

# Automatic Generation of Code-clone Reference Corpus

이 효 섭

한양대학교 프로그래밍언어연구실

2012. 01. 17

PLASSE  
RESEARCH

ROSAEC

# 개념

- 코드클론이란?

소스 프로그램에서 구문적 생김새가 동일한 코드 조각

- 코드클론 탐지 도구의 성능 평가 요소

- ▶ 이식성(portability)

- ▶ 정확성(precision)

- ▶ 탐지성(recall)

- ▶ 확장성(scalability)

- ▶ 견고성(robustness)

- 코드클론 표본 집합체 (code clone reference corpus)

- ▶ 도구의 탐지성을 확인하기 위해서 참조하는 클론들의 모음

- ▶ 도구의 성능을 판별하는 기준으로 사용

# 기존 코드클론 표본 집합체

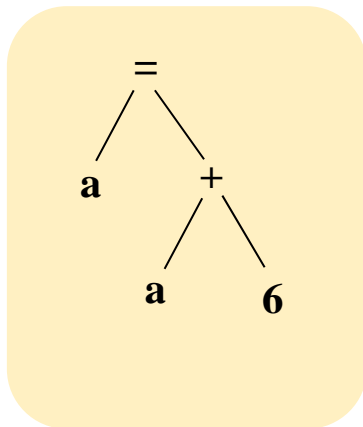
- **생성방법** [Bellon et al., TSE 2007]
  - 여러 도구에서 찾아낸 클론들을 수동으로 모아 생성
- **기존의 클론 탐지 기반 기술과 도구**
  - ▷ **Token** : Dup, CCFinder (1차원 비교: 오탐(false positive))
  - ▷ **PDG** : Duplex (모양이 다른 코드의 그래프/ 계량치가 동일할 수 있음)
  - ▷ **Function metrics** : CLAN : 오탐(false positive)
  - ▷ **Text** : Duploc (단순한 프로그램 포맷변경에 민감: 미탐(false negative))
  - ▷ **AST** : CloneDR

정확한 코드 클론을 찾기 위해서는  
코드의 구문구조 정보가 포함되어 있는 AST를 비교하는 방식이 적절하다.

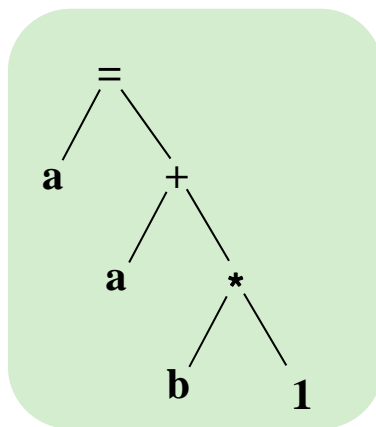
# 우리의 방식

- 코드내의 클론을 정확하고 빠짐없이 찾는 것이 목표
- 방법
  - ▷ 코드의 구문구조 정보가 포함되어 있는 AST를 비교
  - ▷ AST를 샅샅이 뒤져서 비교하되  
계산비용을 줄이기 위해 이미 방문한 트리는 재탐지하지 않음

$=(a,+(a,6))$

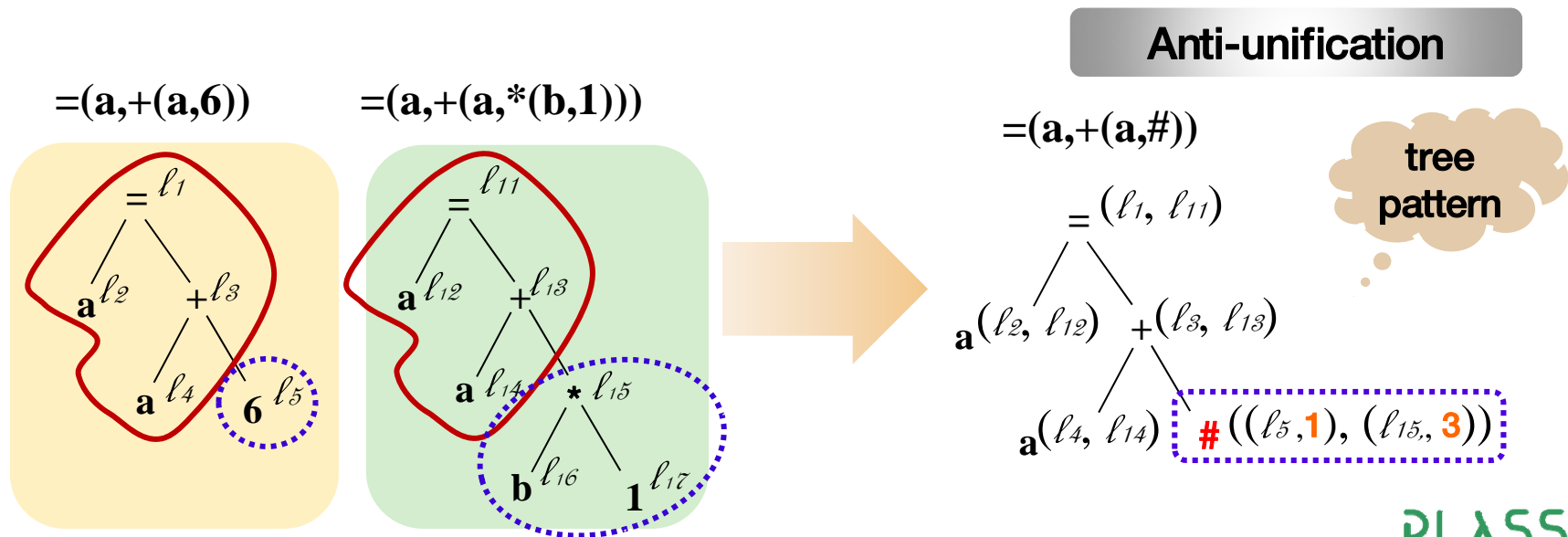


$=(a,+(a,*(b,1)))$



# 우리의 방식

- 코드내의 클론을 정확하고 빠짐없이 찾는 것이 목표
- 방법
  - ▷ 코드의 구문구조 정보가 포함되어 있는 AST를 비교
  - ▷ AST를 살짝이 뒤져서 비교하되  
계산비용을 줄이기 위해 이미 방문한 트리는 재탐지하지 않음



# 구현 및 실험대상

- 구현

- ▷ 구현언어 : Objective Caml 3.09, Python 2.5.1
- ▷ 파서
  - CIL 1.3.6 (C Intermediate Language)
  - Joust 0.8 for Java

- 실험대상

응용 프로그램	크기(MB)	파일 수	줄 수
netbeans-javadoc	0.69	97	14,301
eclipse-ant	1.35	149	29,880
eclipse-jdtcore	6.37	687	135,675
j2sdk1.4.0-javax-swing	8.32	533	202,943

# 우리 도구의 평가

- **정확성**
  - ▷ 오탐 없음
- **탐지성**
  - ▷ 특정 패턴을 제외하고 미탐 없음
- **표본 집합체를 생성하는데 걸린 시간**

응용 프로그램	우리도구
netbeans-javadoc	0h 01m 40s
eclipse-ant	0h 03m 56s
eclipse-jdtcore	2h 40m 11s
j2sdk1.4.0-javax-swing	5h 06m 09s

# Bellon의 클론 표본 집합체와의 비교

- 기존의 표본 집합체를 거의 대부분 포함한다.

응용 프로그램	기존 표본집합체	우리의 표본 집합체 자동 생성기	
		기본값	기본값 변경
netbeans-javadoc	55	43 (78.2%)	53 (96.4%)
eclipse-ant	30	28 (93.3%)	30 (100.0%)
eclipse-jdtcore	1,345	62 (91.0%)	62 (92.5%)
j2sdk1.4.0-javax-swing	777	36 (94.7%)	38 (100.0%)

5%

우리 도구에서 입력값은 총 4개로, 클론의 노드 수, 클론에서 일치하지 않는 부분의 개수, 일치하지 않는 부분의 크기와 최소 줄 수이다.



# Bellon의 클론 표본 집합체와의 비교

- 기존의 표본 집합체를 거의 대부분 포함한다.

응용 프로그램	기존 표본집합체	우리의 표본 집합체 자동 생성기	
		기본값	기본값 변경
netbeans-javadoc	55	43 (78.2%)	53 (96.4%)
eclipse-ant	30	28 (93.3%)	30 (100.0%)
eclipse-jdtcore	67	62 (91.0%)	62 (92.5%)
j2sdk1.4.0-javax-swing	38	36 (94.7%)	38 (100.0%)

우리 도구에서 입력값은 총 4개로,  
클론의 노드 수, 클론에서 일치하지 않는 부분의 개수, 일치하지 않는 부분의 크기와 최소 줄 수이다.

# Bellon의 클론 표본 집합체와 비교한 도구

- **CloneDigger** [Bulychev.P.E. and Minea.M., IWSC 2009]
  - ▷ Anti-unification 알고리즘 사용
  - ▷ 지원언어 : Python, Java, Lua

응용 프로그램	기존 표본집합체	CloneDigger	기존 표본집합체	우리의 표본집합체 자동 생성기
netbeans-javadoc	46	23 (50.0%)	55	43 (78.2%)
eclipse-ant	22	17 (77.3%)	30	28 (93.3%)
eclipse-jdtcore	60	19 (31.7%)	67	62 (91.0%)
j2sdk1.4.0-javax-swing	39	19 (48.7%)	38	36 (94.7%)

# 기본값 변경으로 탐지하는 클론의 예

- 최소 줄 수를 6 -> 5로 변경

응용 프로그램	기존 표본집합체	우리의 표본 집합체 자동 생성기	
		기본값	기본값 변경
netbeans-javadoc	55	43 (78.2%)	53 (96.4%)
eclipse-ant	30	28 (93.3%)	30 (100.0%)
eclipse-jdtcore	67	62 (91.0%)	62 (92.5%)
j2sdk1.4.0-javax-swing	38	36 (94.7%)	38 (100.0%)

# 기본값 변경으로 탐지하는 클론의 예

Reference number: 581

netbeans-javadoc/AutoCommentPanel.java : 315-327

```
315 gridBagConstraints = new java.awt.GridBagConstraints();
316 gridBagConstraints.gridwidth = java.awt.GridBagConstraints.REMAINDER;
317 gridBagConstraints.fill = java.awt.GridBagConstraints.HORIZONTAL;
318 gridBagConstraints.insets = new java.awt.Insets(8, 0, 0, 0);
319 buttonPanel.add(helpButton, gridBagConstraints);
320 gridBagConstraints = new java.awt.GridBagConstraints();
321 gridBagConstraints.weighty = 1.0;
322 buttonPanel.add(jPanel13, gridBagConstraints);
323 gridBagConstraints = new java.awt.GridBagConstraints();
324 gridBagConstraints.gridwidth = java.awt.GridBagConstraints.REMAINDER;
325 gridBagConstraints.fill = java.awt.GridBagConstraints.VERTICAL;
326 gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTH;
327 gridBagConstraints.weighty = 0.6;
```

• 최소 줄 수를 6 -> 5로 변경

①

②

netbeans-javadoc/AutoCommentPanel.java : 360-371

```
360 gridBagConstraints = new java.awt.GridBagConstraints();
361 int i = 8;
362 gridBagConstraints.gridwidth = java.awt.GridBagConstraints.REMAINDER;
363 gridBagConstraints.fill = java.awt.GridBagConstraints.HORIZONTAL;
364 gridBagConstraints.insets = new java.awt.Insets(i, 0, 0, 0);
365 buttonPanel.add(helpButton, gridBagConstraints);
366 gridBagConstraints = new java.awt.GridBagConstraints();
367 buttonPanel.add(jPanel13, gridBagConstraints);
368 gridBagConstraints = new java.awt.GridBagConstraints();
369 gridBagConstraints.gridwidth = java.awt.GridBagConstraints.REMAINDER;
370 gridBagConstraints.fill = java.awt.GridBagConstraints.VERTICAL;
371 gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTH;
```

PLASSE  
RESEARCH

# 탐지하지 못하는 클론 유형의 예

응용 프로그램	기존 표본집합체	우리의 표본 집합체 자동 생성기	
		기본값	기본값 변경
netbeans-javadoc	55	43 (78.2%)	53 (96.4%)
eclipse-ant	30	28 (93.3%)	30 (100.0%)
eclipse-jdtcore	67	62 (91.0%)	62 (92.5%)
j2sdk1.4.0-javax-swing	38	36 (94.7%)	38 (100.0%)

기존 표본집합체 190개에서 7개(3.7%)의 코드클론은 찾지 못함

# 탐지하지 못하는 클론 유형의 예

Reference number: 579

DocumentationSettings.java : 75-88

```
75 public ServiceType getServiceTypeOfDocument() {
76     JavadocSearchType.Handle st = (JavadocSearchType.Handle)getProperty( PROP_SEARCH );
77     JavadocSearchType t = null;
78     int x = 15;
79     if (st != null) {
80         x++;
81         t = (JavadocSearchType)st.getServiceType(); }
82     if (t == null) {
83         --x;
84         if (isWriteExternal()) {
85             x = x > 15 ? 15 : x;
86             return null; }
87     return (JavadocSearchType)Lookup.getDefault().lookup
            (org.netbeans.modules.javadoc.search.Jdk12SearchType.class); }
88     return t; }
```

DocumentationSettings.java : 192-201

```
192 public ServiceType getSearchEngine() {
193     JavadocSearchType.Handle searchType =
            (JavadocSearchType.Handle)getProperty(PROP_SEARCH);
194     JavadocSearchType type = null;
195     if (searchType != null) {
196         type = (JavadocSearchType)searchType.getServiceType(); }
197     if (type == null) {
198         if (isWriteExternal()) {
199             return null; }
200     return (JavadocSearchType)Lookup.getDefault().lookup
            (org.netbeans.modules.javadoc.search.Jdk12SearchType.class); }
201     return type; }
```

# 지금까지 한일과 할 일

- **지금까지 한 일**

- ▷ 정확하면서 빠짐없이 클론을 찾는 알고리즘 개발 및 구현
- ▷ 우리의 도구는 수동으로 생성한 기존 코드클론 표본 집합체의 클론들을 거의 대부분 포함하면서 자동으로 탐지

- **할 일**

- ▷ 기존 코드클론 표본집합체보다 정확히 얼마나 더 많은 클론을 찾아내는지 조사
- ▷ 지원 언어 확대
- ▷ 속도 개선
- ▷ 다른 응용 분야의 적용
  - 리팩토링, 소프트웨어 형상관리, 복사 후 붙이기 오류 찾기, 복제 감정평가