Exercises on Range Analysis with Widening/Narrowing, with Solutions

Grégoire Sutre

http://www.labri.fr/~sutre/Teaching/SV/

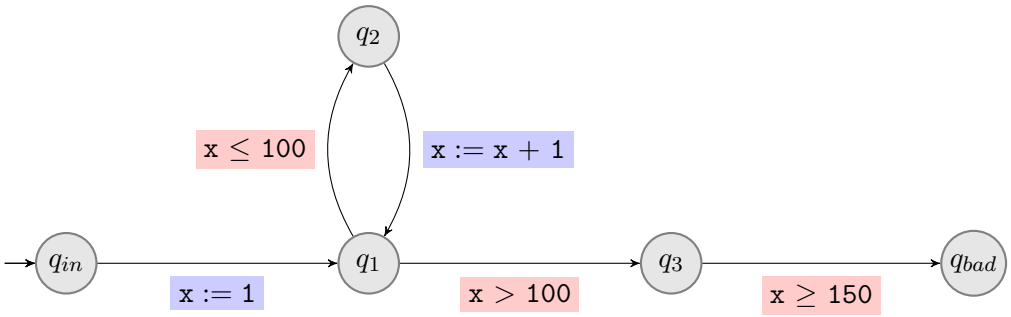
1 A Classical Example

We start this lab session with a classical example, which corresponds to the following C source code snippet:

```
int x;
for (x = 1; x <= 100; x++);
assert (x < 150);
```

Exercise 1 *Translate this program into a control-flow automaton, and perform,* ***manually****, range analysis on it, with widening and narrowing. Is the analysis successful?*

Solution. The control-flow automaton is depicted below. Its set of variables is $X = \{\mathtt{x}\}$, where x ranges over integers.



Let us analyze this control-flow automaton with the interval abstract domain. The widening ∇ is applied without delay (and at each location). Round-robin iteration proceeds as follows. Locations are processed in this order: $q_{in}, q_1, q_2, q_3, q_{bad}$.

	x	x	x	x
q_{in}	$\bot$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
q_1	$\bot$	$(1, 1)$	$(1, +\infty)^1$	$(1, +\infty)^3$
q_2	$\bot$	$(1, 1)$	$(1, +\infty)^2$	$(1, +\infty)$
q_3	$\bot$	$\bot$	$(101, +\infty)$	$(101, +\infty)$
q_{bad}	$\bot$	$\bot$	$(150, +\infty)$	$(150, +\infty)$

Let us explain how these abstract values are computed:

$$(1,1) \nabla ((1,1) \sqcup (2,2)) = (1,1) \nabla (1,2) = (1, +\infty) \quad (1)$$

$$(1,1) \nabla (1,100) = (1, +\infty) \quad (2)$$

$$(1, +\infty) \nabla ((1,1) \sqcup (2, +\infty)) = (1, +\infty) \nabla (1, +\infty) = (1, +\infty) \quad (3)$$

From the previously computed abstract value, a descending iteration with narrowing (and without delay) is performed as follows:

	x	x	x	x
q_{in}	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
q_1	$(1, +\infty)$	$(1, +\infty)^4$	$(1, 101)^7$	$(1, 101)$
q_2	$(1, +\infty)$	$(1, 100)^5$	$(1, 100)$	$(1, 100)$
q_3	$(101, +\infty)$	$(101, +\infty)^6$	$(101, 101)^8$	$(101, 101)$
q_{bad}	$(150, +\infty)$	$(150, +\infty)$	$\bot^9$	$\bot$

$$
\begin{aligned}
(1,+\infty)\,\Delta\,((1,1)\sqcup(2,+\infty)) &= (1,+\infty)\,\Delta\,(1,+\infty) &= (1,+\infty) &\quad (4)\\
(1,+\infty)\,\Delta\,(1,100) &= (1,100) &\quad (5)\\
(101,+\infty)\,\Delta\,(101,+\infty) &= (101,+\infty) &\quad (6)\\
(1,+\infty)\,\Delta\,((1,1)\sqcup(2,101)) &= (1,+\infty)\,\Delta\,(1,101) &= (1,101) &\quad (7)\\
(101,+\infty)\,\Delta\,(101,101) &= (101,101) &\quad (8)\\
(150,+\infty)\,\Delta\,\bot &= \bot &\quad (9)
\end{aligned}
$$

The abstract value in location q_{bad} is $\bot$, meaning that q_{bad} is not reachable, as the concretization of $\bot$ is the empty set. So the analysis is successful. The crucial point is that the analysis is able to infer that `x` is equal to 101 after the loop. ■

Exercise 2 *Implement the functions* `widen` *and* `narrow` *of the modules* `DomConstant`*,* `DomSign` *and* `DomInterval`*. Run* `make test-dom` *to check your implementation.*

Exercise 3 *Analyze the above program with* sai*, and compare the result with your manual analysis.*

Solution. The range analysis performed by sai coincides with the manual analysis performed in the previous exercise.

2 Analysis of the Lecture's Running Example

In this section, sai is applied to the lecture's running example, which is provided in the file `examples/aut/running_example.aut`. Various options are passed to sai in order to obtain the analyses presented in the lecture.

Exercise 4 *Use widening without delay. Is the analysis successful before descending iterations?*

Solution. The command to use is:

```
./sai.byte -v -domain interval examples/aut/running_example.aut
```

The analysis is not successful before descending iterations.

Exercise 5 *Still before descending iterations, find the least widening delay that makes the analysis successful. Is the analysis successful without widening?*

Solution. The command to use is:

```
./sai.byte -v -domain interval -widening-delay 11 .../running_example.aut
```

The analysis is successful before descending iterations with this delay. With smaller delays, the analysis is not successful before descending iterations.

Exercise 6 *Use widening and narrowing, both without delay. Is the analysis successful?*

Solution. The command to use is:

```
./sai.byte -v -domain interval .../running_example.aut
```

The analysis is successful (after descending iterations).

Exercise 7 *Use widening without delay and disable narrowing. Does the analysis converge? Is it successful?*

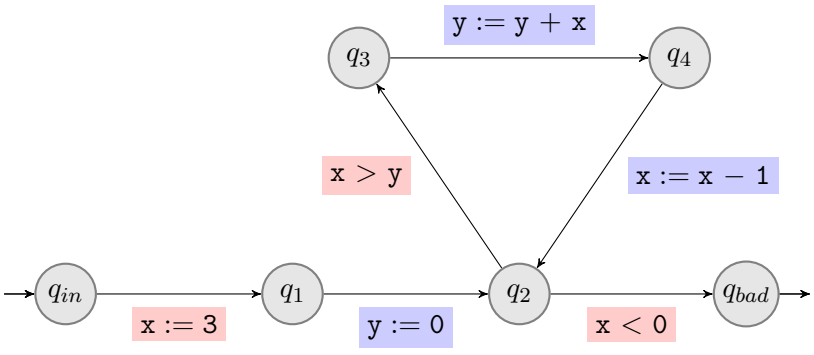
Solution. The command to use is:

```
./sai.byte -v -domain interval -narrowing-delay -1 .../running_example.aut
```

The analysis converges and is successful (after descending iterations).

3 Analysis of Another Example

We consider the control-flow automaton depicted below, with variables $X = \{\mathtt{x}, \mathtt{y}\}$, both ranging over integers, with initial location q_{in} and bad location q_{bad}.



This section performs range analysis of this control-flow automaton. As usual, an analysis will be called *successful* when the abstract value obtained for q_{bad} is $\bot$. Round-robin iteration shall use the following order on locations: $q_{in}, q_1, q_2, q_3, q_4, q_{bad}$. The first two exercises are independent. The third exercise depends on the result of the second exercise.

Exercise 8 *Apply the round-robin algorithm to compute the minimal fixpoint solution. Do* not *use widening and do* not *perform descending iterations. Is the analysis successful?*

Solution. We simply write $\top$ in place of $(-\infty, +\infty)$ to gain space.

	x	y	x	y	x	y	x	y	x	y	x	y
q_{in}	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
q_1	$\bot$	$\bot$	$(3,3)$	$\top$	$(3,3)$	$\top$	$(3,3)$	$\top$	$(3,3)$	$\top$	$(3,3)$	$\top$
q_2	$\bot$	$\bot$	$(3,3)$	$(0,0)$	$(2,3)$	$(0,3)$	$(1,3)$	$(0,5)$	$(0,3)$	$(0,5)$	$(0,3)$	$(0,5)$
q_3	$\bot$	$\bot$	$(3,3)$	$(0,0)$	$(2,3)$	$(0,2)$	$(1,3)$	$(0,2)$	$(1,3)$	$(0,2)$	$(1,3)$	$(0,2)$
q_4	$\bot$	$\bot$	$(3,3)$	$(3,3)$	$(2,3)$	$(2,5)$	$(1,3)$	$(1,5)$	$(1,3)$	$(1,5)$	$(1,3)$	$(1,5)$
q_{bad}	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$

The abstract value in location q_{bad} is $\bot$, meaning that q_{bad} is not reachable, as the concretization of $\bot$ is the empty set. So the analysis is successful.

Exercise 9 *Apply the round-robin algorithm with widening applied in location q_2 only. Do* not *perform descending iterations. Is the analysis successful?*

Solution. We simply write $\top$ in place of $(-\infty, +\infty)$ to gain space. The superscript $\star$ indicates the values that would have been larger (w.r.t. the partial order on intervals) if they had been widened.

	x	y	x	y	x	y	x	y
q_{in}	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
q_1	$\bot$	$\bot$	$(3,3)$	$\top$	$(3,3)$	$\top$	$(3,3)$	$\top$
q_2	$\bot$	$\bot$	$(3,3)$	$(0,0)$	$(-\infty,3)$	$(0,+\infty)$	$(-\infty,3)$	$(0,+\infty)$
q_3	$\bot$	$\bot$	$(3,3)$	$(0,0)$	$(1,3)^\star$	$(0,2)^\star$	$(1,3)$	$(0,2)$
q_4	$\bot$	$\bot$	$(3,3)$	$(3,3)$	$(1,3)$	$(1,5)$	$(1,3)$	$(1,5)$
q_{bad}	$\bot$	$\bot$	$\bot$	$\bot$	$(-\infty,-1)$	$(0,+\infty)$	$(-\infty,-1)$	$(0,+\infty)$

The abstract value in location q_{bad} concretizes to a non-empty set of environments (i.e., a non-empty subset of $\mathbb{Z}^X$), meaning that q_{bad} might be reachable. So the analysis is not successful.

Exercise 10 *Starting from the result of the previous exercise, perform a descending iteration with narrowing. Is the analysis successful?*

Solution.

	x	y	x	y	x	y
q_{in}	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
q_1	$(3,3)$	$\top$	$(3,3)$	$\top$	$(3,3)$	$\top$
q_2	$(-\infty,3)$	$(0,+\infty)$	$(0,3)$	$(0,5)$	$(0,3)$	$(0,5)$
q_3	$(1,3)$	$(0,2)$	$(1,3)$	$(0,2)$	$(1,3)$	$(0,2)$
q_4	$(1,3)$	$(1,5)$	$(1,3)$	$(1,5)$	$(1,3)$	$(1,5)$
q_{bad}	$(-\infty,-1)$	$(0,+\infty)$	$\bot$	$\bot$	$\bot$	$\bot$

The abstract value in location q_{bad} concretizes to an empty set of environments, meaning that q_{bad} is not reachable. So the analysis is successful.

4 Example Hunt

Exercise 11 *Find a control-flow automaton (or a program) that can be successfully verified with the sign domain and that cannot be verified with the interval domain (regardless of widening/narrowing use).*

Solution. The following program is successfully verified with the sign domain.

```
int x;
if (x != 0)
       assert (x != 0);
```

At the third line, just before the assertion, the concrete environments $\{\texttt{x} \mapsto -1\}$ and $\{\texttt{x} \mapsto 1\}$ are reachable. By convexity of intervals, this entails that every interval containing all reachable concrete environments before the assertion also contains the environment $\{\texttt{x} \mapsto 0\}$ which violates the assertion. So this program cannot be verified with the interval domain.

Exercise 12 *Find a control-flow automaton (or a program) that can be successfully verified* backwards *with the sign and interval domains and that cannot be verified* forwards *with these domains.*

Solution. The control-flow automaton `examples/aut/pre_only.aut` works.

Exercise 13 *Find a control-flow automaton (or a program) for which interval analysis with widening enabled and with narrowing disabled*[1] *has a diverging descending iteration.*

Solution. The control-flow automaton `examples/aut/infinite_descent.aut` works.

[1]The corresponding command is `./sai.byte -v -domain interval -narrowing-delay -1 <file>`.