Langkit

source code analyzers for the masses

Pierre-Marie de Rodat Raphaël Amiard

Software Engineers at AdaCore



Langkit: A meta compiler

High level vision

A collection of DSLs to implement language parsing and analysis front-ends.

Front ends generated by Langkit could be the basis for:

[BULLET POINTS]



Original use case: Libadalang

[Description of Libadalang]



The DSL

DSL Episode 1: Lexing

DSL Episode 2: Tree

DSL Episode 3: Parsing

DSL Episode 4: Scoping

DSL Episode 5: Semantic analysis

AdaCore

10

Crafted for Incremental analysis

The generated libraries

Base library: Ada (W00T!)

Requirements for the target language:

- Fast
- Low level enough
- Memory management agnostic (no GC)
- Easy to bind to C and other languages

Candidates

■ C, C++, Ada, Rust, ...

Chosen one: Ada

Since the project is developed at AdaCore: no surprises:)



Bindings to other languages

Automatically generated C bindings

So that it is very easy to generate bindings to any languages the users wants.

First class citizen Python bindings

Python is the de-facto scripting language of the Langkit ecosystem.



Easy to generate bindings to new languages



Easy to interface

[explain ecore stuff]



Tree walking



Rewriting



Generic tools shipping with the libraries

Playground

AdaCore

20

Vim plug-ins



Language server protocol? (not done)

Demo!