scorpio-e

What is Scorpio ?

Scorpio is a program that detects code clones (duplicate code) from Java source code. Scorpio detects the following types of code clones by using a special kind of program dependency graph.

- Exact code clones, like just copy and pasted.
- Parametrized code clones, which have different user-defined names such as variable names or method names.
- Gapped code clones are code clones including at least 1 gap, which is not duplicated with its corresponding code clones.

Condition of Use

- If you would like to detect code clones in the context of education or research, you can freely use Scorpio. Also, redistribution is not restricted in the cases.If you would like to use Scorpio in other cases, please contact by e-mail, higo@ist.osaka-u. (append "ac.jp").
- We don't have any guarantee about the quality of Scorpio and troubles caused by using it.
- The source code is open to the public on sourceforge.net.

Download

You can download Scorpio from the following link. Note that you can use Scorpio if you follow the above condition of use.

Scorpio (Version 201103030634)

Scorpio (Version 201010221745)

Scorpio (Version 201006141725)

Scorpio (Version 200910062138).

Scorpio (Version 200906191042).

How to use Scorpio

1. Download and unpack

Download the package of Scorpio from the above link, and unpack it. We recommend that unpacked directory is located on the file path that doesn't include white space. The followings are files included in the package.

- antlr.jar, masu.jar: library for analyzing Java source code.
- cfg.jar: library for generating control flow graph.
- pdg.jar: library for generating program dependency graph.
- commons-cli-1.1.jar: library for handling command line options (Commons CLI in Apache Project).
- scorpio.jar: program for detecting code clones.
- scorpio.jar: simple GUI for analyzing detected code clones.

2. Detect code clones

Detect code clones. The following command will be a good template.

> java -jar scorpio.jar -d *directory* -s *minimumsize* -o *resultfile*

- You need to put commons-cli-1.1.jar, antlr.jar, masu.jar, cfg.jar, and pdg.jar on the same directory.
- *directory* is a directory where the target files are.
- *minimumsize* is a minimum size that Scorpio. Size is specified by the number of nodes on PDGs. It is approximately the same as the number of statements and expressions in the source code.
- *resultfile* is a file that Scorpio store the detection result to. The file is in XML format.
- Scorpio need a lot of heap and stack memory. In the case that you are going to detect code clones from 10 KLOC or bigger software, we strongly recommend that you allocate a lot of memory to them.
  - -XmxAAAm: Allocate AAA MBytes memory as heap.
  - -XssBBBm: Allocate BBB MBytes memory as stack.

Also, the following options are available.

- -c *threshold*: Doesn't use program elements (statements and expressions) that have the same hash values with more than other *threshold* elements as the starting points of program slicing.
- -s *size*: Specify the minimum size of identified isormophics subgraphs (code clones). The size of a code clone means the number of program elements forming it. The default value is 7.
- -t *both*|*forward*|*backward*: Specify which types of slices are used for detecting code clones. *forward* means that only the forward slice is used, *backward* means that only

the backward slice is used, and *both* means that both of forward and backward slices are used. Using *both* will detect more code clones than *forward* or *backward*. However, *both* will requires more time to detect. The default value is *both*.

- -e *yes|no*: specify whether consecutive duplicate statements is merged or not. If consecutive duplicate statements are merged, the amount of false positives will be reduced. Also, Required computational cost for detection is reduced. The default value is*no*.
- -q *data|control|execution*: Specify which types of dependencies are used for building PDG. *data* means data dependency, *control* means control dependency, and *execution* means execution dependency. data dependency and control dependency are the same as ones in traditional PDG. Execution dependency exists between two program elements if one may be executed just after the other. Execution dependency is the same as *flow* in control flow graph. You can specify more than two terms by separating them with comma. For example, if you would like to use data and control dependencies, specify "*-t data,control*". The default value is *data,control,execution*.
- -u: *yes|no*: Specify whether only control nodes are used for slice points. *yes* means only control nodes are used, and *no* means all nodes are used. if only control nodes are used, detection time will decrease drastically. However, *yes* prevents some code clone from being detected.
- -v: *yes|no*: Specify whether verbose output or not. The default value is *no*.
- -w *thread*: specify the number of threads performing code clone detection. The number of logical CPUs is the optimal value. The default value is 1.
- -x *distance*: Specify the maximum distance that data dependency is constructed. If two statements are separated more than the *distance*, data dependency never be constructed. If you use this option appropriately, the detection will be precious and the detection time will shorten. The default value is java.lang.Integer.MAX_VALUE.
- -y *distance*: Specify the maximum distance that control dependency is constructed. The default value is java.lang.Integer.MAX_VALUE.
- -z *distance*: Specify the maximum distance that execution dependency is constructed. The default value is java.lang.Integer.MAX_VALUE.
- -pv *no|type|all*: Specify how variables are transformed before the matching process. *no* means that all variables are used as they are. *type* means that variables are transformed into their type name. For example, *i++* is transformed into *int++*. *all* means that all variables are transformed into the same token. The default value is *type*.
- -pi *no|type_with_arg|type_without_arg|all*: Specify how method invocations are transformed before the matching process. *no* means that all invocations are used as they are. *type_with_arg* means that the invoked method name was transformed into their return type, and arguments remains after the transformation. *type_without_arg* means that the invoked method name was transformed into their return type, and arguments do not remain after the transformation. *all* means that all method

invocations are transformed int the same token.

- -po *no*|*type*|*all*: Specify how operations (monomial, binomial, trinomial) are transformed before the matching process. *no* means that all operations are not transformed, they are used as they are. *type* means that all operations are transformed into its type name. *all* means that all operations are transformed into the same token. The default value is *no*.
- -pl *no*|*type*|*all*: Specify how literals are transformed before the matching process. *no* means that all literals are used as they are. *type* means that literals are transformed into their type names. *all* means that they are transformed into the same token. The default value is *type*.
- -pc *no*|*type*|*all*: Specify how casts are transformed before the matching process. *no* means that cast are used as they are. *type* means that casted items are transformed into their casting type names. *all* means that all casted items are transformed into the same token. The default value is *no*.
- -pr *no*|*all*: Specify how class references are transformed. *no* means that the class references are not transformed, they are used as they are. *all* means that all the class references are transformed into the same token. The default value is *no*.

3. Browse detected code clones

Browse detected code clones. The following command will be a good template.

> java -jar scorpioui.jar -i *resultfile*

- *resultfile* is a file that are generated in the step 2.