Demystifying the Role of Feedback in GPT Self-Repair for Code Generation

Microsoft Research

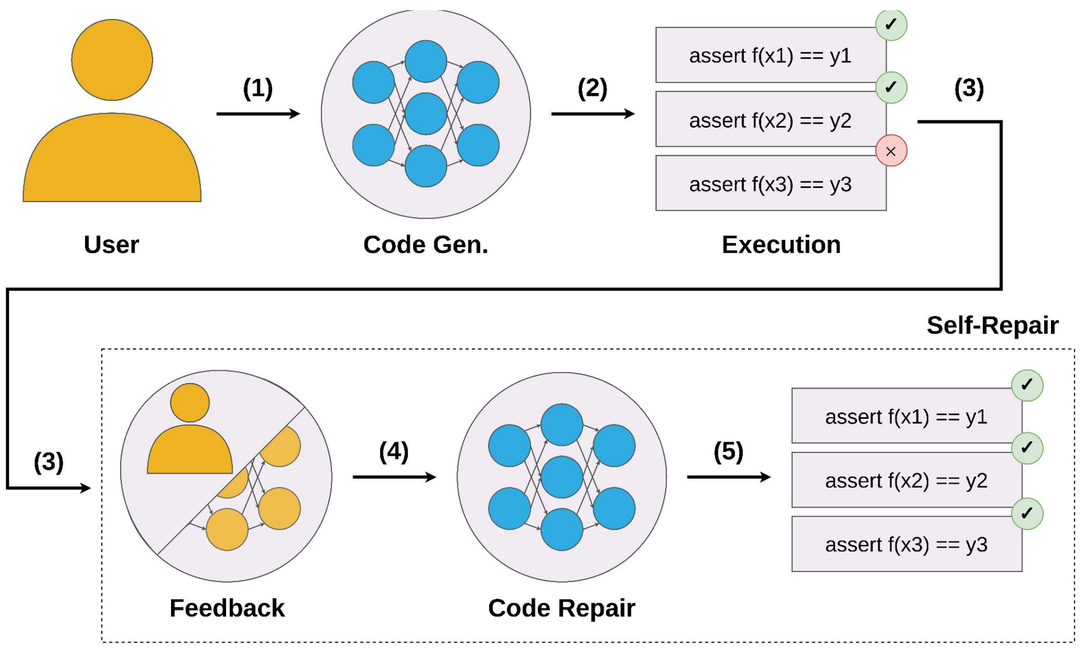
Jeevana Priya Inala
Chenglong Wang
Jianfeng Gao

MIT CSAIL

Theo X. Olausson
Armando Solar-Lezama

arXiv

1. Code Generation With & Without Self-Repair



Given is a string s representing the day of the week today. s is one of SUN, MON, TUE, WED, THU, FRI, or SAT. After how many days is the next Sunday (tomorrow or later)?

```
# UNIT TESTS
# (EXECUTABLE)
assert f('MON') == 6
assert f('WED') == 4
assert f('SUN') == 7
```

(1)

```
def f(s):
  return (7 - ['SUN',  ... , 'FRI', 'SAT'].index(s)) % 7
```

(2)

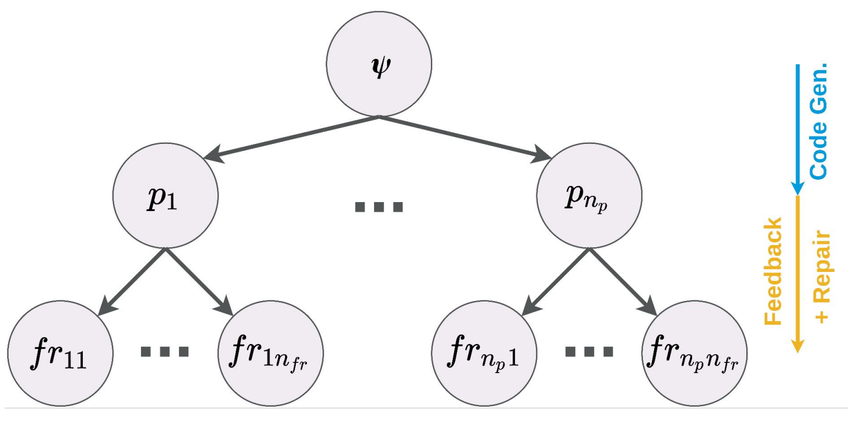
Given input 'SUN', the program returned 0, but the expected output was 7. (3)

The code does not account for the case where the input is 'SUN' and the output should be 7. This can be fixed by removing the modulo operation. (4)

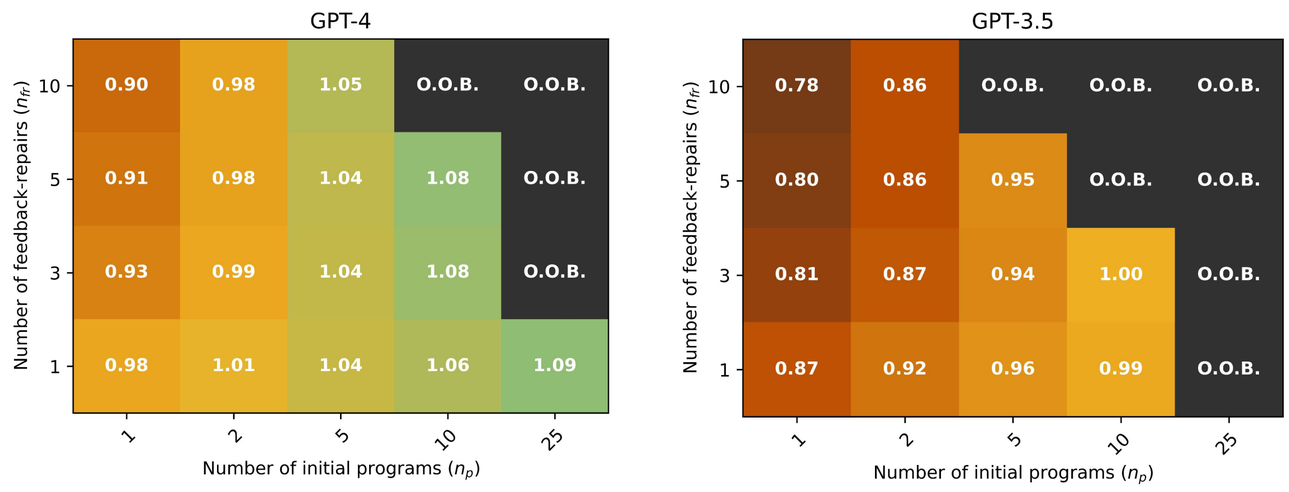
```
def f(s):
  return (7 - ['SUN',  ... , 'FRI', 'SAT'].index(s)) # % 7
```

(5)

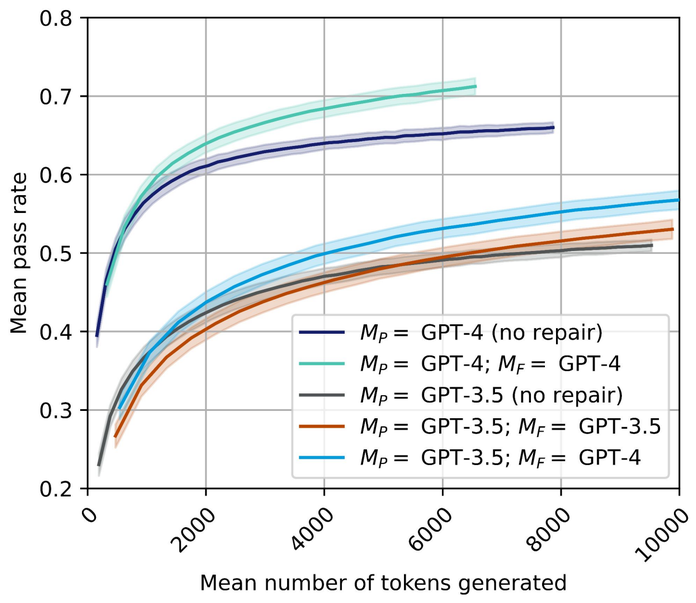
2. pass@t: Pass Rate vs. Token Count



3. Result A: Modest gains only for GPT-4



4. Result B: GPT-4 feedback enables GPT-3.5 repair



5. Result C: Human-in-the-loop significantly improves GPT-4 repair

- Small human experiment:
  - 20 tasks
  - 2 programs/task
  - 2 participants/program
  - 16 unique participants

Difficulty	GPT-4 Feedback	Human Feedback
Introductory	42.64%	62.21%
Interview	19.33%	45.67%
Competition	3.67%	14.67%
Overall	33.30%	52.60%

- Qualitatively, human feedback:
  - is much less often “obviously” wrong (7/80 vs. 32/80)
  - focuses less on explicit, small changes to code (42/80 vs. 54/80)
  - sometimes expresses uncertainty (7/80 vs 0/80 for GPT-4)

6. Summary of findings

- GPT-3.5 does not benefit from self-repair (on APPS)
- GPT-4 gains are:
  - modest (66% → 71% pass rate with a budget of 7000 tokens)
  - sensitive to the system design (more samples up front is better)
- Better feedback → better results:
  - GPT-4 feedback enables GPT-3.5 repair (50% → 54% @ 7000 tokens)
  - Human feedback boosts GPT-4 repair success rate (33.3% → 52.6%)

Conclusion: **better code explanation tools + self-repair = a recipe for success?**

arXiv