**Range Analysis Example**

**Analysis Domain**

Continue the example from [Sets of States at Each Program Point](http://lara.epfl.ch/web2010/cc09:sets_of_states_at_each_program_point) but instead of allowing to denote all possible sets, we will allow sets represented by expressions   
\begin{displaymath}
    [L, U]
\end{displaymath}  
which denote the set $\{ x \mid L \le x \land x \le U \}$.

Thus, $L$is the lower bound and $U$is the upper bound. To ensure that we have only a few elements, we let   
\begin{displaymath}
    L, U \in \{ \mbox{MININT}, -128, 1, 0, 1, 127, \mbox{MAXINT} \}
\end{displaymath}

**Example:** $[0,127]$denotes integers between $0$and $127$.

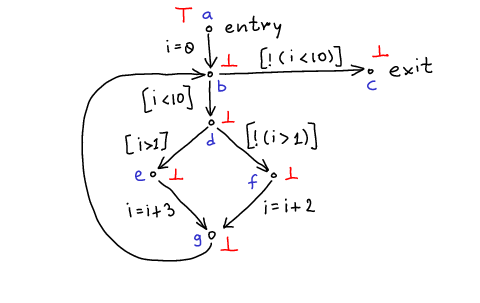
Furthermore, $[\mbox{MININT},\mbox{MAXINT}]$denotes all possible integers, and we denote it $\top$.

Instead of writing $[1,0]$and other empty sets, we will always write $\bot$, whose meaning is empty set, and therefore require that in $[L,U]$we have that $L \le U$.

So, we only work with a finite number of sets   
\begin{displaymath}
   1 + {7 \choose 2} = 22
\end{displaymath}  
Denote the family of these sets by $D$(domain).

**Initial Sets**

In the ‘entry’ point, we put $\top$, in all others we put $\bot$.

[](http://lara.epfl.ch/web2010/_detail/cc09:cfg-inc-abst-init.png?id=cc09%3Arange_analysis_example&cache=cache)

**Updating Sets**

Now we want to write the same set of equations. But because we have only a finite number of sets, we must approximate. We approximate sets with possibly larger sets.

$S_1 \sqcup S_2$denotes the approximation of $S_1 \cup S_2$: it is the set that contains both $S_1$and $S_2$, that belongs to $D$, and is otherwise as small as possible.

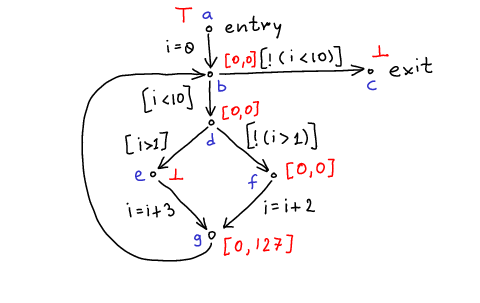
Also, we need to use approximate functions $T^\#(S,c)$that give result in $D$.

Now we replace the equations in [Sets of States at Each Program Point](http://lara.epfl.ch/web2010/cc09:sets_of_states_at_each_program_point) with approximate equations:

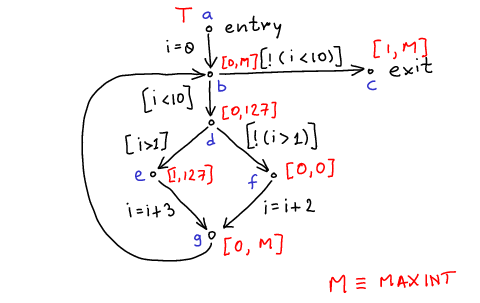
\begin{displaymath}
\begin{array}{l}
S^\#(a) = \top \\
S^\#(b) = T^\#(S^\#(a),i=0) \sqcup T(S(g),skip) \\
S^\#(c) = T^\#(S^\#(b),[\lnot(i<10)]) \\
S^\#(d) = T^\#(S^\#(b),[i<10]) \\
S^\#(e) = T^\#(S^\#(d),[i>1]) \\
S^\#(f) = T^\#(S^\#(d),[\lnot(i>1)]) \\
S^\#(g) = T^\#(S^\#(e),i=i+3) \sqcup T(S(f),i=i+2) 
\end{array}
\end{displaymath}

We solve the equations by starting in the initial state and repeatedly applying them. One can prove that (because this computation is monotonic), the sequence is growing and must terminate (in worst case when all sets are $\top$).

Sets after first iteration:

[](http://lara.epfl.ch/web2010/_detail/cc09:cfg-inc-abst-oneiter.png?id=cc09%3Arange_analysis_example&cache=cache)

Final values of sets:

[](http://lara.epfl.ch/web2010/_detail/cc09:cfg-inc-abst-final.png?id=cc09%3Arange_analysis_example&cache=cache)

Note:

* analysis terminates even if the original program does not terminate or takes very long (important for a compiler!)
* it proves that:
  + in all executions i=0 at point $f$
  + $i$cannot be zero at point $e$
  + value of $i$is always non-negative

With a larger domain $D$we can get better results, but analysis can take longer.