Master Parisien de Recherche en Informatique 2010–2011
Proof Assistants

TD/TP 8 – Verifying Imperative Programs

2011-02-15

The goal is to verify some C functions from a string-manipulation library by using Frama-C + Jessie + Why for computing weakest preconditions and Alt-Ergo for checking them. The tools can be run with `frama-c -jessie file.c`.

Complete the specification of these functions. Use Alt-Ergo on the generated verification conditions. Fix the implementation of the functions if needed. Add some other functions and their specification.

```
/*@
axiomatic string {
  logic integer strlen_{L}(char *src);
  axiom strlen_inside{L}:
    \forall char *src; \forall integer x;
    0 <= x < strlen_(src) ==> src[x] != 0;
  axiom strlen_end{L}:
    \forall char *src; src[strlen_(src)] == 0;
}
*/

/*@
requires \valid_range(src, 0, strlen_(src));
ensures \result == strlen_(src);
*/
int strlen(char *src)
{
  int res = 0;
  /*@ loop invariant 0 <= res && res <= strlen_(src);
      loop variant strlen_(src) - res; */
  for (; src[res]; ++res);
  return res;
}

void strcat(char *dst, const char *src)
{
  dst += strlen(dst);
  for (; *src; ++src, ++dst) *dst = *src;
  *dst = 0;
}
```

Remarks:

- C strings are usually represented by a pointer to their first character, and they extend till character 0 inclusive.
- The `\valid_range`(p, i, j) predicate states that pointer p is inside an allocated memory block big enough for all the pointers $p + i \ldots p + j$ to be valid.
- Type `integer` is the type of unbounded integers ($\mathbb{Z}$) while type `int` is the type of machine integers; they can overflow.