

ניתוח מתקדם של שפות תכנות

תרגיל 4

להגשה עד 24/01/2018

- ניתן להגיש את התרגיל בזוגות.
- הקוד הרלוונטי לתרגיל נמצא ב

https://bitbucket.org/tausigplan/soft-prod16/src/master/demos/chaotic_iteration/

1. "May-Be-Garbage" Analysis

Extend the Python code used in class to demonstrate chaotic iteration with a "May-Be-Garbage" analysis. The analysis should infer for every program location a set of variable that may be garbage, in the sense that they may not be initialized. In an initial state, all variables may be uninitialized. An assignment operation initializes a value, but it may still be garbage if it is assigned to a value that depends on a variable that may be garbage. For example, after the following program:

```
x:=1;y:=x+z; z:=2;
```

The variable x is definitely not garbage, but y may still be garbage, since z may be garbage at the point where $x+z$ is assigned to y ;

1. Define the abstract domain, lattice operations (join and meet), and a transformer for assignments.
2. Implement the abstract domain for may-be-garbage analysis in a file called "mbg.py"
3. Demonstrate the analysis on two interesting examples where the analysis gives a precise result, in files "mbg_prog1.py" and "mbg_prog2.py"
4. Create in "mbg_prog3.py" an example where the analysis loses precision, and gives a false alarm. This means the analysis indicates that some variable may be garbage, when in fact it is not garbage in any program execution.

2. Interval Analysis

Extend the Python code used in class to demonstrate chaotic iteration with interval analysis, with widening and narrowing, as explained in class.

1. Implement the interval abstract domain for a single variable in a file called "interval.py". You should implement join, meet, and transformers, and also the widening and narrowing operators as defined in class.
2. Extend "chaotic.py" in a new file called "chaotic_wn.py" to perform widening and narrowing.
3. Demonstrate the analysis on two interesting examples where the analysis gives a precise result, in files "interval_prog1.py" and "interval_prog2.py" Your examples must demonstrate the need for widening and narrowing.
4. Create in "interval_prog3.py" an example where the analysis loses precision, even when using widening and narrowing.