

# ICFP Programming Contest 2019

## Parallel Wrapping

Version 2.0

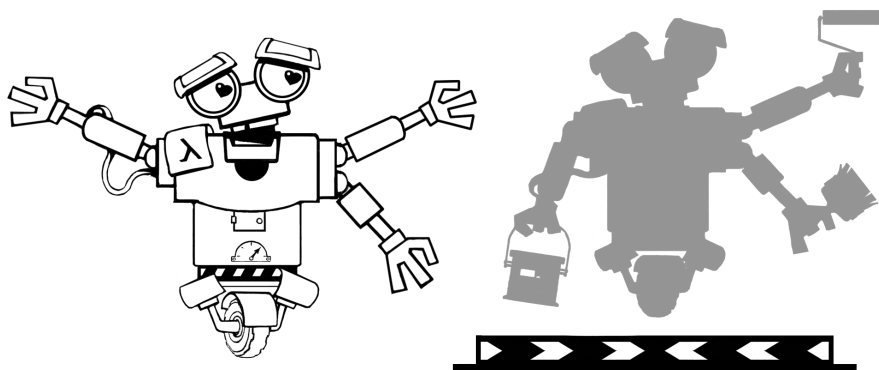
ICFP Programming Contest Organisers

June 22, 2019, 0:00am UTC

The sizes of the maps that need wrapping grow larger, and our worker-wrapper could use some help. This is made possible in a new pack of maps featuring a new kind of booster and revealing the purpose of the mysterious points from the previous tasks.

### New booster: Cloning

Code: C



**Effect:** With this booster, a worker-wrapper can replicate itself by creating a *replica*. Unfortunately, the support for the cloning technology is still not very mature, and the cloning booster can be only used in designated *spawn points*, marked on a map with the code X. A replica of a worker-wrapper created this way acts independently of its original and may create more replicas using the cloning boosters, thus providing parallel wrapping speed-up. Boosters collected by any replica are ~~immediately~~ then available to use by *any other worker-wrapper*,<sup>1</sup> with the effect applied to the user only.

Several worker-wrappers may occupy the same location. Cloning takes one time unit, and the created replica starts acting from the next turn from the location of the used spawn point. If several worker-wrapper replicas exist concurrently, they make their steps in the same time unit, in the order they were spawned.

### Extended task format

BoosterCode ::= ... | C

### Extended solution format

A solution can now provide a non-empty list of routes for each worker-wrapper replica, separated by #. The next route from the list is assigned to a newly created replica according to the spawning

<sup>1</sup>A moment when a booster becomes available is determined by the spawn order and is described in Appendix A.

order. The ends of the routes don't have to be aligned: if one replica exhausts its route description early, it will simply wait until all other worker-wrappers finish their executions.

```
action ::= ...
        | C          (perform the clone operation, spawning a new worker-wrapper)
actions ::= rep (action)
solution ::= repSep (actions, "#")
```

### Example

An example task with cloning and parallel wrapping is available at

<https://icfpcontest2019.github.io/download/part-3-clones-examples.zip>

### More tasks

The final pack with 80 maps can be downloaded at

<https://icfpcontest2019.github.io/download/part-3-clones.zip>

The contestants could notice that most of the maps we've seen in the past have spawn points installed, but, apparently, the supply of cloning boosters on those maps has been exhausted by the lambda miners working there previously. Who knows, maybe we will manage to make some use of those spawning facilities in the future.

## A Parallel Collection and Use of Boosters

A worker-wrapper collects a booster from a location *before* performing any action in it (*i.e.*, *not* after stepping into it in the previous turn). In the presence of parallel replicas, this leads to the following effect: a booster collected by a replica  $R_i$  that has stepped into a location containing it in a turn  $N$

- can be used by any replica  $R_j$ , such that  $i \leq j$ , starting from the turn  $N + 1$ ;
- can be used by any replica  $R_k$ , such that  $k < i$ , starting from the turn  $N + 2$ .