Erkennung von Typ-3 Klonen

Exposé

Elvis Raoul Wega Teubou

Universität Bremen
07. April 2009

1 Aufgabenstellung

Diese Arbeit befasst sich mit der Erweiterung des existierenden Kloneerkennungstools *clones*, mit dem anhand bestimmter Kriterien bessere Typ-3 Klone gefunden werden sollen. Die Bearbeitung dieser Aufgabe wird in 2 Schritten erfolgen.

1. Kriterien ermitteln, nach denen zwei oder mehrere Typ-1 bzw. Typ-2 Klone zu einem Typ-3 Klon zusammengefasst werden.

2. Implementierung eines Algorithmus, welcher anhand der ermittelten Kriterien eine Liste von Typ-1 bzw. Typ-2 Klone bearbeitet und am Ende die Typ-3 Klone hinzufügt.

2 Vorgehensweise

Aus der Analyse einiger bereits vorhandener Typ-3 Klone und der Durchführung einer Literaturrecherche werden sinnvolle Kriterien selektiert.
Wie in Abbildung 1 zu sehen ist der Algorithmus in 3 Schritte aufgeteilt. Im ersten Schritt wird zuerst eine Gruppierung der Liste von Typ-1 bzw. Typ-2 Klonen vorgenommen. Ziel dieser Gruppierung ist:

1. Das Rausfiltern von Dateien, die nur ein Klon-Paar enthalten.

2. Die Gruppierung von Klon-Paaren, deren Fragmente zu gleichen Dateien gehören.

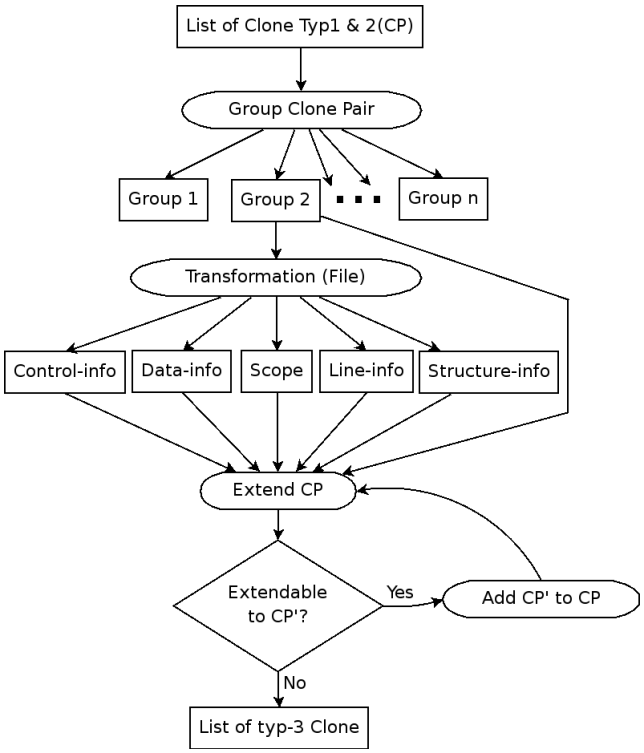

Abbildung 1: Dieses Diagramm stellt die Schritte des Algorithmus dar

Im zweiten Schritt werden nach der Gruppierung die entsprechenden Dateien transformiert, die aus der Gruppierung entstanden sind. Dabei wird für jeden Ausdruck (jede Zeile) folgende Informationen gespeichert:

Control-info speichert für jeden Ausdruck den vorigen und den nächsten Ausdruck, welcher ein Kontroll-Schlüsselwort (zum Beispiel: if) enthält.

Data-info speichert für jede Variable den vorigen und den nächsten Ausdruck, welcher diese Variable referenziert (setzen oder benutzen).

Structure-info speichert die Struktur jedes Ausdruckes (Hash Werte).

Scope speichert für jeden Ausdruck die Verschachtelungstiefet.

Line-info speichert für jeden Ausdruck die Zeilennummer.

Im dritten Schritt wird anhand der Kriterien geprüft, ob Typ-3 Klone in den jeweiligen Gruppen vorhanden sind. Wenn ja, werden sie in der Klon-Liste hinzugefügt.
Die folgenden Kriterien werden betrachtet:

- Blöcke
- Die Datenabhängigkeit wie in [1] beschrieben.
- Die Kontrollschlüssel wie in [1] beschrieben.
- Der absolute Abstand wird vom Benutzer festgelegt.
- Der relative Abstand. Der Abstand zwischen zwei Klon-Paaren darf nicht länger als das größte Fragment sein, weil sonst die Wahrscheinlichkeit gering wäre, die daraus resultierenden Klone als Typ-3 zu bezeichnen.

Mit dem folgenden Algorithmus werden Klon-Paare zu Typ-3 Klonen zusammengefügt:
Input: Liste mit n Typ-1,2 Klon-Paaren und *d_user* (bei absolutem Abstand)
Output: Liste von Typ-3 Klon-Paaren
initialer Abstand $d = 0$

1. Iteration durch die Klon-Paar-Liste
2. Hole das aktuelle Klon-Paar x (KPx) und betrachte ab dieser Position die nächsten Klon-Paare als neue Liste.

3. Iteration durch die neue Liste
4. Hole das aktuelle Klon-Paar y (KPy) aus der neuen Liste
5. Berechne den relativen Abstand *rd = max (df1KPx, df1KPy).*
   - *df1KPx* ist die Größe des ersten Fragments des Klon-Paares x
   - *df1KPy* ist die Größe des ersten Fragments des Klon-Paares y
6. Prüfe nach Kontrollschlüssel, Datenabhängigkeit und Blöcken, ob die beiden Klon-Paare zu einem Typ-3 Klon zusammengefügt werden können.
7. Wenn mindestens einer der in Schritt 6 durchgeführten Prüfungen positiv ist, wird d aktualisiert.
8. Wenn $d <= rd$ oder $d <= d\_user$ (Falls es mit dem Absoluten Abstand berechnet wird), werden die beiden Klon-Paare zusammengefügt. Das Ergebnis wird festgehalten als neues Klon-Paar x, dann weiter mit Punkt 4.
9. Wenn die neue Liste abgearbeitet ist, und die Klon-Paare zusammengefügt wurden, wird das Ergebnis in die Liste der Typ-3 Klone hinzugefügt. Dann wird mit Punkt 2 fortgeführt.

3 Erwartetes Ergebnis

Am Ende dieser Arbeit ist zu erwarten, dass die Typ-3 Klone qualitativ besser sind als vorher. Die Erkennung von Typ-3 Klonen darf nicht länger dauern als die von Typ-1 und Typ-2 Klonen.

4 Evaluation

Bei der Evaluation werden folgende Punkte betrachtet:

- Auswahl eines Szenarios. Dieses könnte zum Beispiel sein, dass ein Benutzer Klonen aus seiner Software entfernen möchte.
- Die Qualität der Klone wird evaluiert, indem die von dem neuen Algorithmus gefundenen Typ-3 Klone mit denen von dem vorhandenen Algorithmus verglichen werden. Zusätzlich können dafür die im Vorfeld bekannten Typ-3 Klonen zum Vergleich genommen werden.

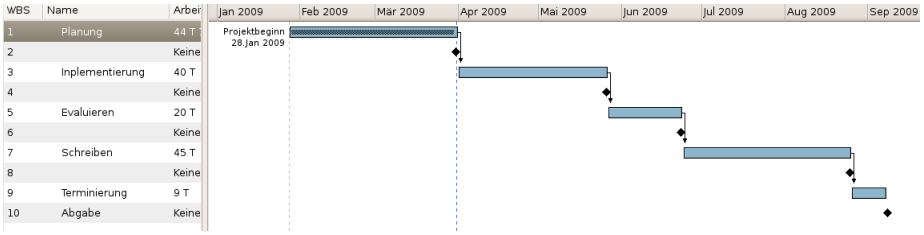

Abbildung 2: Dieses Diagramm stellt die Phasen der Arbeit dar

- Die benötigte Zeit für die Erkennung von Typ-3 Klonen wird mit der benötigten Zeit der Erkennung von Typ-1 und 2 Klonen verglichen. Der Vergleich der Klone wird vorerst manuell durchgeführt, wenn keine geeignetere Methode gefunden wird

5 Meilensteine

Die Abbildung 2 zeigt die verschieden Phasen und Meilensteine.

1. Ende der Planungsphase

   (a) Thema der Diplomarbeit festgelegt.
   (b) Arbeitsumgebung eingerichtet.
   (c) Literaturrecherche durchgeführt.
   (d) Exposé fertiggestellt.
   (e) Diplomarbeit angemeldet.

2. Ende der Implementierung

   (a) Gruppierung von Klon-Paaren implementiert.
   (b) Transformation von Dateien implementiert.
   (c) Erkennung von Typ-3 Klonen implementiert.

3. Ende der Evaluierung

   (a) Definition eines Anwendungsszenarios
   (b) Evaluation der Qualität der Typ-3 Klone
   (c) Evaluation der Schnelligkeit
   (d) Evaluation der Unterschiede zu den anderen Typ-3 Verfahren

4. Ende des Schreibens

(a) Das Schreiben ist abgeschlossen.
(b) Rechtschreibung und Grammatik geprüft.
(c) Diplomarbeit von einer aussenstehenden Person gelesen.

5. Fertigstellung der Diplomarbeit

   (a) Diplomarbeit gedruckt und gebunden.
   (b) Diplomarbeit abgegeben.

6 Risiken

- Die unzureichenden Ada Kenntnisse könnten dazu führen, dass die Umsetzung einiger Module länger dauert als geplant.

- Verfügbarkeit von Personen (wegen Kranheit oder Urlaub). Diese Personen werden für die Besprechung von Unklarheiten und geben von Feedbacks benötigt. Ein anderer kritischer wäre, wenn keine geeignete Person für das Korrekturlesen gefunden wird.

Literatur

[1] Yue Jia, David Binkley, Mark Harman, Jens Krinke, and Makoto Matsushita. KClone: A proposed approach to fast precise code clone detection. In *Workshop Proceeding of the 13th European Conference on Software Maintenance and Reengineering*, pages 12–16, 2009.