Program analysis in the framework of abstract interpretation requires:

1. specifying an abstract domain $\Sigma_A$ and a partial order $\subseteq_A$ on $\Sigma_A$;
2. giving meaning to the objects in the abstract domain by means of a concretization function $\gamma: \Sigma_A \mapsto 2^\Sigma$;
3. providing an abstraction function $\alpha: 2^\Sigma \mapsto \Sigma_A$, to enable abstracting the initial condition and transition relation;
4. constructing abstract operators necessary for forward propagation:
   - conjunction (meet) $\cap_A$;
   - disjunction (join) $\cup_A$;
   - quantifier elimination;
   - check for inclusion $\subseteq_A$;
5. (possibly) constructing a widening operator.

Many different abstract domains have been studied. Some examples are

- Linear equalities [6]
- Intervals [3]
- Difference-bound matrices [7]
- Octagons [8]
- Octahedra [1]
- Polyhedra [4]
- Uninterpreted functions [5]
- Boolean algebra over a set of predicates [2]

1. For three of the domains above give/describe:
   (a) $\Sigma_A, \subseteq_A$: state what are the top and bottom elements in the lattice; discuss representation of the elements in $\Sigma_A$; state whether $\Sigma_A$ is finite or infinite.
   (b) $\gamma: \Sigma_A \mapsto 2^\Sigma$;
   (c) $\alpha: 2^\Sigma \mapsto \Sigma_A$;
(d) $\cap A$, $\cup A$: quantifier elimination procedure, $\square A$: describe how the operations can be performed; for $\cap A$ and $\cup A$, discuss whether the operation is exact or an approximation, that is, whether $\gamma(a_1) \land \gamma(a_2) \equiv \gamma(a_1 \cap_A a_2)$ and $\gamma(a_1) \lor \gamma(a_2) \equiv \gamma(a_1 \cup_A a_2)$

(e) a widening operator if necessary; if not necessary explain why.

2. Briefly discuss some advantages and disadvantages of the three domains.

3. Compute the strongest invariant using forward propagation in each of your three abstract domains for the following program

```plaintext
i=0, j=0, k=0
while i <= 4 do
  <i,j,k> = <i+1, j+3, k+i+j
```

References


