

可逆グラフアルゴリズム

2016SE085 鳥居大樹

2016SE098 吉田翔亮

指導教員 横山哲郎

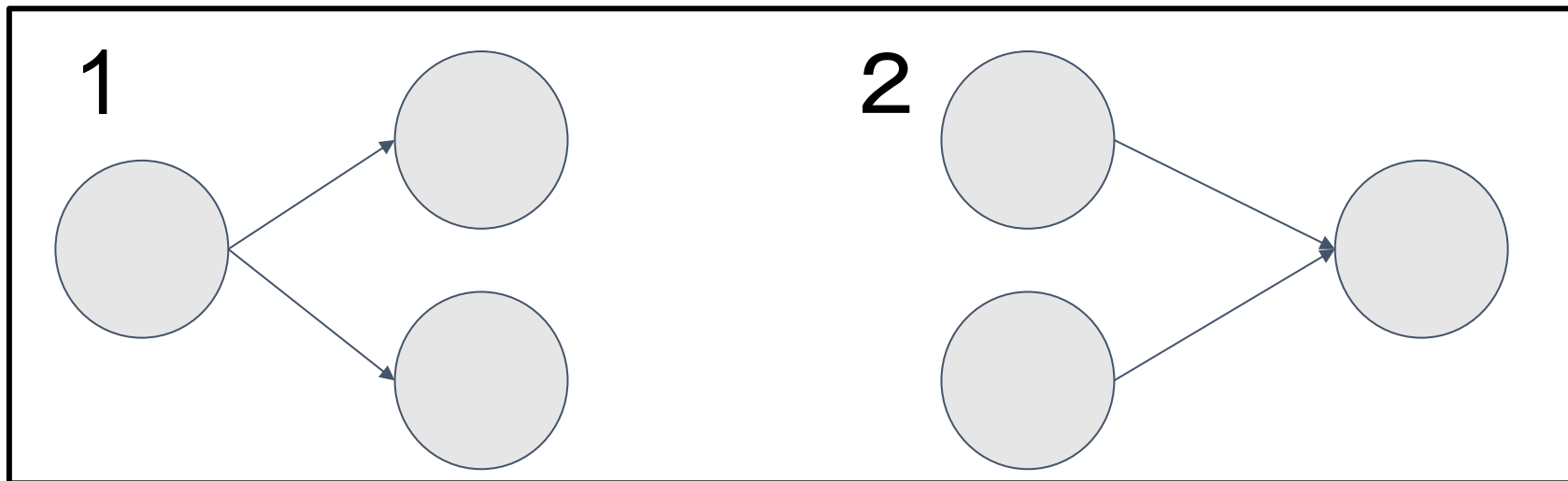
目次

1. はじめに
 - a. 背景
 - b. 目的
2. 関連研究
3. アプローチ
4. 提案手法
5. 計算量の解析
6. まとめ
7. 今後の課題

可逆とは

状態遷移図において

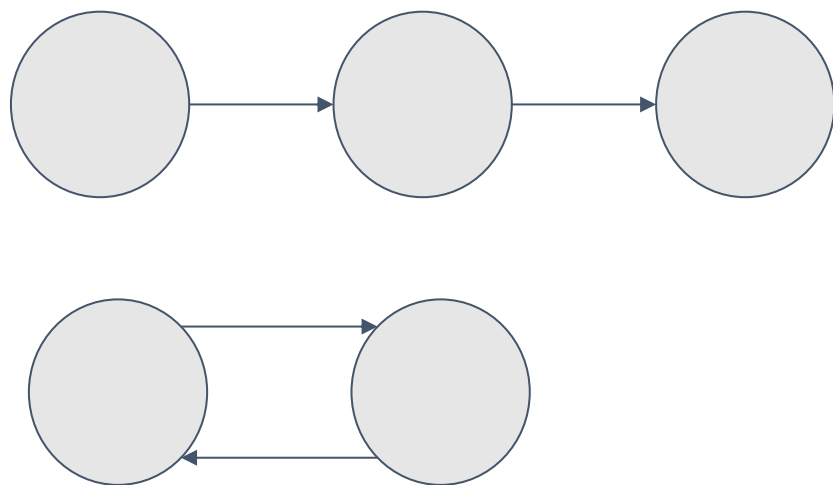
1. 一つの状態から複数の状態に遷移しないかつ
2. 複数の状態から一つの状態に遷移しない



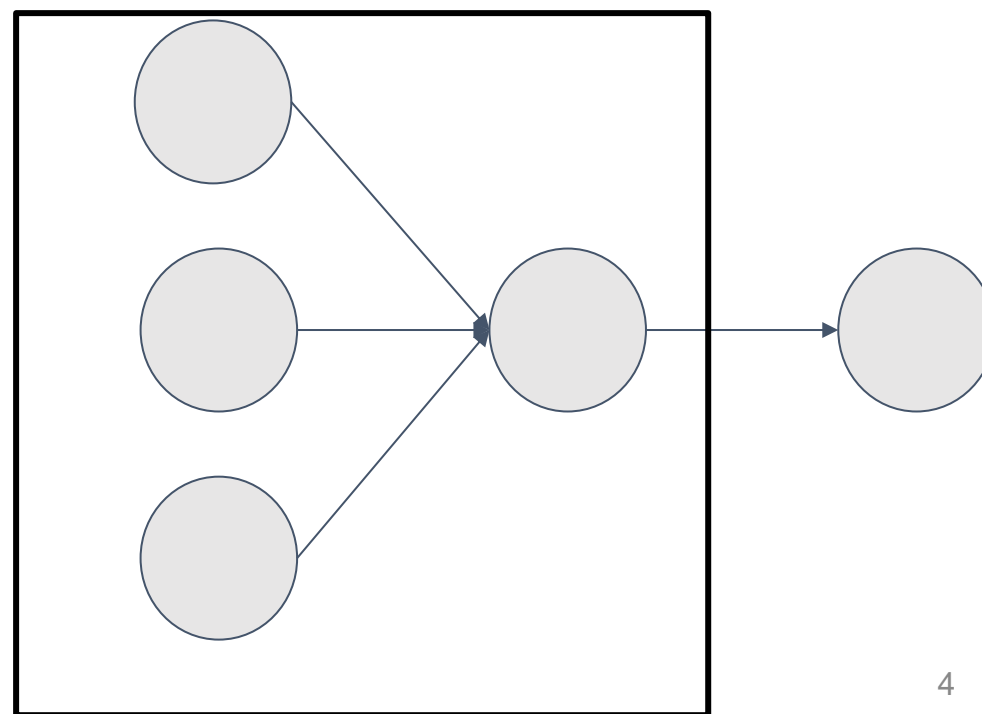
1.2が現れなければ可逆

可逆とは

可逆



非可逆



背景：可逆アルゴリズム研究

- 既存の可逆アルゴリズム
 - 可逆ソート、可逆線形探索
- 可逆アルゴリズムの応用
 - COMS回路による低消費エネルギー化
 - 可逆論理ゲート
 - 情報消失のない非破壊的計算
 - PDESの効率化
 - 量子アルゴリズム・量子回路 ※観測を除くと可逆

目的：可逆アルゴリズムの効率化

- 深さ優先探索アルゴリズムの可逆化
 - 一般解法による可逆化
 - 適切なデータ構造を明確化
- 可逆深さ優先探索アルゴリズムの効率化
 - 効率の指標の定式化
 - トレードオフの解析
- 可逆深さ優先探索プログラムの実装
 - 可逆プログラミング方法論の応用
 - 実装上の課題解決

関連研究

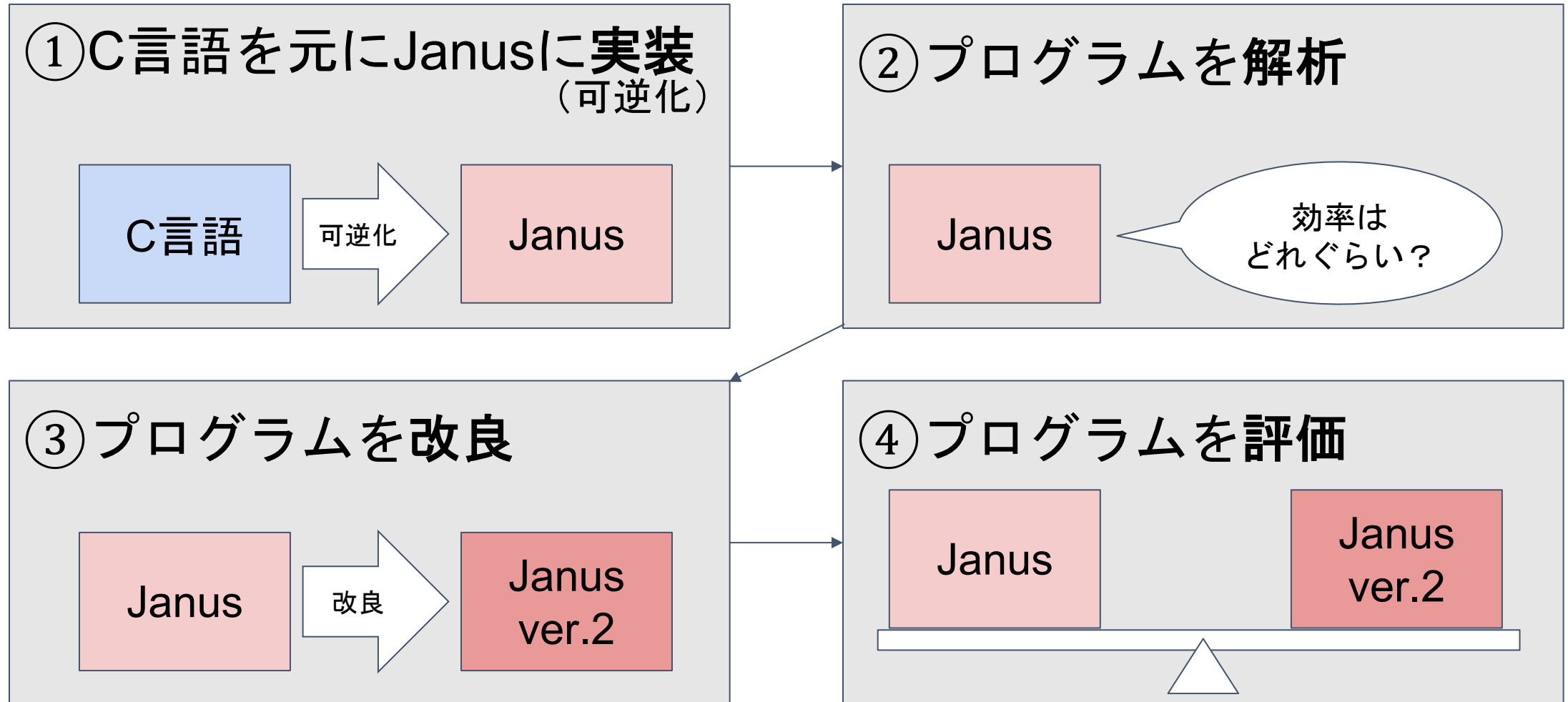
- ・ 可逆線形探索の効率化

一般解法で可逆化されたプログラムよりも効率の良いものを提案

- ・ ベネットの小石アルゴリズム

メモリ使用量の効率が良い可逆シミュレーション

アプローチ



グラフアルゴリズム

グラフアルゴリズム

深さ優先探索

深さ制限探索

反復深化
深さ優先探索

幅優先探索

最良優先探索

均一コスト探索

A^*

双方向探索

分枝限定法

ゴミ情報

- 可逆計算特有の概念
- 可逆では求めたい出力に加えて幾つかのデータが出力

例) $1+3 = 4 (+3)$

- 求めていた出力以外のデータはゴミ情報
 - 実行終了時のゴミ → 最終ゴミ情報
 - 実行中のゴミ → 中間ゴミ情報

Janus

Janus	C言語
命令型可逆プログラミング言語	命令型プログラミング言語
条件分岐や繰り返しなども可逆	条件分岐や繰り返しは非可逆
可逆性が保証されている	可逆性が保証されていない

一般解法

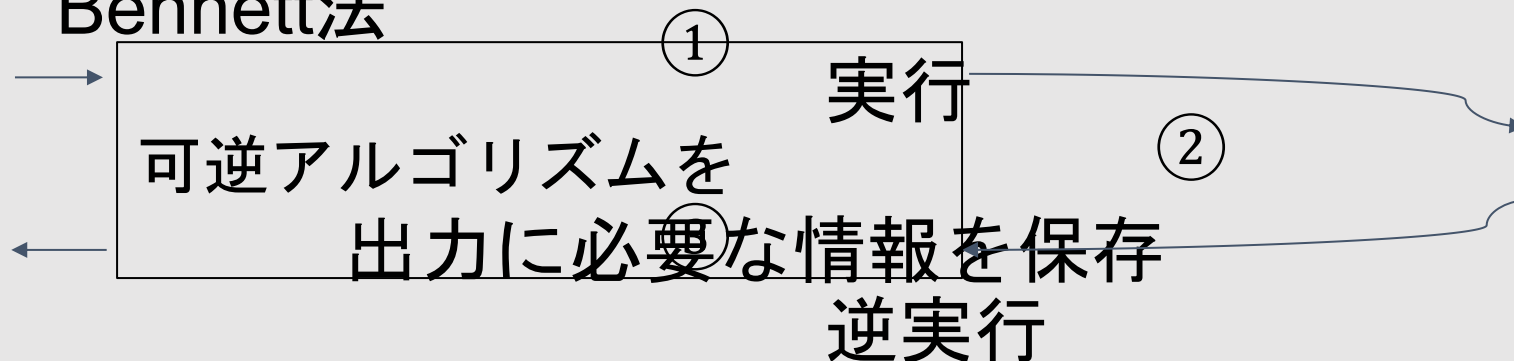
- Landauer法（埋め込み法）

非可逆計算で失われる情報をゴミ情報として保存

利点：実行時間が元の計算から大きく増加しづらい

欠点：メモリ使用量が多い

- Bennett法



利点：最終ゴミが入力のみになる

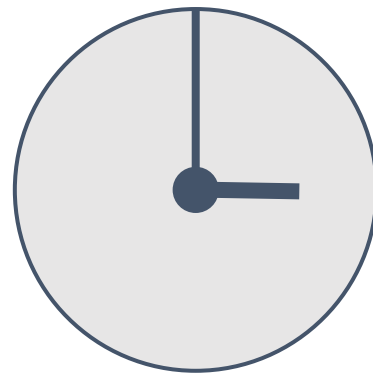
欠点：実行時間がほぼ2倍，中間のメモリ使用量が多い

計算量の解析の準備

- 評価基準
 - a. 時間計算量
 - b. 空間計算量
 - c. 完全性
 - d. 最適性
 - e. ゴミ情報
- 使用する計算モデルの決定
- 入力/出力の整理
- 基本演算の決定
 - a. 解析を行うアルゴリズムの特徴的な部分の発見
 - b. 基本演算を変えた場合の影響の模索

計算量の解析

- 時間計算量
 - 可逆探索アルゴリズムを用意
 - 基本演算の実行回数の計算
 - アルゴリズム毎の値の比較
- 空間計算量
 - 使用するメモリの量の計算
 - ↳ 配列や変数



まとめ

- 深さ優先探索アルゴリズムを可逆化
 - Bennett法とLandauer法による可逆化
- 可逆深さ優先探索アルゴリズムの効率化
 - 主に時間計算量に着目
 - 【一般解法より効率的/非効率的な点. トレードオフの解析】
- 可逆深さ優先探索プログラムの実装
 - Janusに実装
 - 【使用したプログラミング方法論】

今後の課題

n:頂点の数

L:探索する頂点が何番目にあるか

S:探索成功なら1失敗なら0

比較項目	提案解法a	提案解法b
メモリ使用量	$3L-2S+5$	1
ステップ数	$3L-4S+8$	$6L-8S+16$
ゴミ出力	$3L-2S+5$	0

a,bともにnに着目すると効率化は行えていない

参考文献

- [1]浅野早紀,山口春樹:可逆な深さ優先探索, 南山大学2018年度卒業論文(2019)
- [2]家崎雄太,水野竣太郎:可逆線形探索, 南山大学2017年度卒業論文(2018)
- [3]Axelsen, H. B. and Yokoyama, T.:Programming Techniques for Reversible Comparison Sorts(2015)
- [4]Michael P. Frank :Reversibility for Efficient Computing,pp.64-70(1999)

担当

鳥居担当分：1,2,3,4,7,8,11,12,17(ページ)

吉田担当分：4,8,9,10,12,13,14,17(ページ)