Calculation and Application of the Dynamic Function Level Dependencies in Java Programs

Ferenc Fischer

Dynamic dependencies are dependencies occurring during program runs on inputs. These dependency sets are smaller than the static dependency sets calculated from the source code, hence they are more informative and they characterize better the behaviour of the analyzed program than their static counterparts if the inputs are good.

We can define a distance based on the dynamic call graph (Dynamic Function Coupling - DFC) [1]. By putting the functions that are farther and farther from the selected one into the dependency set we can approximate the ExecuteAfter set generalized by us. After ER, EA and EB relations used in the definition of DFC we can compute $ImpactSet^{(d)}$, $EAfterSet^{(d)}$ set and $EBeforeSet^{(d)}$ set. These sets contain the impacting and/or impacted functions in a given direction and distance with the cut-off value d for a set of functions. $ImpactSet^{(d)}$ can be useful in change impact analysis, $EAfterSet^{(d)}$ in regression testing and $EBeforeSet^{(d)}$ in debugging.

These approximate sets can be easier computed than the static and dynamic statement level dependency sets. The intuition behind our approach is that the 'closer' the execution of a function is to the execution of another function in some of the runs of the program, the more likely it is that they are really dependent on each other. We used Jadys developed at our department to produce statement level slices. These slices were raised to function level to validate the precision of our method (precise slice). We examined how the precision and the recall of the approximate dependency sets change compared to those of the precise slices. Our results show that the precision of the sets that contain only close functions (with distance 1) is significantly (twice) larger than the precision of the sets created by the original conservative method.

References

[1] Árpád Beszédes, Tamás Gergely, Szabolcs Faragó, Tibor Gyimóthy, Ferenc Fischer. The Dynamic Function Coupling Metric and Its Use in Software Evolution, *Proceedings of 11th European Conference on Software Maintenance and Reengineering (CSMR'2007)*, pages 103-112, 2007.