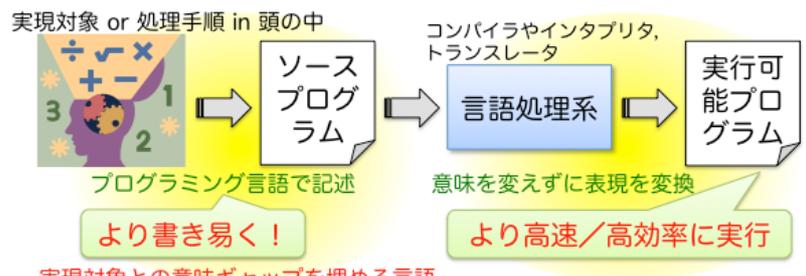


■研究分野：プログラミング言語・言語処理系

■なぜこの研究が必要か？

実行性能のよいプログラムを作るにはそのアルゴリズムが重要なのは言うまでもありません。しかし、①プログラミング言語処理系の性能もプログラムの実行性能に大きな影響を与えます。また、②高級な言語機能がもたらすプログラムの記述性・表現力は、開発効率などの点で「よいプログラム作り」に影響を及ぼします。ただし、③そのような機能の効率的な実装法は難しいのが一般的です。

そこで本研究室では、記述性・表現力に優れた言語機能及びその効率的な実装方法の研究を行っています。また、ごみ集め（最近の言語には必ず入っている自動メモリ管理機構）などのプログラミング言語に関連する技術の研究も行っています。



実現対象との意味ギャップを埋める言語機能により記述力・表現力向上

■この研究の楽しいところや身に付くこと

- 自分で考えた仕組みに基づく言語や言語機能を新しく設計し、その言語/言語機能によるプログラムが実際に動く高性能処理系を開発することでプログラム実行の一通りの理屈を理解できる
- コンピュータにおける「計算」を抽象的に捉えることでプログラム/言語の本質を知る（フレームワークや API の使い方を覚えるだけでは獲得できない見方）

■具体的に何をするのか？

コンパイラやインタプリタ、トランスレータ等の言語処理系やランタイム等のソフトウェアを作ります。既存の言語処理系を改良・拡張して高速化・高機能化することもあります。どんな言語・言語処理系等を作るか、それをどのような手法で実現するか等はその目的によって様々です。プログラムの解析やツールをテーマとすることもあります。

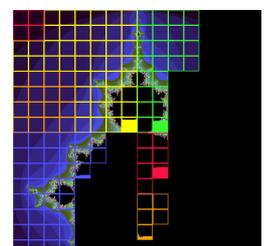
■研究テーマの一例（裏面に過去のテーマを掲載）：

- 動的言語機能の実行性能向上の研究
- 一級継続の仕様や実装方法、応用方法の研究 — Web アプリケーションへの応用、PC やモバイル端末等様々なコンピュータをシームレスに利用するためのプロセス移送の研究など
- ガーベジコレクション(GC)付き言語の実装を手助けするようなごみ集め方式の研究 — 実時間 GC や参照局所性を考慮した GC の研究、GC 付き言語の実装を手助けする GC など
- プログラミング言語の研究知見を活用したツールの開発 — ソースコード変換による動的情報流解析など
- Web アプリケーション向け言語/言語機能の研究
など

【参考】過去の修論／卒論テーマ（@前職のものを含む）

- メモリリーク原因となる参照経路の作成箇所を特定可能な GC アサーション
- 関数型言語の諸概念を取り入れたシェル
- SBFL とスライシングの併用による欠陥位置特定精度の向上
- 高階言語-to-C 処理系における真の末尾再帰実現の一手法
- 動的負荷分散の高級言語による記述と性能特性を考慮した実装
- インタプリタとコンパイラ生成コードの対比によるコンパイラ実行時環境理解支援ツール
- 末尾呼び出し最適化を考慮したバックトレース
- 数値計算ライブラリ IFN の数値表現の改良と合成演算の導入によるメモリ効率の改善(岩崎英哉先生の指導によるテーマ)
本卒論発表にて、 本学同窓会日黒会令和 4 年度日黒会賞受賞 および
第 25 回プログラミングおよびプログラミング言語ワークショップ (PPL 2023) 発表賞 (学生の部) 受賞
- アセンブリ言語プログラムのレジスタ値の定義元表示ツール
- クロージャ生成のメモ化による最適化
- 第一級関数に対応した表示的プログラムスライシング
- 値への追跡子付与を活用したデバッグ機構
本テーマの学会発表にて、 情処第 84 回全国大会学生奨励賞受賞
- VM 命令セットの違いによるスーパー命令の性能への影響調査
- Siek らによる C++ テンプレートのインスタンス化意味論の実装
- 前処理による静的解析ツール Solhint の解析性能向上
- プログラム記述の違いによるプログラミング言語の性能差評価
- 近年の CPU における cdr コーディングの評価
- 関数型リアクティブプログラミングのためのビジュアル開発環境の実装
本卒論発表にて、 本学同窓会日黒会令和 2 年度日黒会賞受賞
- アサーションの情報を取り入れた静的解析に基づくデバッグ支援ツールの開発
- ごみ集めにおける自動ルート保護と手動ルート保護の併用手法
- TypeScript における型定義を切り替え可能な型検査補助ツールの実装
- FFI をもつ Scheme インタプリタのための部分コンパイル方式コンパイラの構成法
本テーマの学会発表にて、 情処第 81 回全国大会学生奨励賞受賞
- Web ブラウザにおける Scheme 仮想機械の実装
- 様々な実装方式の仮想機械の性能比較
- タグ付きポインタを考慮した動的メモリ解析手法
- アロケーションサイトに応じて GC アルゴリズムを選択可能なハイブリッド GC
- LLDB を用いたソース-to-C 型トランスレータ用ソースレベルデバグの実装手法
- オフライン Web アプリケーションにおける事前データ取得の半自動化
- 対話型環境を持つ分散型スクリプト言語の設計と実装
本テーマの学会発表にて、 情処第 77 回全国大会学生奨励賞受賞
- 仮想メソッド呼出しを用いることによるプログラム難読化のオーバーヘッド削減
本テーマの学会発表にて、 情処第 76 回全国大会学生奨励賞受賞
- 実行時情報を活用した第一級継続のオーバーヘッド削減手法
- 遅延スイープ法におけるメモリアクセスの局所性改善
- JavaScript における動的情報流解析のソースコード変換による実装
- メモ化の適用範囲を拡張するプログラム変換器の設計と実装
- Scheme における動的ソフトウェア更新機構の設計と実装
- ページ遷移を考慮した Web アプリケーション記述言語の設計と実装
- 継続の共有化による継続ベース Web サーバのメモリ使用量削減
- Web アプリケーションのための動的適応可能な処理分担機構の設計と実装
- Web アプリケーションのための計算移送機構
本テーマの学会発表にて、 情処第 70 回全国大会学生奨励賞 及び SPA2007 優秀ポスター発表賞受賞
- CPS¹ 実行方式におけるスタックのごみ集めの実時間化手法
- CPS 実行方式に基づく Scheme-to-PHP コンパイラの効率の良い実装
- Scheme による XML 操作機能の設計と実装
- マルチプラットフォーム分散処理言語機能に関する研究
- Web ブラウザを用いた TUTScheme の対話型フロントエンドの実装
- Scheme-to-C コンパイラにおけるスタックサイズの自動調整
- CPS 実行方式における部分継続の実装
- マルチページ CGI 言語の設計と実装
- 移植性のよいデバグの実装のためのプログラム変換手法
- CPS 実行方式におけるモバイル継続の実装
- CPS 実行方式におけるリターンバリアの実装
- CPS 実行方式における LTC²の実装

```
void fib(Env *cc_e,
{
  Cont k_k28 = (Cont
int cps2c sp;
CHK STKOV2(fib, cc
if (BOOL((int)n <
  k_k28(cc_e, n);
} else {
  Env cc_new[] = {
  fib(cc_new, fib
}
}
```



¹ Continuation Passing Style

² Lazy Task Creation