ライブプログラミング環境は多言語化/多開発環境化の夢を見るか

高橋 修祐　伊澤 侑祐　増原 英彦（東京工業大学）

目標

多言語(・多開発環境)で高性能なライブプログラミング環境の構築

対応言語・開発環境の追加を用意に

背景
データ構造を可視化するLP環境

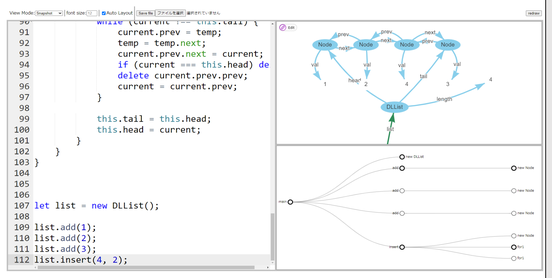
Program → Analyzer → Graph Generator → Graph

プログラムの各時点における
オブジェクトの状態を記録

- **Kanon**
  [Oka et al. '18]

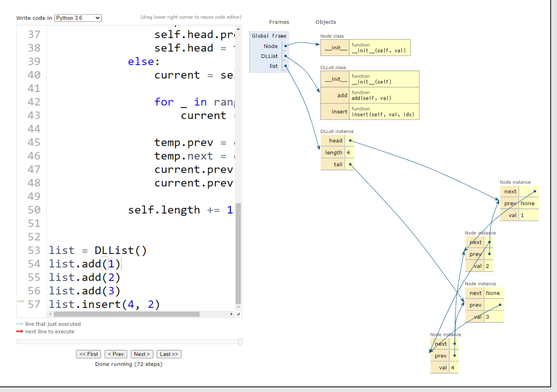
対象言語:

JS



- **Python Tutor**
  [Guo '13]

対象言語:



方針

Graal/Truffle[Würthinger et al. '12]を用いて実装

- オブジェクトの状態を記録するAPIを用意
- 言語ごとにインタプリタを準備し，適当な箇所で上記APIを呼び出す

JavaScript Program | Python Program | C/C++ Program | …

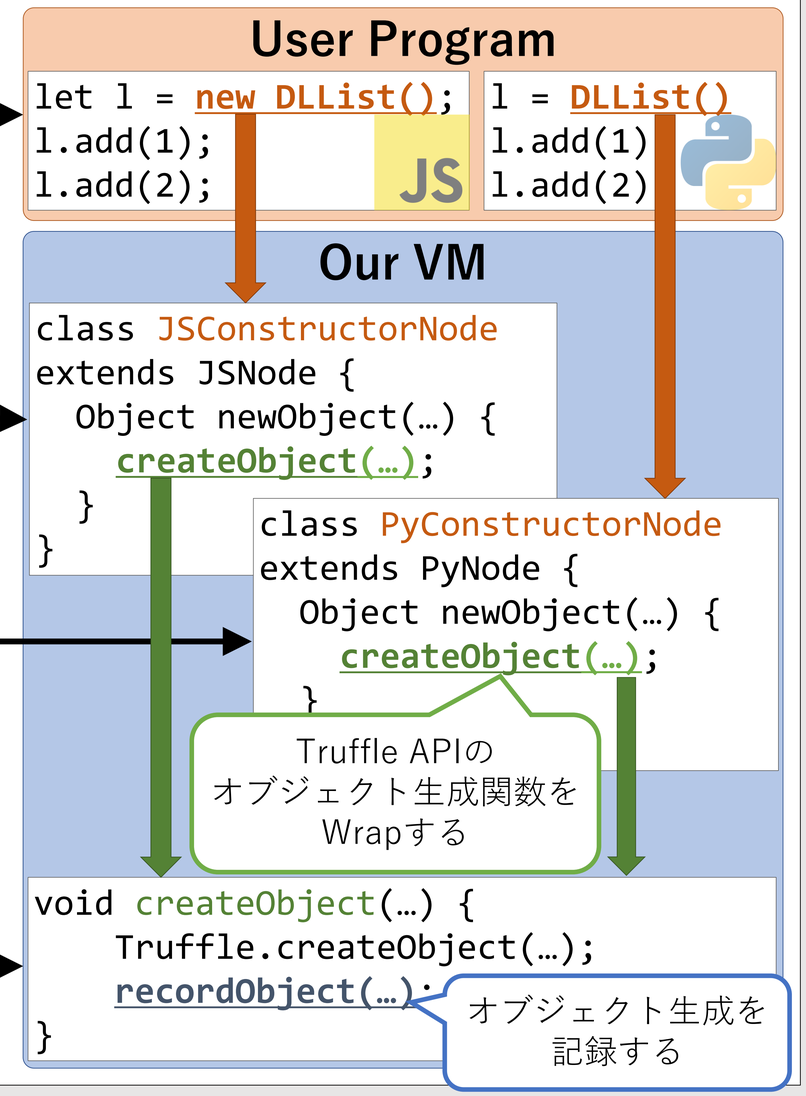
JavaScript Interpreter | Python Interpreter | C/C++ Interpreter | …

Our Framework

Graal/Truffle

Q. 新しく言語対応するには？
A. ASTインタプリタを実装するだけ

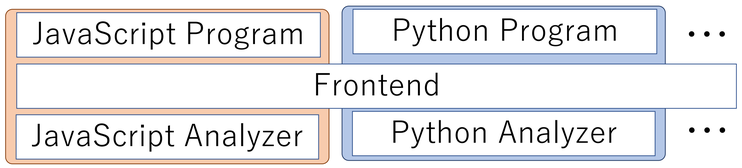
Q. パフォーマンスは？
A. Graal/TruffleがASTを元に最適化をし，かつJITコンパイルする為高速に実行される



既存実装の問題点

- パフォーマンス上の問題
  - プログラムが大きいときに低速
    - ➢ ライブ性の喪失

- 多言語化への障害
  - 解析部の実装が言語によってバラバラ



今後の課題

- 上記方針に基づいたフレームワークの設計，実装および性能の評価
- 既存のGraal/Truffleを用いたインタプリタの再利用の検討