Tommy Ingulfsen
July 8, 2003

Invariant extensions to AUTOGAP

1 Introduction

The file `invariant11.pl` holds the heuristics for discovering loop invariants expressing bounds for loop counters. An example of such an invariant is `I>=1 & J<=101` in the Polish Flag problem. Here we give a brief account of how AUTOGAP discovers these invariants.

2 Algorithm

This is the pseudocode for this part of the invariants generation:

```
mono = any monotonic variable (always associated with a loop)

init_val = mono's initial value

check that mono does not change between initialisation and the loop

check that mono is part of the loop's exit condition

if mono is monotonically increasing then
    create an invariant of the form ''mono >= init_val''

    else create an invariant of the form ''mono <= init_val''
```

It may seem unnecessarily conservative to insist upon using the unchanged initial value of `mono` in order to ascertain the bound, but this rule works fine in practice. As usual, there is more than one way to define a heuristic.

There is often another conjunct being generated, namely that specifying the upper/lower bound for counter variables (as opposed to its initial value). The code for this has been described in the dissertation and resides in the other invariant generation files.

[*] Notes in this series are for ϵ-baked ideas, for $1 \geq \epsilon \geq 0$. Only exceptionally should they be cited or distributed outwith the NuSPADE project (EPSRC GR/R24081).