A Novel Approach to Automated Evaluation of Programming Assignments

Sujit Kumar Chakrabarti

IIITB

November 20, 2021

Outline for section 1

Introduction

AEPA System

Why Automated Evaluation?

- Online learning platforms: Coursera, Udacity, EdX ...
- Online programming contests: ACM ICPC, HakerEarth, HackerRank, CodeChef ...

Why Automated Evaluation?

- Online learning platforms: Coursera, Udacity, EdX ...
- Online programming contests: ACM ICPC, HakerEarth, HackerRank, CodeChef ...
- Introductory programming courses

Why Automated Evaluation?

- Online learning platforms: Coursera, Udacity, EdX ...
- Online programming contests: ACM ICPC, HakerEarth, HackerRank, CodeChef ...
- Introductory programming courses
- Error prone, labour intensive, repetitive

Automated Evaluation

1. Speed
2. Scalability
3. Objectivity
4. Transparency

Outline for section 2

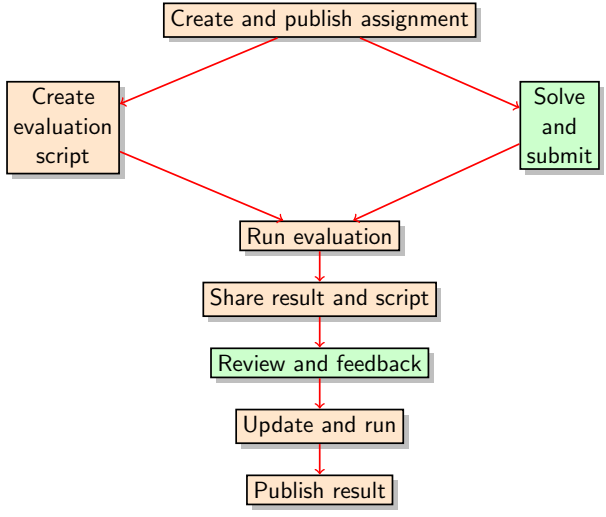
Introduction

AEPA System

Automated Evaluation System

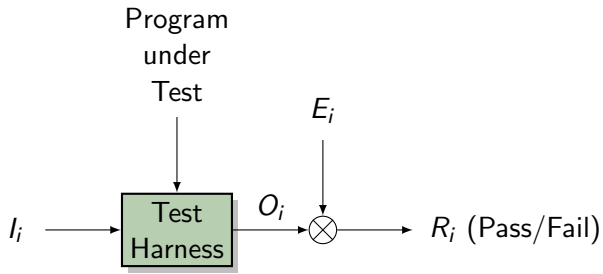
- Automatically evaluates programming assignments using testing
- Several human weeks $\longrightarrow$ a few seconds
- Has enabled more frequent, deeper formative assessments with shorter feedback cycles

AEPA Workflow



Testing

A Test Setup

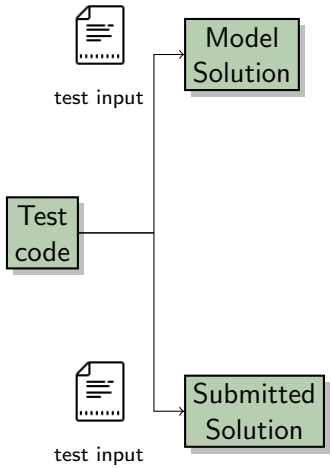


I_i	Test input
E_i	Expected output
O_i	Actual output
R_i	Test result

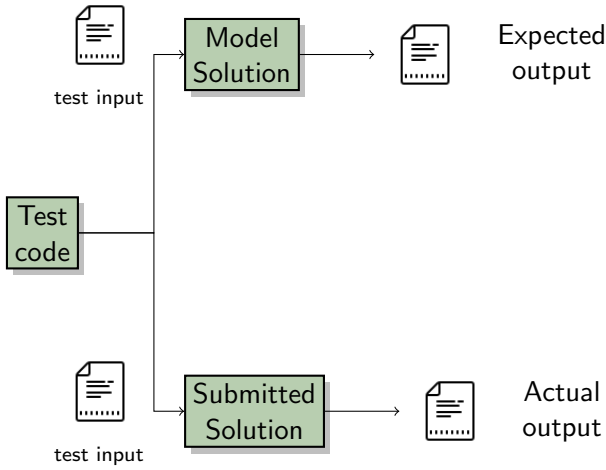
Assigning Marks:

$$M = \sum k_i R_i$$

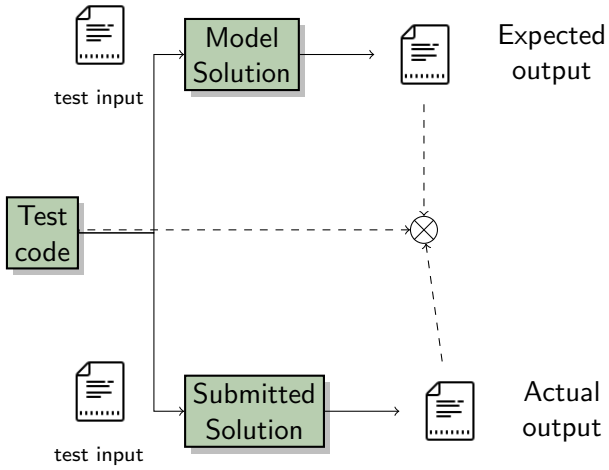
AEPA Artefacts



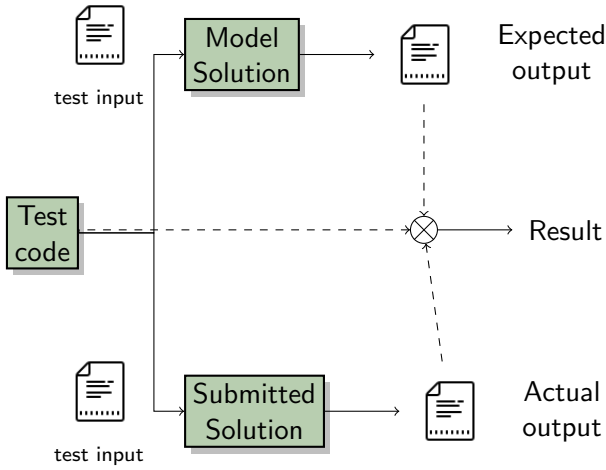
AEPA Artefacts



AEPA Artefacts

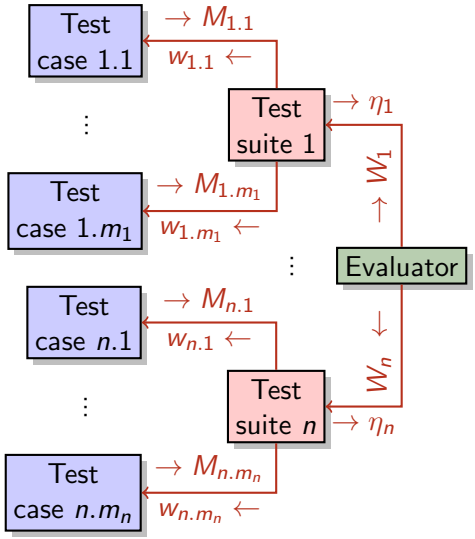


AEPA Artefacts



Design

System Architecture



Design

Types of Programs

- Programs with input/output
- Programs using/computing values
- Functions returning values
- Functions with input/output
- Questions about structural properties

Design

`evaluate.py` API[1]

- `equals`
- `eval_matfun`
- `eval_named_proc_1`
- `eval_named_proc_2`
- `eval_f_calls_g`
- `is_recursive`
- `is_inner_function`
- `num_of_classdefs`
- `num_of_classdefs`

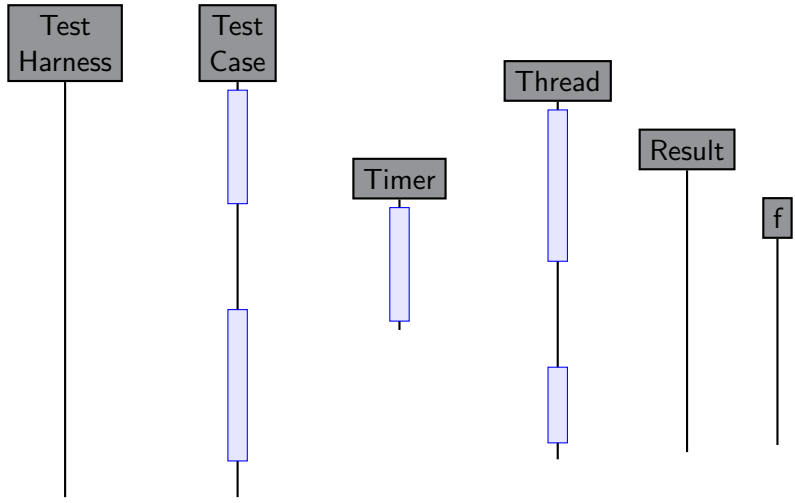
[1] https://github.com/sujitkc/evaluate/

Design

Dealing with Infinite Loops – Failing Tests

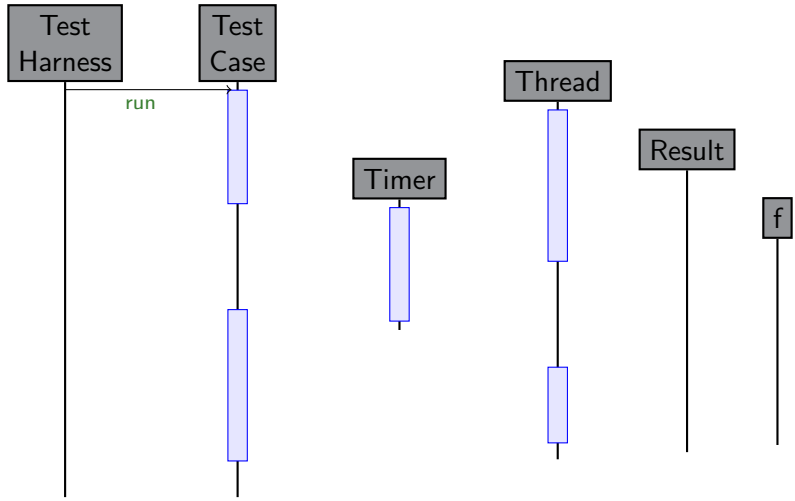
Design

Dealing with Infinite Loops – Failing Tests



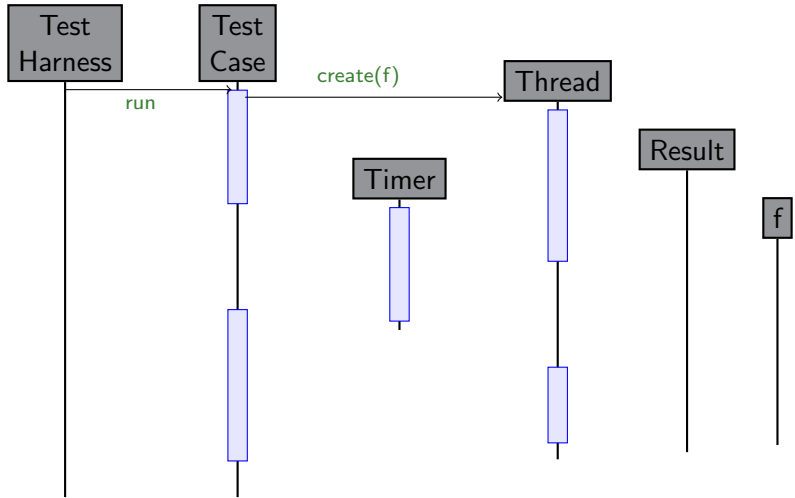
Design

Dealing with Infinite Loops – Failing Tests



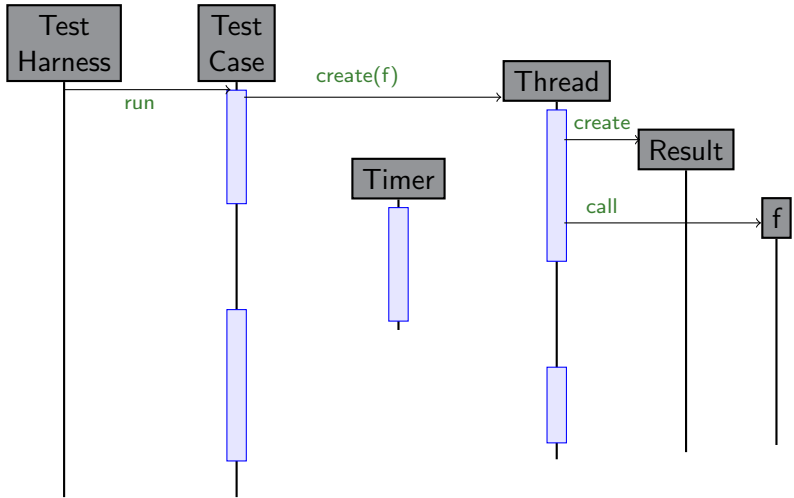
Design

Dealing with Infinite Loops – Failing Tests



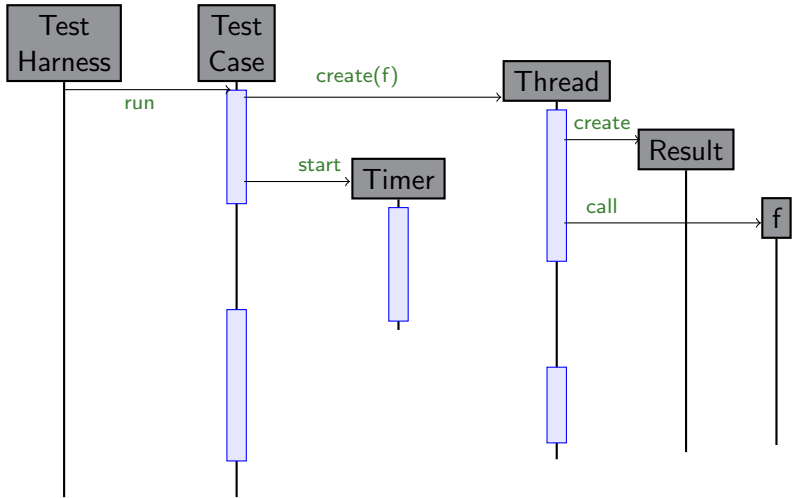
Design

Dealing with Infinite Loops – Failing Tests



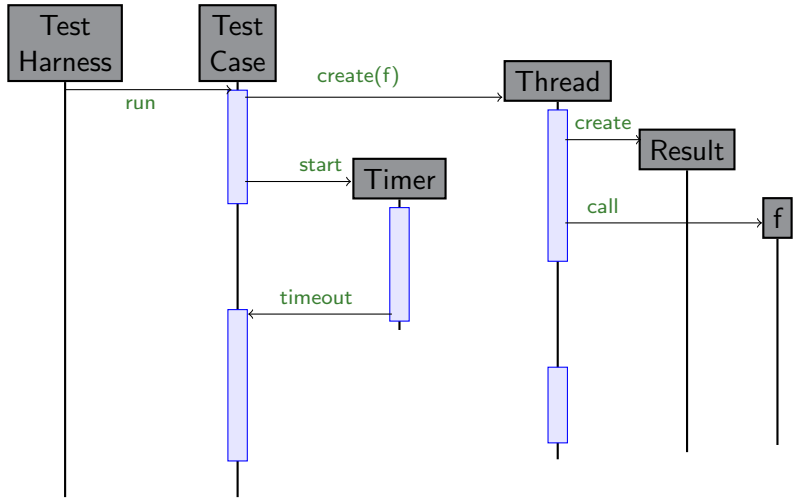
Design

Dealing with Infinite Loops – Failing Tests



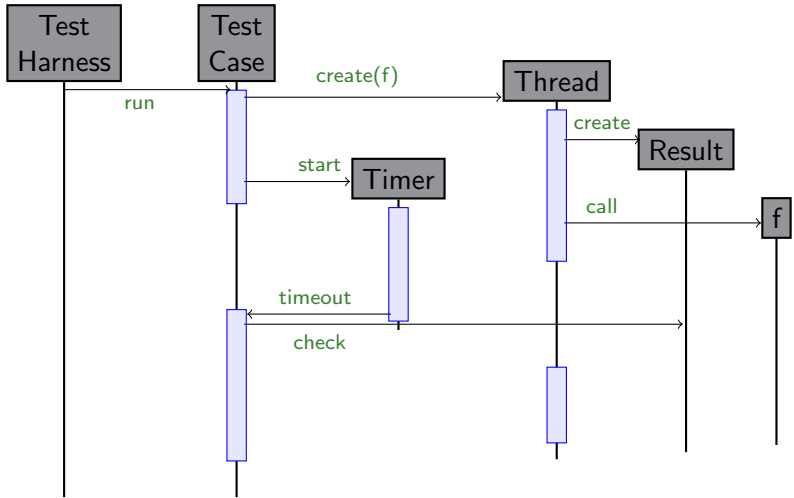
Design

Dealing with Infinite Loops – Failing Tests



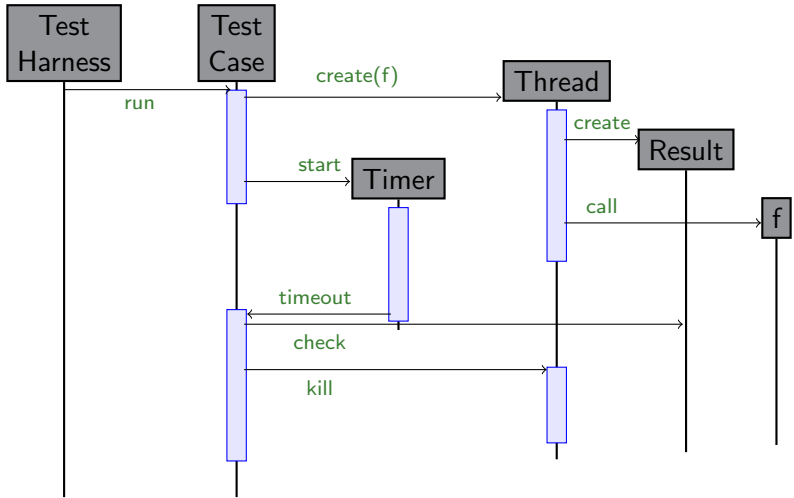
Design

Dealing with Infinite Loops – Failing Tests



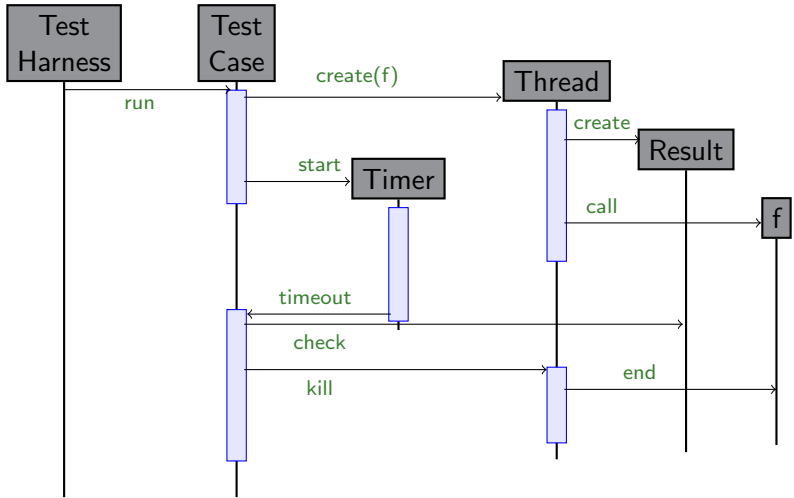
Design

Dealing with Infinite Loops – Failing Tests



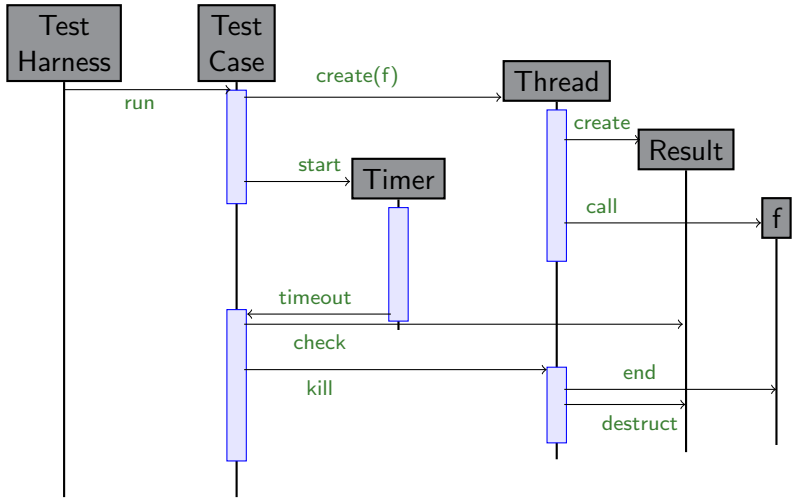
Design

Dealing with Infinite Loops – Failing Tests



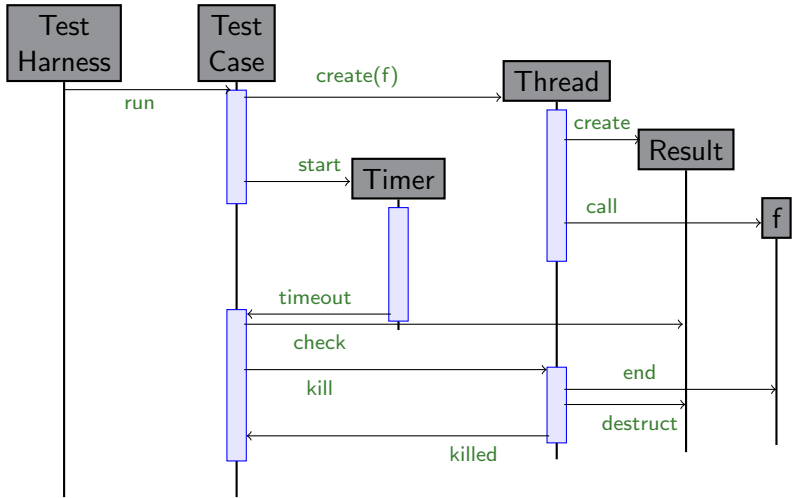
Design

Dealing with Infinite Loops – Failing Tests



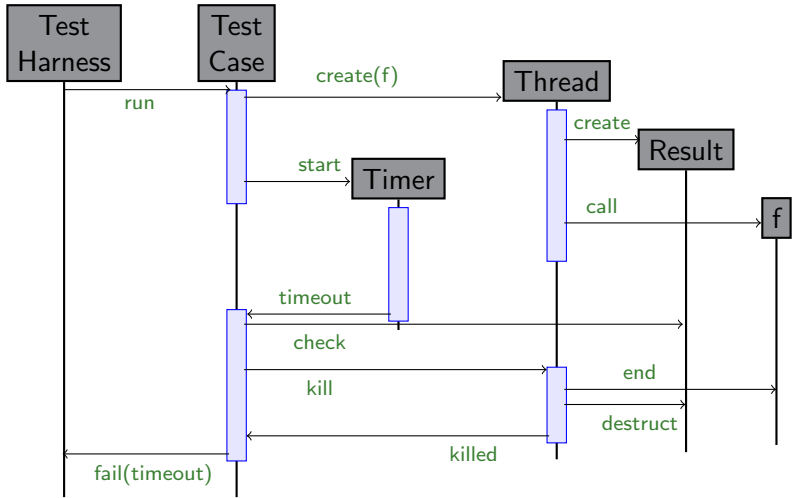
Design

Dealing with Infinite Loops – Failing Tests



Design

Dealing with Infinite Loops – Failing Tests

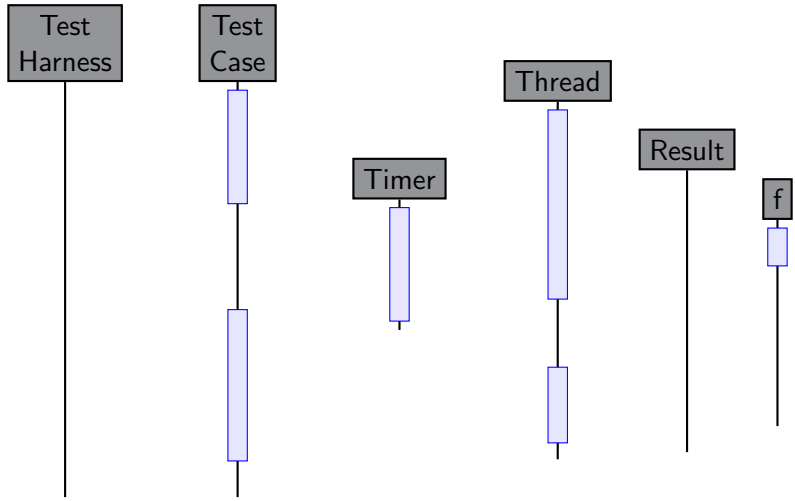


Design

Dealing with Infinite Loops – Succeeding Tests

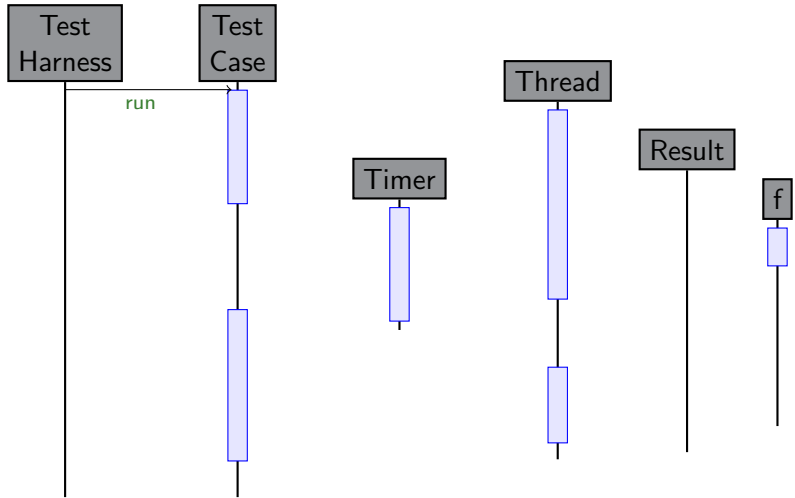
Design

Dealing with Infinite Loops – Succeeding Tests



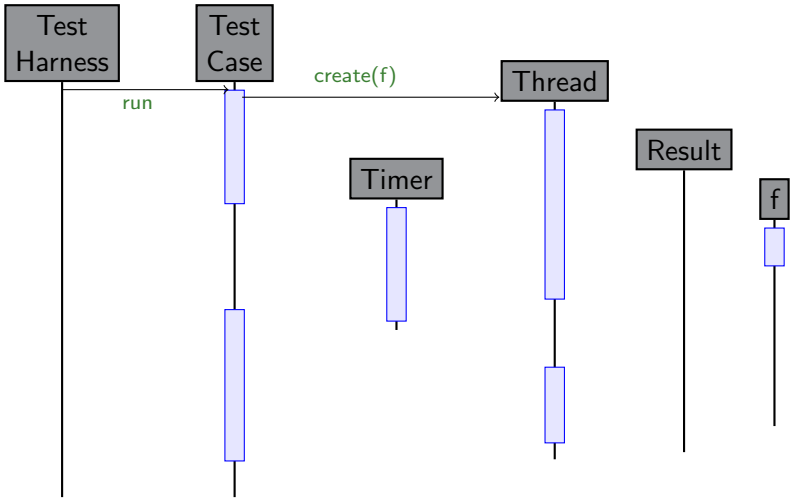
Design

Dealing with Infinite Loops – Succeeding Tests



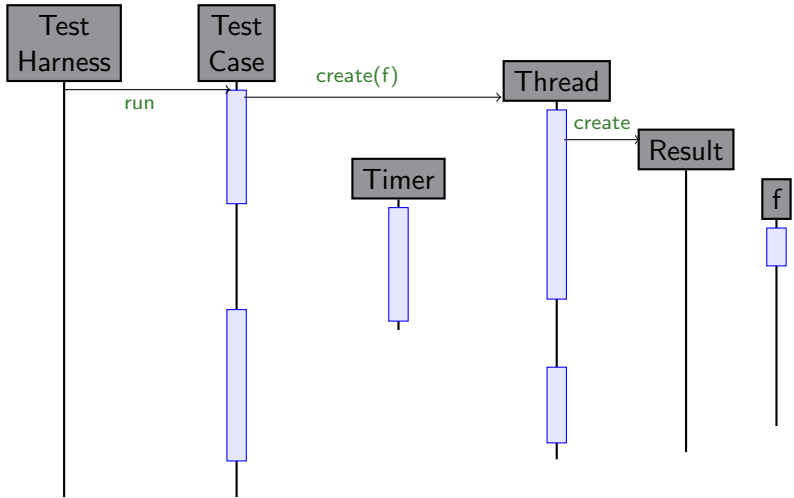
Design

Dealing with Infinite Loops – Succeeding Tests



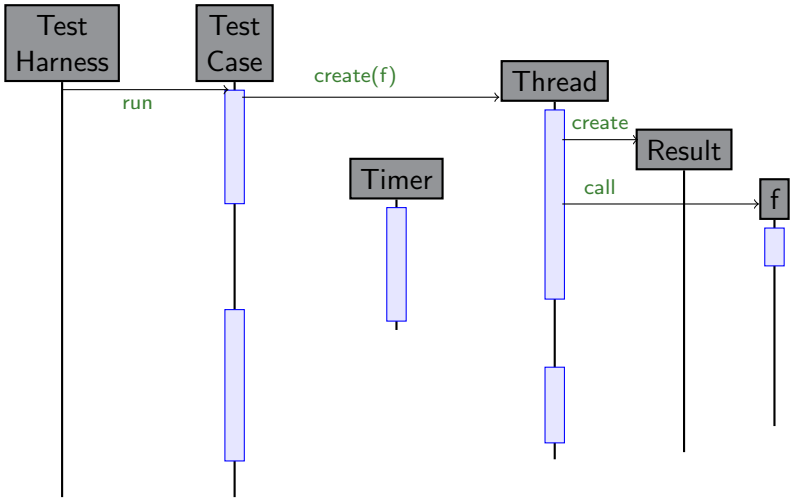
Design

Dealing with Infinite Loops – Succeeding Tests



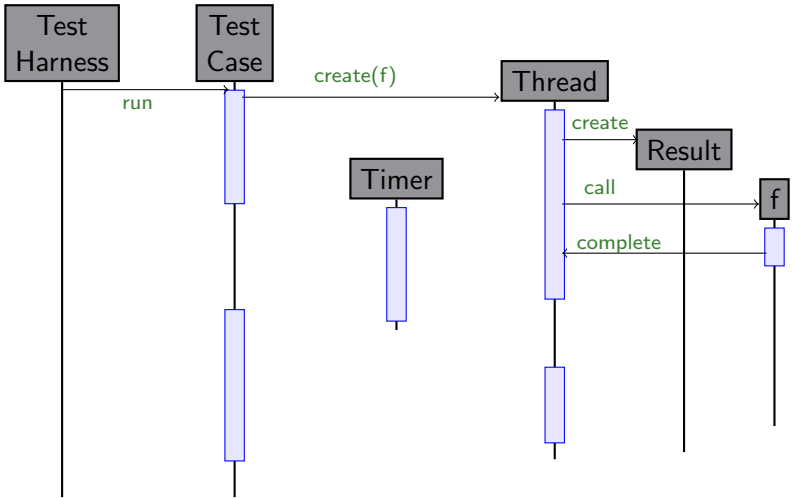
Design

Dealing with Infinite Loops – Succeeding Tests



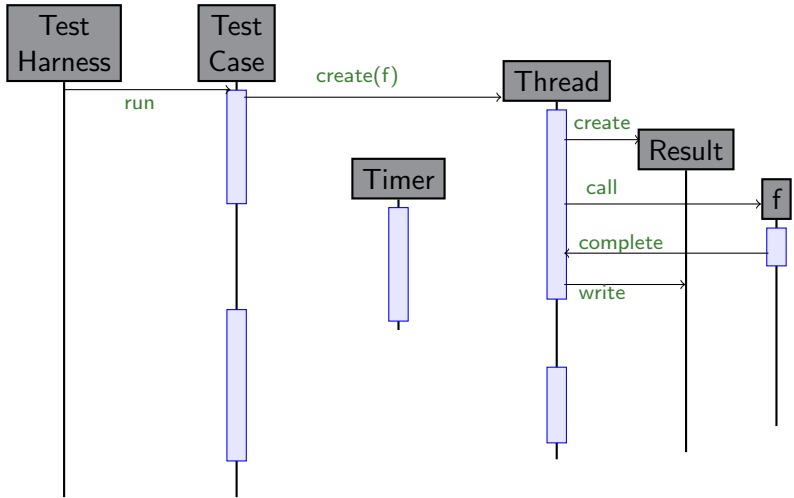
Design

Dealing with Infinite Loops – Succeeding Tests



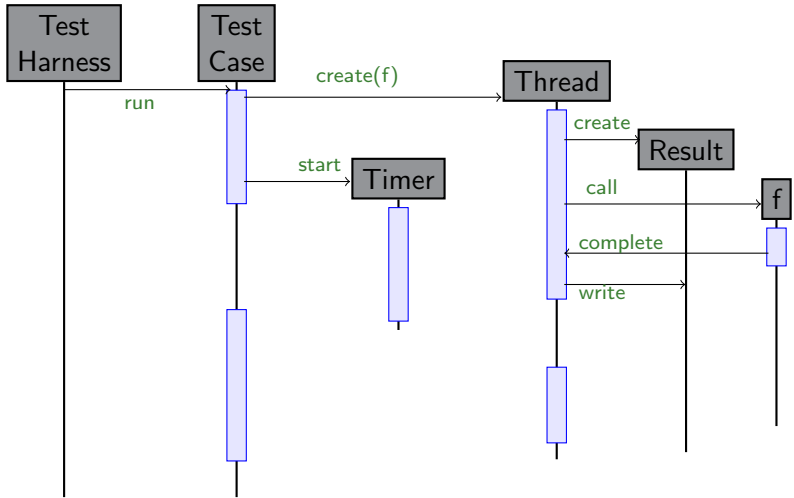
Design

Dealing with Infinite Loops – Succeeding Tests



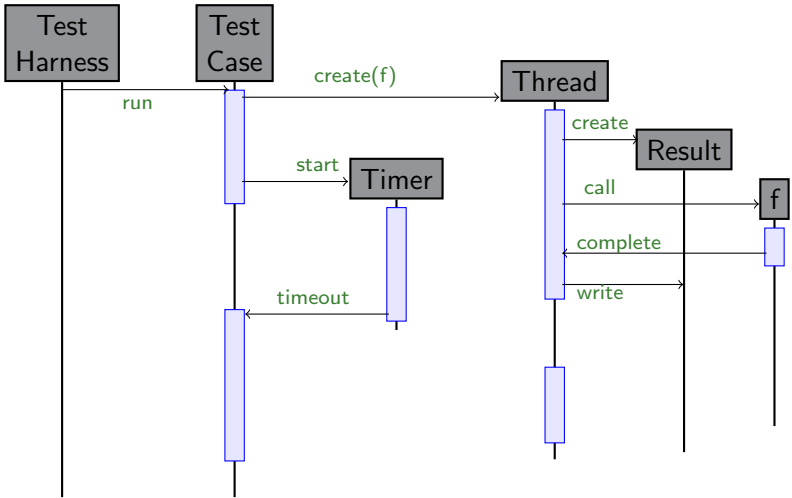
Design

Dealing with Infinite Loops – Succeeding Tests



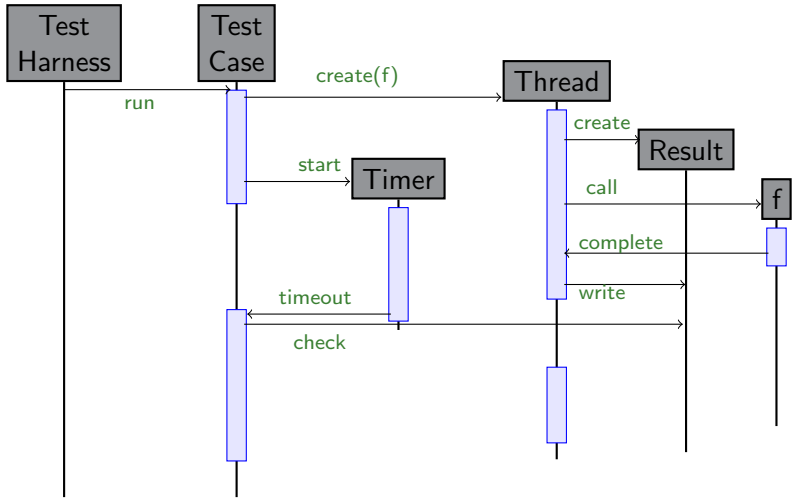
Design

Dealing with Infinite Loops – Succeeding Tests



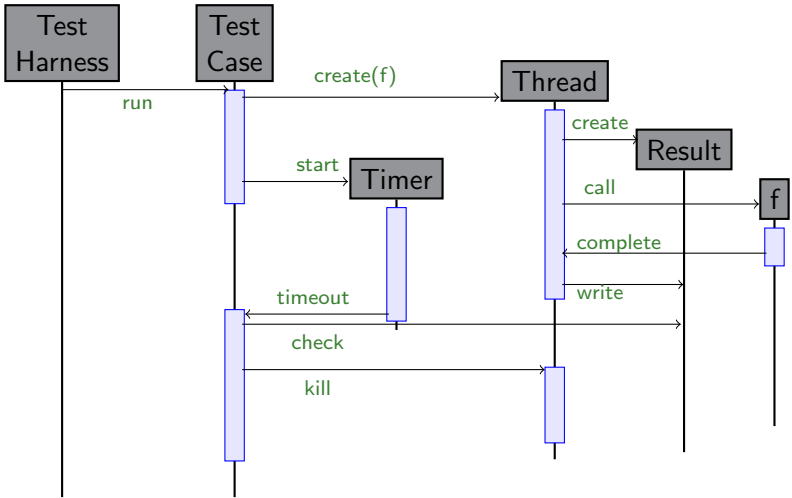
Design

Dealing with Infinite Loops – Succeeding Tests



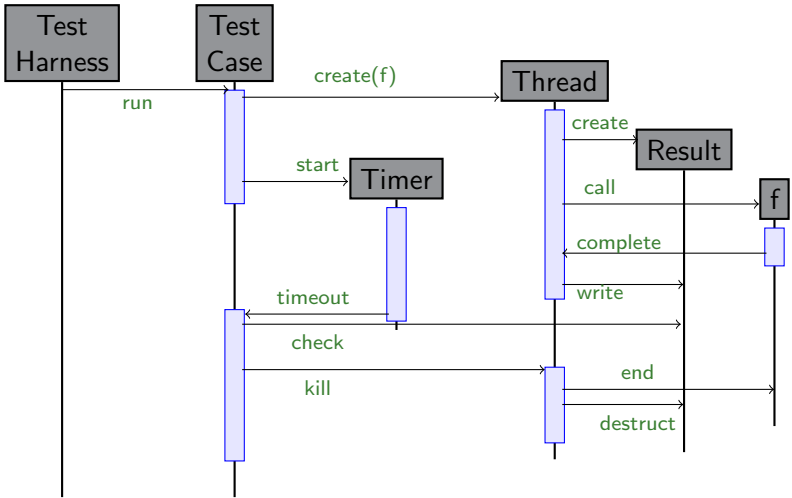
Design

Dealing with Infinite Loops – Succeeding Tests



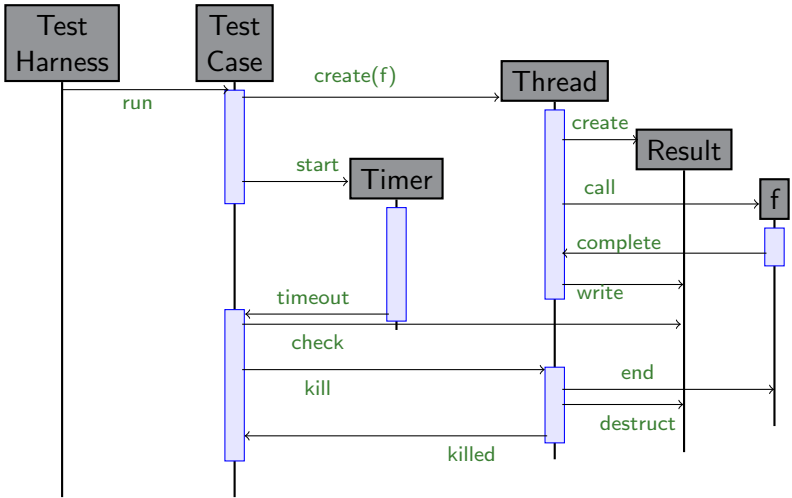
Design

Dealing with Infinite Loops – Succeeding Tests



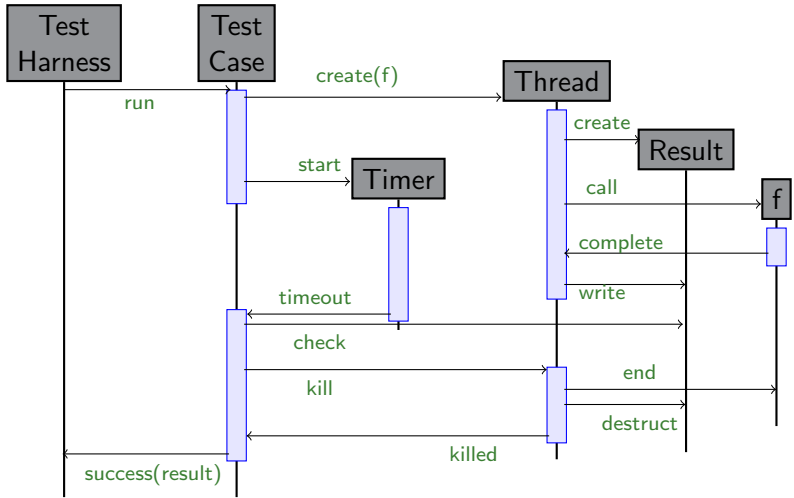
Design

Dealing with Infinite Loops – Succeeding Tests



Design

Dealing with Infinite Loops – Succeeding Tests



Experience

Advantages

- Simple setup
- Simple use
- Language independence. *The system has already been used by us in two flavours: Python and OCaml.*
- Data availability. *Extensively used by us for our other related research work.*
- Transparency
- Crowdsourced debugging
- Teaching by example

Demo

Goal

- Showcase to intructors of programming (intensive) courses
- Seek feedback – features, usability

Demo

Goal

- Test case
- Test suite
- Running an individual test suite: `eval_mathop.py`
- Running all the test suites: `eval_all.py`
- Evaluating all the submissions: `eval_all_rollnums.sh`
- `evaluate.py` library
- Sharing feedback with the class: `pack.sh` and `send-reports.sh`