修論中間発表に向けて
(2006/7/26)

M2 佐藤秀明

概要

- コードクローンの解消
  - 高速
  - 使いやすい

コードクローンとは

- 機能または文面が似ているコード
  - ソースをコピー & ペースト
  - 設計の洗練不足
- 大規模なシステム開発において有害
  - メンテナンス性の低下
- ソースコードの類似性を解析して発見
  - どうやって発見するか？

CCFinder[Kamiya et al.]

- トークン列をマッチング判定
- Parameterized suffix tree[Baker] を採用
  - すべての接尾語を登録した trie
  - 同じ変数の出現を直前の出現位置からの距離に置換
    - 例 :xbyyxbx ⇒ 0b014b2
  - 変数名の違いを吸収したマッチング

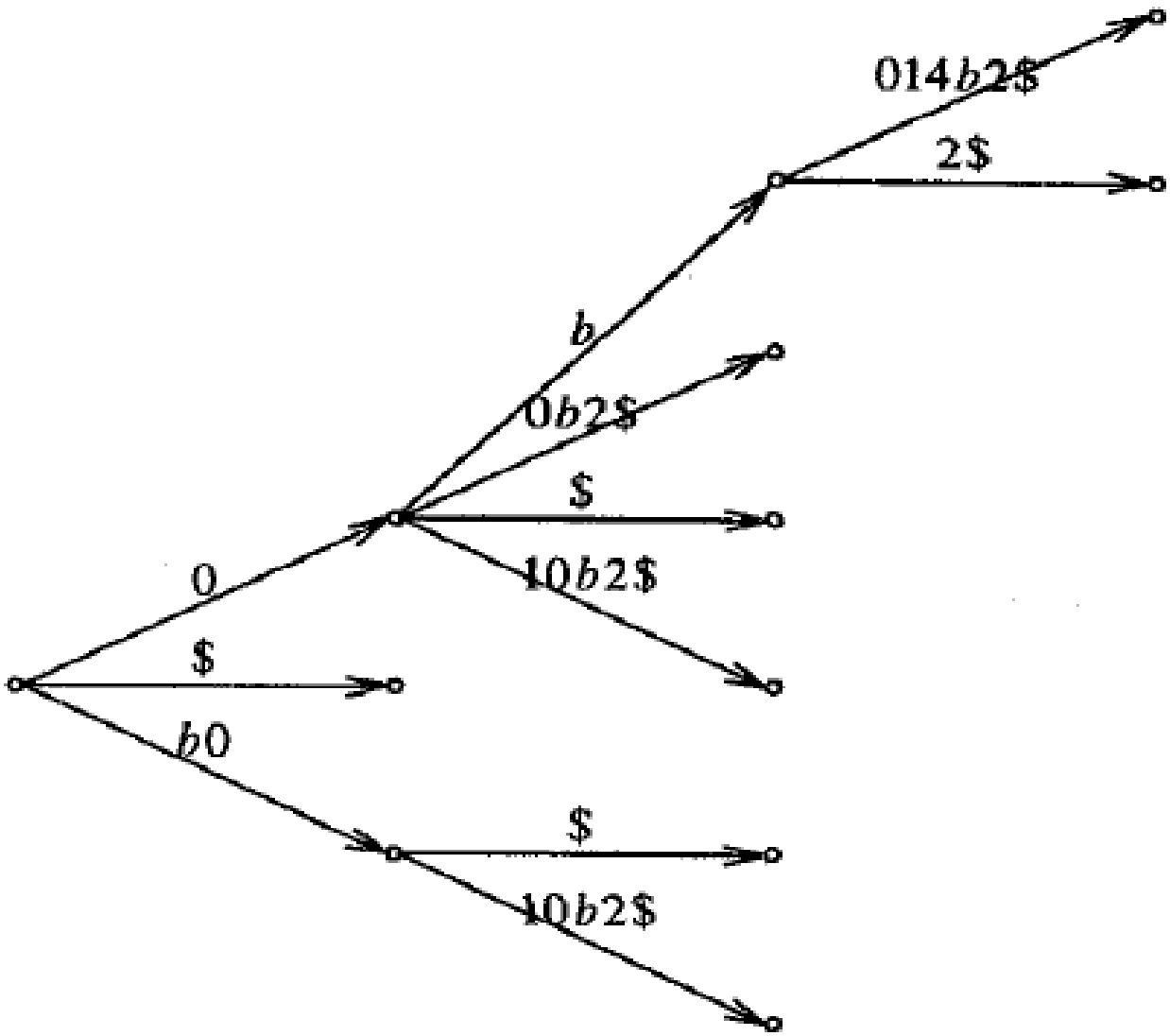


Figure 5: A p-suffix tree for the p-string $S = xbyyxbx\$$.

問題

- 解析作業は重い
  - tree のメモリ消費量が甚大
  - 時間もかかる
- 一度に提示するクローンの数が多すぎる
  - 膨大な情報に溺れる
- クローンをどう除去するかについてはノータッチ
  - 実際の除去方法はユーザに任せきり

基本的なアイデア

- 毎日少しずつリファクタリング
  - 昨日からの差分のみを解析対象にする
  - コードを常にきれいに保つ
- 多段階のステップでクローン候補を選抜
  - 重い解析を行う対象をできるだけ減らす
- クローン除去を視覚的に支援
  - 瑣末な書き換えミスを防止
  - 「面倒」というイメージを改善

バージョン管理システムとの連携

- リビジョンを上げるときにクローン解析を行う
  - プログラマにとって妥当なタイミングであろう
- バージョン管理システムから必要な情報を取得
  - 更新されたファイルの一覧

Suffix Array[Manber et al.] の導入

- 各 suffix をソートしたもの
- tree より扱いやすい
  - メモリの節約
  - 単純な構造
  - キャッシュミスの軽減

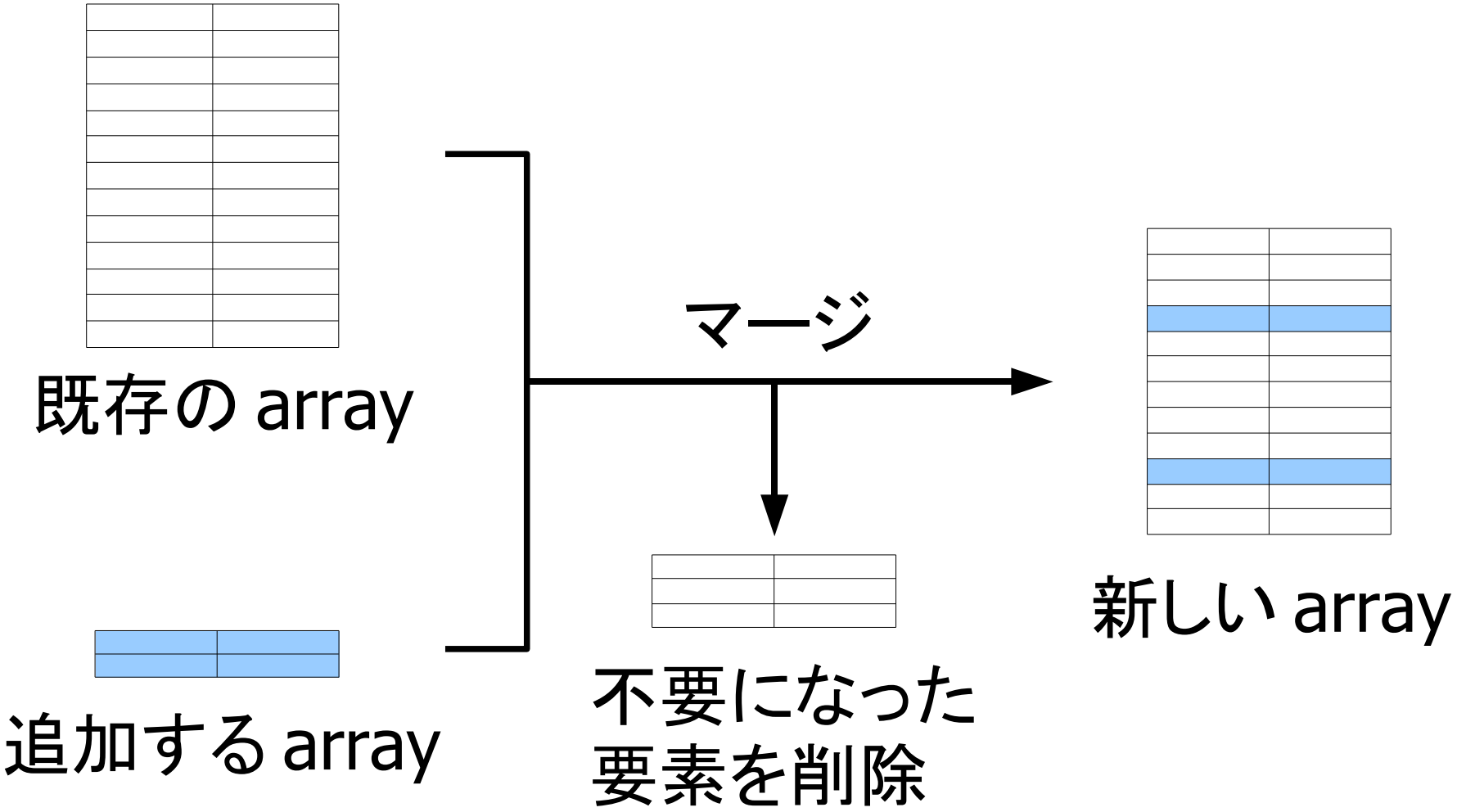
acaaacatat$ 

index	pos	実体
0	2	aaacatat$
1	3	aacatat$
2	0	acaaacatat$
3	4	acatat$
4	6	atat$
5	8	at$
6	1	caaacatat$
7	5	catat$
8	7	tat$
9	9	t$
10	10	$

Suffix Array の逐次更新

- 毎回 array を一から作り直すのは無駄
  - 昨日からの差分のみを反映させれば済むはず
- array の内容を後から改変できるしくみを導入
  - 更新範囲は最小限に抑える
- array 構造は普段はファイルに書き出しておく
  - 解析時に読み込み

Suffix Array の更新手順



構文 / 意味解析

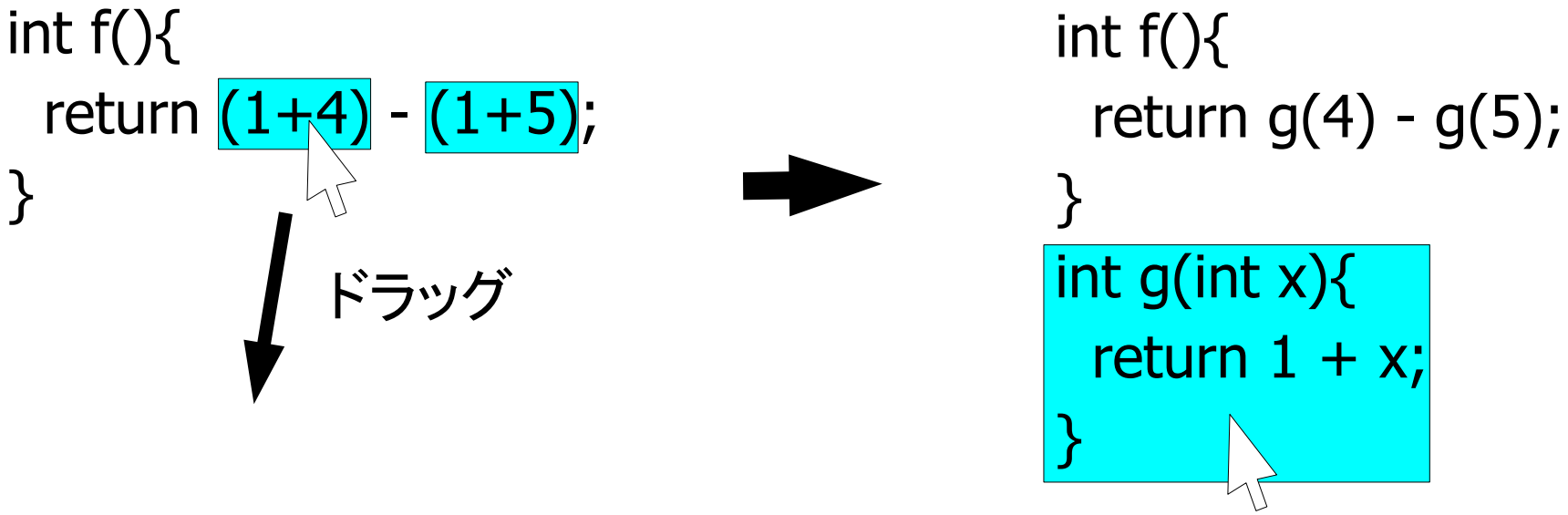
- suffix array で絞られたクローン候補に適用
  - 全体の計算時間は小さいはず
- プログラムの変換に必要な情報を取得
  - パーズ
  - 型情報の取得
  - 生存期間解析
  - 自由変数解析
  - ライブラリ / 変数への依存関係
- 変換方法の候補を列挙
  - 常に 1 通りとは限らない

コード書き換え

- 実際に書き換えるか否かの判断は人間に任せる
  - プログラムの構造は設計者のポリシーに依存
- 全自動ではなく人間自身に書き換えさせるべき
  - 人間の意図をコードへ正しく反映させるため
- 書き換えが簡単かつ面白い作業ならなおよい
  - リファクタリングしたくない人を説得

視覚的なコード書き換え

- クローンの抽出先をドラッグ＆ドロップで指示
  - どこへ抽出するかによって抽出手順を柔軟に調整
  - プログラムの意味を変えずに正しく変換



各プログラマ間の連携

- 複数プログラマの担当範囲を跨るコード書き換え
  - 他人のコードを勝手に書き換えていいのか
    - 自分のコードを拝借された場合にのみ書き換えを許す？
    - 書き換えたコードの担当者に連絡する？
- 考え中です

これからの予定

- Suffix Array によるクローン候補解析 (8 月)
- プログラムの構文解析 / 意味解析 (9 月)
- 視覚的コード書き換え (10 月)
- 予備月 (11 月)
- 評価 (12 月)
- 論文執筆 (1 月)

まとめ

- コードクローンの高速 / 簡単な除去
  - 更新可能な suffix array の導入
  - クローン候補の段階的選抜
  - ドラッグ＆ドロップによる抽出先の指定