A Formal Model and Interactive Visualization of miniKanren Search Semantics

Brysen Pfingsten Jason Hemann

Abstract

Mechanized executable semantics are a valuable tool for implementers and users alike to better understand their programs' behavior. The need is particulary acute in the context of novel logic programming languages with complex search strategies. Few such tools exist for logic-based programming languages however, and existing work only models languages' search behavior at a somewhat coarse-grained level. In this work, we present the first small-step semantics for a logic language that models interleaving search at the level of individual goal execution. The model is implemented in PLT Redex, a Racket-based semantics workbench, and allows users to step through the language's execution at a sub-interleave level.

We also present a visualization tool implemented through interactive JS in the browser with our miniKanren Redex model at its core. This visualizer allows users to input their own miniKanren programs as Racket source code; these programs are transpiled to the language of our model. At each step, users can easily see the evaluation context of their programs, trace goals both to and from the source code and the search tree visualization, and readily understand the history of the computation that led parts of the current state to arise. This not only enables better performance analysis and a unique, visual way of debugging but also allows novice miniKanren programmers to more readily understand the semantics and structure underlying the miniKanren search evolution.

A block containing a list

Nam vulputate nunc felis, non condimentum lacus porta ultrices. Nullam sed sagittis metus. Etiam consectetur gravida urna quis suscipit.

- **Mauris tempor** risus nulla, sed ornare
- **Libero tincidunt** a duis congue vitae
- **Dui ac pretium** morbi justo neque, ullamcorper

Eget augue porta, bibendum venenatis tortor.

A highlighted block

This block catches your eye, so **important stuff** should probably go here.

Curabitur eu libero vehicula, cursus est fringilla, luctus est. Morbi consectetur mauris quam, at finibus elit auctor ac. Aliquam erat volutpat. Aenean at nisl ut ex ullamcorper eleifend et eu augue. Aenean quis velit tristique odio convallis ultrices a ac odio.

- **Fusce dapibus tellus** vel tellus semper finibus. In consequat, nibh sed mattis luctus, augue diam fermentum lectus

Grammar

p	::=	`(prog` Γ e`)`	
Γ	::=	`((`$r_!$ d g`)` ...`)`	
d	::=	`(`$x_!$ ...`)`	
e	::=	`()`	Empty Tree
	|	`(`$\top$ σ`)`	Singleton Answer
	|	s	Search Tree
	|	`((`$\top$ σ`) +` e`)`	Answer Stream
s	::=	`()`	Empty Tree
	|	`(`g σ`)`	Goal with State
	|	`(`$s \rightarrow s$`)`	Right Tree Disjunction
	|	`(`$s \leftarrow s$`)`	Left Tree Disjunction
	|	`((`$\top$ σ`) +` s`)`	Answer Stream
	|	`(`s `×` g`)`	Tree Conjunction
	|	`(proceed ((`r t ...`)` σ`)`	Proceed
	|	`(delay` s`)`	Delay
g	::=	$\top$	Trivial Success
	|	`(`t `=?` t`)`	Unification
	|	`(`r t...`)`	Relation Call
	|	`(`$g \vee g$`)`	Goal Disjunction
	|	`(`$g \wedge g$`)`	Goal Conjunction
	|	`(`$\exists$ d g`)`	Fresh Variable Declaration
c	::=	`natural`	Logic Variables
o	::=	`boolean`	Booleans
	|	`string`	Strings
x	::=	(variable-prefix x)	Variables
r	::=	(variable-prefix r)	Relation Names
t	::=	c	
	|	o	
	|	x	
	|	`empty`	Empty List
	|	`(`t t`)`	Non-Empty List
sub	::=	`((`c t`)` ...`)`	Substitution
$maybe\text{-}sub$	::=	sub	
	|	$\#f$	
σ	::=	`(state` sub c`)`	State
$E\Gamma$	::=	`(prog` Γ `hole)`	Program
Ev	::=	`hole`	Answer Stream
	|	`((`$\top$ σ`) +` Ev`)`	
Es	::=	`hole`	Search Tree
	|	`(`$Es \leftarrow s$`)`	
	|	`(`$s \rightarrow Es$`)`	
	|	`(`$Es \times g$`)`	
Ex	::=	$E\Gamma[\![Ev[\![Es[\![$`hole`$]\!]]\!]]\!]$	Main

Fusce aliquam magna velit

Et rutrum ex euismod vel. Pellentesque ultricies, velit in fermentum vestibulum, lectus nisi pretium nibh, sit amet aliquam lectus augue vel velit. Suspendisse rhoncus massa porttitor augue feugiat molestie. Sed molestie

A highlighted block containing some math

A different kind of highlighted block.

$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}$$

Interdum et malesuada fames $\{1, 4, 9, \ldots\}$ ac ante ipsum primis in faucibus. Cras eleifend dolor eu nulla suscipit suscipit. Sed lobortis non felis id vulputate.

A heading inside a block

Praesent consectetur mi $x^2 + y^2$ metus, nec vestibulum justo viverra nec. Proin eget nulla pretium, egestas magna aliquam, mollis neque. Vivamus dictum $\mathbf{u}^\intercal\mathbf{v}$ sagittis odio, vel porta erat congue sed. Maecenas ut dolor quis arcu auctor porttitor.

Another heading inside a block

Sed augue erat, scelerisque a purus ultricies, placerat porttitor neque. Donec $P(y \mid x)$ fermentum consectetur $\nabla_x P(y \mid x)$ sapien sagittis egestas. Duis eget leo euismod nunc viverra imperdiet nec id justo.

Nullam vel erat at velit convallis laoreet

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Phasellus libero enim, gravida sed erat sit amet, scelerisque congue diam. Fusce dapibus dui ut augue pulvinar iaculis.

First column	Second column	Third column	Fourth
Foo	13.37	384,394	α
Bar	2.17	1,392	β
Baz	3.14	83,742	δ
Qux	7.59	974	γ

Table 1. A table caption.

Donec quis posuere ligula. Nunc feugiat elit a mi malesuada consequat. Sed imperdiet augue ac nibh aliquet tristique. Aenean eu tortor vulputate, eleifend lorem in, dictum urna. Proin auctor ante in augue tincidunt tempor. Proin pellentesque vulputate odio, ac gravida nulla posuere efficitur. Aenean at velit vel dolor blandit molestie. Mauris laoreet commodo quam, non luctus nibh ullamcorper in. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.