

# Abstract Semantic Differencing for Numerical Programs

**Abstract.** We address the problem of correlating closely related versions of a program.

## 1 Introduction

## 2 Overview

In this section, we informally describe our approach with an example program.

## 2.1 Motivating Example

```
static void
print_numbers (long first, long step, long last, ...)
{
    long i;
    for (i = 0; /* empty */; i++) {
        long x = first + i * step;
        if (step < 0 ? x < last : last < x) break;
        if (i) fputs (separator, stdout);
        printf (fmt, x);
    }
    if (i)
        fputs (terminator, stdout);
}
```

---

coreutils seq.c v6.9

```
static void
print_numbers (long first, long step, long last, ...)
{
    bool out_of_range = (step < 0 ? first < last : last < first);
    if (! out_of_range) {
        long x = first;
        long i;
        for (i = 1; /* empty */ ; i++) {
            printf (fmt, x);
            if (out_of_range) break;
            x = first + i * step;
            out_of_range = (step < 0 ? x < last : last < x);
            if (out_of_range){
                bool print_extra_number = false;
                ... // print_extra_number is decided here
                if (! print_extra_number) break;
            }
            fputs (separator, stdout);
        }
        fputs (terminator, stdout);
    }
}
```

coreutils seq.c v6.10

**Fig. 1.** Original and patched version of coreutils seq.c's print\_numbers procedure