Mining Repair Actions for Automated Program Fixing

Matias Martinez, Martin Monperrus

To cite this version:

Matias Martinez, Martin Monperrus. Mining Repair Actions for Automated Program Fixing. Conférence GDR-GPL-CIEL, Jun 2012, Rennes, France, France. <hal-00696590>

HAL Id: hal-00696590

https://hal.inria.fr/hal-00696590

Submitted on 14 May 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Mining Repair Actions for Automated Program Fixing

Matías Martínez, Martin Monperrus

Context

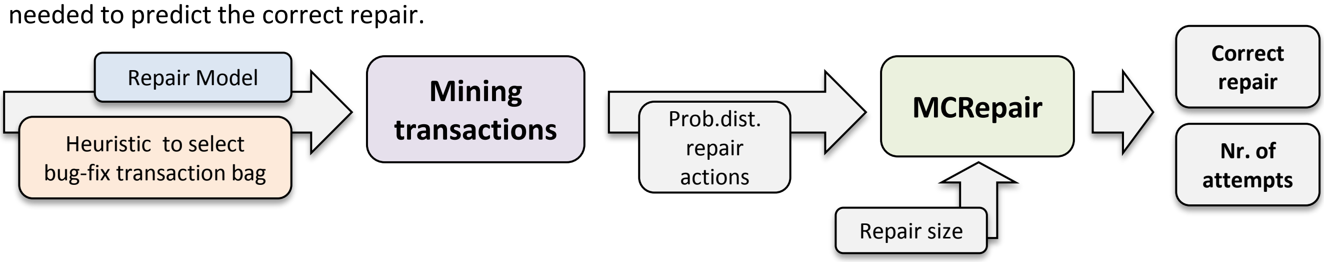
Repair actions are a kind of modification on source code that is made to fix a bug.
Automated program fixing consists of generating repair actions in order to fix bugs in an automated manner.

Objective

We present MCRepair, an automatic software repair process that minimizes the repair time.

Overview

We mine repair actions written by developers from bug-fix in software repositories (e.g. CVS, SVN or Git).
MCRepair uses the mined data to predict an unordered tuple of repairs actions and also counts the number of attempts needed to predict the correct repair.



Repair Model

Heuristic to select bug-fix transaction bag

Mining transactions

Prob.dist. repair actions

MCRepair

Repair size

Correct repair

Nr. of attempts

Methodology

Repair models are sets of semantic repair actions
We define two repair models:

- Coarse-grain: 48 repairs actions.
  E.g. *Statement_Insert*
- Fine-grain: 186 repairs action. E.g. *Condition_Expression_Changes_of_If_Statement*

Heuristics to select transaction bags T representative of software repair are based on:

- Commit text: contains words *fix, bug, path* (BFP).
- Syntactic features: T with N lines changes (N-LC).
- Semantic features: T with N semantic source code change (N-SC).

Small transactions are very likely to only contain a bug-fix and unlikely to contain a new feature.

MCRepair uses the probability distribution of repair actions from software repositories to maximize the likelihood of finding a correct repair action.

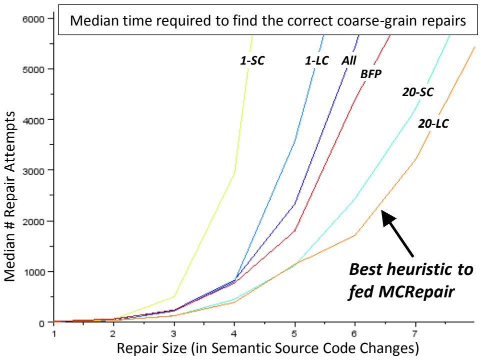
$$prob(repairAction_i) = \frac{cov(repairAction_i)}{\sum_j cov(repairAction_j)}$$

where cov($repairAction_i$) = percentage of transactions which include at least one $repairAction_i$

Mining transaction to get probabilities from repair actions based on the intuition: *Different definitions of “bug-fix transactions” yield different topologies for repair models.*

Evaluation

- Analyzed 14 open-source Java projects repositories: 89993 versioning transactions, 1377337 repair actions.
- Computed the median time required to find the correct repair of fix transactions for both repair model. MCRepair was fed with 6 heuristics.
- *Coarse grains*: bug-fix with 6 repair action < 2000 attempts and with 1 between 3 and 8.
- *Fine-grain*: bug-fix with 3 repair action < 22000 attempts, and with 1 between 3 and 16.



Median time required to find the correct coarse-grain repairs

Future work

- Coupling MCRepair with fault localization approaches to reduce the time to find correct repairs
- Instantiating repairs actions to produce more precise repairs
- Predict higher order repair actions: co-occurring repair actions