



untrusted code is accompanied by a proof for its safety w.r.t. to some safety  
property and the code receiver has just to generate  $\text{code}(\text{prop})$

## 2    **Framework**





```
public class ListArray {
    Object[] list;
    //@requires list != null;
    //@ensures \result == (\exists s int i; 0 <= i && i < list.length &&
        list[i] == o );
    public boolean isElem(Object obj)
    {
        int i = 0;
        //@loop_modifies i;
        //@loop_invariant i <= list.length && i
```

loop frame condition, which declares

`nresult = 1`

`()`

`9var(0):`



the loops in a metho







## 6 Comparison between source and bytecodes proofs

The purpose of this section is to give a comparison between bytecode and proof obligations. In particular, we illustrate ~~the~~ the proof obligations of the example program in Fig.2.

We ~~the~~ the relationship between the 2source code proof obligations generated by the 2standard

Hypothesis on bytecode:	Hypothesis on source level:
<code>l v[2]_at.ins_20</code> <code>len(#19(l v[0]))</code>	<code>i_at.ins_26</code> <code>len(ListArray:l</code>

## References

- [1] A. V. Aho, R. Sethi, and J. D. Ullman. *Compilers-Principles, Techniques & Tools*. Addison-Wesley, 1976.

[15] G. C. Necula and P. Lee. The design and implementation of