:
A decision tree is a pattern-matching automaton in which every state except the initial state has a unique predecessor.

More details will go here as needed.