

Experiment ‘p4_8_seperate_lines’ Results

December 18, 2025

Experiment outcome: SUCCESS

Bad responses: 0

Responses containing assume : 0

Resolution attempts: 4

Hard fails (resolution): 1

Soft fails (resolution): 0

Verification attempts: 3

Problem Specification

Problem name: p4_8_seperate_lines

Natural language statement: Write a method that reads a word and returns each character of the word on a separate line.

Method signature: p4_8_seperate_lines(s: string) returns (chars: string)

Ensures

- $|\text{chars}| == 2 * |s|$
- $\forall i :: 0 \leq i < |s| \implies i \% 2 == 1 \implies \text{chars}[i] == '\n'$
- $\forall i :: 0 \leq i < |s| \implies i \% 2 == 0 \implies \text{chars}[i] == s[i / 2]$

GenAI interactions

Below you will find all interactions between the ‘user’ (program) and the ‘assistant’ (OpenAI).

Program → GenAI

You are given the following task to perform in Dafny:

Write a method that reads a word and returns each character of the word on a separate
↳ line.

The signature should be:

```
method p4_8_seperate_lines(s: string) returns (chars: string)
```

The method should respect the following contract:

```
ensures |chars| == 2 * |s|, ensures forall i :: 0 <= i < |s| ==> i % 2 == 1 ==> chars[i]
    ↳ == '\n', ensures forall i :: 0 <= i < |s| ==> i % 2 == 0 ==> chars[i] == s[i / 2]
```

Produce and show only the Dafny body of this method, including the curly braces that
↳ surround it. Do not show the signature nor contract. You must not use ‘assume’
↳ anywhere in your code.

GenAI → Program

System fingerprint:

ID: chatcmpl-Co4fAjq8EWSCRfVA8a4wA8uVSamrG

Model: gpt-5.1-chat-latest

Created at: 1766050360

Finish reason: stop

Usage: 386 tokens in, and 136 tokens out

```
{
    var res := "";
    var i := 0;
    while i < |s|
        invariant 0 <= i <= |s|
        invariant |res| == 2 * i
        invariant forall k :: 0 <= k < i ==> res[2*k] == s[k] && res[2*k+1] == '\n'
    {
        res := res + s[i] + "\n";
        i := i + 1;
    }
    chars := res;
}
```

Program → GenAI

When using dafny resolve, the below error is emitted and resolve fails:

```

problem_data/p4_8_seperate_lines/program.dfy(13,18): Error: sequence has type string
  ↪ which is incompatible with expected type seq<char> (element type char is
  ↪ incompatible with char)
|
13 | res := res + s[i] + "\n";
| ^

1 resolution/type errors detected in program.dfy

Correct the error by altering only the method body. Produce and show only the Dafny body,
  ↪ including the curly braces that surround it. Do not show the signature nor
  ↪ contract. You must not use 'assume' anywhere in your code.

```

GenAI → Program

System fingerprint:

ID: chatcmpl-Co4fDqDiguA7TsUT8t1A9wP6puDZp

Model: gpt-5.1-chat-latest

Created at: 1766050363

Finish reason: stop

Usage: 618 tokens in, and 204 tokens out

```
{
    var res := "";
    var i := 0;
    while i < |s|
        invariant 0 <= i <= |s|
        invariant |res| == 2 * i
        invariant forall k :: 0 <= k < i ==> res[2*k] == s[k] && res[2*k+1] == '\n'
    {
        res := res + s[i..i+1] + "\n";
        i := i + 1;
    }
    chars := res;
}
```

Program → GenAI

When using dafny verify, the below error is emitted and verify fails:

```

problem_data/p4_8_seperate_lines/program.dfy(5,0): Error: a postcondition could not be
  ↪ proved on this return path
|
5 | {
| ^

problem_data/p4_8_seperate_lines/program.dfy(3,9): Related location: this is the
  ↪ postcondition that could not be proved
|
3 | ensures forall i :: 0 <= i < |s| ==> i % 2 == 1 ==> chars[i] == '\n'
| ~~~~~

```

```
Dafny program verifier finished with 1 verified, 1 error

Correct the error by altering only the method body. Produce and show only the Dafny body,
    ↪ including the curly braces that surround it. Do not show the signature nor
    ↪ contract. You must not use 'assume' anywhere in your code.
```

GenAI → Program

System fingerprint:

ID: chatmpl-Co4fHVTYHsVL5bmHR0pARc94K8DmV

Model: gpt-5.1-chat-latest

Created at: 1766050367

Finish reason: stop

Usage: 1006 tokens in, and 695 tokens out

```
{
    var res := "";
    var i := 0;
    while i < |s|
        invariant 0 <= i <= |s|
        invariant |res| == 2 * i
        invariant forall k :: 0 <= k < i ==> res[2*k] == s[k] && res[2*k+1] == '\n'
    {
        res := res + s[i..i+1] + "\n";
        i := i + 1;
    }

    // Help Dafny prove the postcondition
    assert forall j :: 0 <= j < |s| && j % 2 == 1 ==> res[j] == '\n';

    chars := res;
}
```

Program → GenAI

When using dafny verify, the below error is emitted and verify fails:

```
problem_data/p4_8_seperate_lines/program.dfy(18,2): Error: assertion might not hold
  |
18 | assert forall j :: 0 <= j < |s| && j % 2 == 1 ==> res[j] == '\n';
  | ^~~~~~
```

```
Dafny program verifier finished with 1 verified, 1 error
```

```
Correct the error by altering only the method body. Produce and show only the Dafny body,
    ↪ including the curly braces that surround it. Do not show the signature nor
    ↪ contract. You must not use 'assume' anywhere in your code.
```

GenAI → Program

System fingerprint:

ID: chatcmpl-Co4fRppRiHUhu8CO0kfPMAJOJu08

Model: gpt-5.1-chat-latest

Created at: 1766050377

Finish reason: stop

Usage: 1283 tokens in, and 414 tokens out

```
{  
    var res := "";  
    var i := 0;  
    while i < |s|  
        invariant 0 <= i <= |s|  
        invariant |res| == 2 * i  
        invariant forall j :: 0 <= j < 2*i ==> j % 2 == 1 ==> res[j] == '\n'  
        invariant forall k :: 0 <= k < i ==> res[2*k] == s[k]  
    {  
        res := res + s[i..i+1] + "\n";  
        i := i + 1;  
    }  
    chars := res;  
}
```

Final Program

```
method p4_8_seperate_lines(s: string) returns (chars: string)  
    ensures |chars| == 2 * |s|  
    ensures forall i :: 0 <= i < |s| ==> i % 2 == 1 ==> chars[i] == '\n'  
    ensures forall i :: 0 <= i < |s| ==> i % 2 == 0 ==> chars[i] == s[i / 2]  
{  
    var res := "";  
    var i := 0;  
    while i < |s|  
        invariant 0 <= i <= |s|  
        invariant |res| == 2 * i  
        invariant forall j :: 0 <= j < 2*i ==> j % 2 == 1 ==> res[j] == '\n'  
        invariant forall k :: 0 <= k < i ==> res[2*k] == s[k]  
    {  
        res := res + s[i..i+1] + "\n";  
        i := i + 1;  
    }  
    chars := res;  
}
```

Total Token Usage

Input tokens: 3293

Output tokens: 1449

Reasoning tokens: 832
Sum of ‘total tokens’: 4742

Experiment Timings

Overall Experiment started at 1766050362346, ended at 1766050385006, lasting 22660ms (22.66 seconds)

Iteration #4 started at 1766050379169, ended at 1766050385006, lasting 5837ms (5.84 seconds)

Iteration #1 started at 1766050362346, ended at 1766050365266, lasting 2920ms (2.92 seconds)

Iteration #2 started at 1766050365266, ended at 1766050369400, lasting 4134ms (4.13 seconds)

Iteration #3 started at 1766050369400, ended at 1766050379169, lasting 9769ms (9.77 seconds)