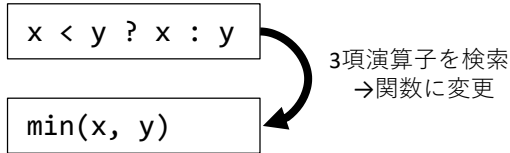


研究背景：コード片検索

- コード片検索：ソフトウェアの管理に役立つ
 - ✓ 既知のバグコード片から類似コード片を探す
 - ✓ コーディングパターンを探してリファクタリング



- コード片から**コードクローンを検索**するツールは少ない
 - grepは広く使用されるが複雑なコード検索には不向き

ccgrep：検索仕様

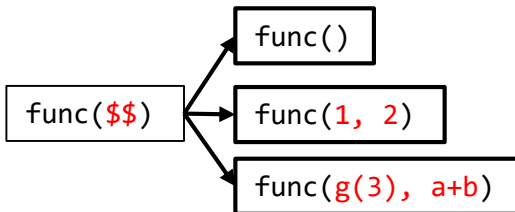
- クエリに対し、パラメータ化されたクローンにマッチ
 - ✓ \$を付けると完全一致

クエリ例

```
T $addValue(T v) {
    v = v + 1;
}
```

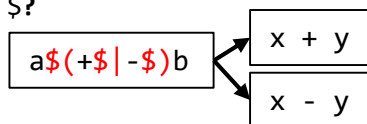
```
int addValue(int value) {
    value = value + 1;
}
```

- \$\$ X:Xで終わる最短トークン列
 - 括弧の釣合いを考慮してマッチ



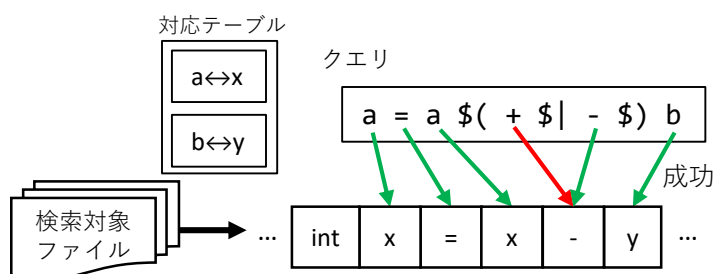
正規表現

- 選択 \$|
- 繰返し \$*, \$+, \$?
- 任意の1トークン \$.
- グループ핑 \$ (, \$)



ccgrep：アルゴリズム

- 対象トークン列の先頭から順にマッチするか調べる
- 対応テーブルにより識別子をパラメータ化



ccgrep：検索例

```
クエリ cccgrep 'a<b?a:b' -r src -l c -p nf
```

```
src/arch/cris/arch-v32/drivers/cryptocop.c
2782:cop->tfrm_op.indata[i].iov_len = tmp1en < PAGE_SIZE ? tmp1en : PAGE_SIZE;
src/arch/cris/arch-v32/kernel/fasttimer.c
701:num_to_show = (fast_timers_started < NUM_TIMER_STATS ? fast_timers_started:
702: NUM_TIMER_STATS);
```

既存のコードクローン検索ツール

- NCDSearch[1] (正規圧縮距離を使用)
- CBCD[2] (プログラム依存グラフを使用)
 - 検索時間が長い
 - 検索を詳細に制御できない

提案手法：ccgrep

- 特定クローン検索ツール**ccgrep**を開発
- ✓ クエリに**マッチ**するコード片を検索
- ✓ **grep**に似たUIで手軽に使用可能
- ✓ **トークン単位**での複雑な検索に対応
 - ✓ 空白・コメントを無視できる
 - ✓ 正規表現などを使用し、**検索を制御可能**
- ✓ **検索時間が短く**対話的に使用可能
- ✓ 対応言語：C/C++, Java, Python

実験1：クエリの作りやすさ

- 空白やコメントの処理

```
grep '\s*T\s+a\s*=\s*b\s*;' -r src/
```

(空白のみ対応)

```
ccgrep '$T $a = $b;' -r src/
```

- 任意の識別子

```
grep '[a-zA-Z][a-zA-Z_0-9]* = [a-zA-Z][a-zA-Z_0-9]*;' -r src/
```

```
ccgrep 'T = a;' -r src/
```

- 括弧の釣合い

```
grepでは書けない
```

```
ccgrep 'if(a == b) { $$ }' -r src/
```

grepより書きやすい



実験2：検索時間

- cccgrep, grep, NCDSearchの検索時間を比較

クエリA (ccgrep, NCDSearch)

```
a < b ? a : b
```

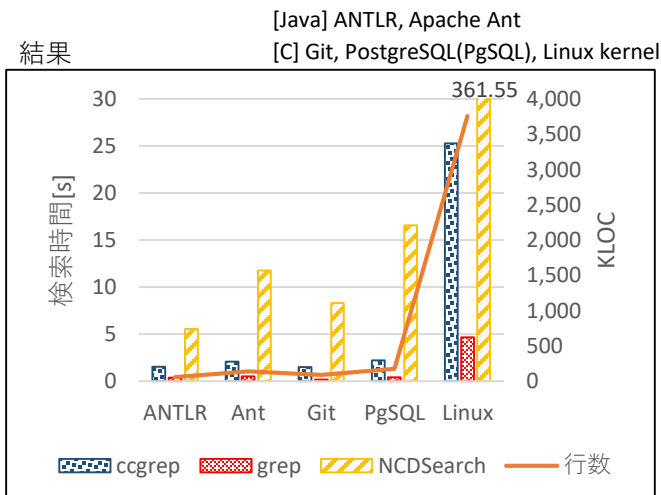
クエリA' (grep)

```
([a-zA-Z][a-zA-Z_0-9]*)
\s*<([a-zA-Z][a-zA-Z_0-9]*)\s*\?
\s*\1\s*:\s*\2
```

素早く検索可能



OS	Windows 10 Pro for Workstations 64bit
CPU	Intel(R) Xeon(R) CPU E5-1603 v4 @ 2.80GHz
RAM	32.0GB



今後の課題：機能拡張, アルゴリズム改善, 被験者実験

[1] Takashi Ishio, Naoto Maeda, Kensuke Shibuya, and Katsuro Inoue. 2018. Cloned Buggy Code Detection in Practice Using Normalized Compression Distance. In 2018 IEEE International Conference on Software Maintenance and Evolution, ICSME 2018, Madrid, Spain, September 23-29, 2018. 591-594

[2] J. Li and M. D. Ernst. 2012. CBCD: Cloned Buggy Code Detector. In 2012 34th International Conference on Software Engineering (ICSE). 310-320. https://doi.org/10.1109/ICSE.2012.6227183

本研究はJSPS 科研費18H04094 の助成を受けたものです。