USING DISCRETE LOOPS FOR EASY COMPREHENSION OF ALGORITHMS

ROLAND LIEGER AND JOHANN BLIEBERGER

DEPARTMENT OF AUTOMATION (183/1)
TECHNICAL UNIVERSITY VIENNA
TREITLSTR. 3/4
A-1040 VIENNA
AUSTRIA

PHONE: (43) 1-58801-8193
FAX: (43) 1-5863260
EMAIL: RLIEGER@AUTO.TUWIEN.AC.AT
EMAIL: BLIEB@AUTO.TUWIEN.AC.AT

Abstract

In this paper *discrete loops* are introduced to narrow the gap between general loops (e.g. while- or repeat- loops) and for- loops. While for-loops are easy to understand and guaranteed to complete they often lack the expressive power needed to elegantly express an algorithm. General loops on the other hand cover all possible iteration-conditions but all too often the computation of the condition is hidden within the remaining code, making it difficult to understand the algorithm. Furthermore it is hard to determine whether a general loop completes at all.

Discrete loops combine the advantages of for-loops and general loops. While they are easy to understand and guaranteed to complete they have sufficient expressive power to be used in many places that would otherwise require general loops. Thus they are of great value in understanding algorithms.

Furthermore discrete loops permit the compiler to automatically perform checks for programming errors. In addition to that it is possible to compute an upper bound for the number of iterations of a discrete loops at compile-time; an important feature for real-time programming.

In the following we will give an example of a discrete loop used for binary search in a sorted table:

```
function binary_search(item: in element) return integer is
  l: integer := list'first;
  u: integer := list'last;
  m: integer;
begin
discrete(l, u) new (l, u) := (l, (l+u)/2-1) | ((l+u)/2+1, u)
    with d := u-l+1 new d <= d/2 loop
  m := (u+l)/2;
  if item < list[m] then
    u := m-1;
  elsif item > list[m] then
    l := m+1;
  else
    return m;
  endif
end loop;
end binary_search;
```

Key words and phrases. Discrete Loops, Teaching of Algorithms.
Supported by the Austrian Science Foundation (FWF) under grant P10188-MAT.