

Experiment 'p6_6_alternating_sum' Results

December 18, 2025

Experiment outcome: SUCCESS

Bad responses: 0

Responses containing assume : 0

Resolution attempts: 6

Hard fails (resolution): 0

Soft fails (resolution): 0

Verification attempts: 6

Problem Specification

Problem name: p6_6_alternating_sum

Natural language statement: Compute the alternating sum of all elements in an array. For example, if your method reads the input [1, 4, 9, 16, 9, 7, 4, 9, 11] then it computes $1 - 4 + 9 - 16 + 9 - 7 + 4 - 9 + 11 = -2$.

Method signature: p6_6_alternating_sum(arr: seq<int>) returns (alt_sum: int)

Ensures

- `alt_sum == sum_array(arr, 1)`

Functional Code Given

```
function sum_array(arr : seq<int>, i : int) : int
{
  if |arr| == 0 then 0
  else arr[0] * i + sum_array(arr[1..], i * -1)
}
```

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:

Compute the alternating sum of all elements in an array. For example, if your method
→ reads the input [1, 4, 9, 16, 9, 7, 4, 9, 11] then it computes $1 - 4 + 9 - 16 + 9 - 7 + 4 - 9 + 11 = -2$.

The signature should be:

```
method p6_6_alternating_sum(arr: seq<int>) returns (alt_sum: int)
```

The method should respect the following contract:

```
ensures alt_sum == sum_array(arr, 1)
```

The contract uses the following dafny code:

```
function sum_array(arr : seq<int>, i : int) : int
{
  if |arr| == 0 then 0
  else arr[0] * i + sum_array(arr[1..], i * -1)
}
```

These function(s) must not be used in your implementation of the method.

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-Co5AyB7gTHX0mcbzmtQemu1IDWvXP

Model: gpt-5.1-chat-latest

Created at: 1766052332

Finish reason: stop

Usage: 548 tokens in, and 341 tokens out

```
{
  var i := 0;
  var sign := 1;
  alt_sum := 0;
  while i < |arr|
    invariant 0 <= i <= |arr|
    invariant alt_sum == sum_array(arr[..i], 1)
    invariant (sign == 1 && i % 2 == 0) || (sign == -1 && i % 2 == 1)
  {
```

```

    alt_sum := alt_sum + arr[i] * sign;
    sign := sign * -1;
    i := i + 1;
  }
}

```

Program → GenAI

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

```

problem_data/p6_6_alternating_sum/program.dfy(9,0): Error: a postcondition could not be
  ↳ proved on this return path

```

```

|
9 | {
| ~

```

```

problem_data/p6_6_alternating_sum/program.dfy(8,17): Related location: this is the
  ↳ postcondition that could not be proved

```

```

|
8 | ensures alt_sum == sum_array(arr, 1)
| ^^

```

```

problem_data/p6_6_alternating_sum/program.dfy(15,22): Error: this invariant could not be
  ↳ proved to be maintained by the loop

```

```

Related message: loop invariant violation
|
15 | invariant alt_sum == sum_array(arr[..i], 1)
| ^^

```

Dafny program verifier finished with 1 verified, 2 errors

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-Co5B6eiIItQittZEyJ8Vi4KFrqic

Model: gpt-5.1-chat-latest

Created at: 1766052340

Finish reason: stop

Usage: 918 tokens in, and 120 tokens out

```

{
  alt_sum := 0;
  var i := 0;
  while i < |arr|
    invariant 0 <= i <= |arr|
    invariant alt_sum == sum_array(arr[..i], 1)

```

```

{
  var sign := if i % 2 == 0 then 1 else -1;
  alt_sum := alt_sum + arr[i] * sign;
  i := i + 1;
}
}

```

Program → GenAI

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

```

problem_data/p6_6_alternating_sum/program.dfy(9,0): Error: a postcondition could not be
  ↳ proved on this return path

```

```

|
9 | {
|  ^

```

```

problem_data/p6_6_alternating_sum/program.dfy(8,17): Related location: this is the
  ↳ postcondition that could not be proved

```

```

|
8 | ensures alt_sum == sum_array(arr, 1)
|  ^^

```

```

problem_data/p6_6_alternating_sum/program.dfy(14,22): Error: this invariant could not be
  ↳ proved to be maintained by the loop

```

```

Related message: loop invariant violation
|
14 | invariant alt_sum == sum_array(arr[..i], 1)
|  ^^

```

Dafny program verifier finished with 1 verified, 2 errors

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-Co5B9lmXq4cH88Dn4cVVLbBVRlwN5

Model: gpt-5.1-chat-latest

Created at: 1766052343

Finish reason: stop

Usage: 1277 tokens in, and 458 tokens out

```

{
  alt_sum := 0;
  var i := 0;
  var sign := 1;
  while i < |arr|

```

```

    invariant 0 <= i <= |arr|
    invariant alt_sum == sum_array(arr[..i], 1)
    invariant sign == (if i % 2 == 0 then 1 else -1)
  {
    alt_sum := alt_sum + arr[i] * sign;
    sign := sign * -1;
    i := i + 1;
  }
}

```

Program → GenAI

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

```

problem_data/p6_6_alternating_sum/program.dfy(9,0): Error: a postcondition could not be
    ↪ proved on this return path
|
9 | {
| ^

problem_data/p6_6_alternating_sum/program.dfy(8,17): Related location: this is the
    ↪ postcondition that could not be proved
|
8 | ensures alt_sum == sum_array(arr, 1)
| ^^

problem_data/p6_6_alternating_sum/program.dfy(15,22): Error: this invariant could not be
    ↪ proved to be maintained by the loop
Related message: loop invariant violation
|
15 | invariant alt_sum == sum_array(arr[..i], 1)
| ^^

Dafny program verifier finished with 1 verified, 2 errors

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-Co5BGlAiEqTupRNyctMw16g1khhhY

Model: gpt-5.1-chat-latest

Created at: 1766052350

Finish reason: stop

Usage: 1654 tokens in, and 248 tokens out

```

{
  alt_sum := 0;

```

```

var i := 0;
while i < |arr|
  invariant 0 <= i <= |arr|
  invariant alt_sum == sum_array(arr[..i], 1)
{
  var sign := if i % 2 == 0 then 1 else -1;
  alt_sum := alt_sum + arr[i] * sign;
  i := i + 1;
}
}

```

Program → GenAI

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

```

problem_data/p6_6_alternating_sum/program.dfy(9,0): Error: a postcondition could not be
  ↳ proved on this return path

```

```

|
9 | {
| ^

```

```

problem_data/p6_6_alternating_sum/program.dfy(8,17): Related location: this is the
  ↳ postcondition that could not be proved

```

```

|
8 | ensures alt_sum == sum_array(arr, 1)
| ^^

```

```

problem_data/p6_6_alternating_sum/program.dfy(14,22): Error: this invariant could not be
  ↳ proved to be maintained by the loop
Related message: loop invariant violation

```

```

|
14 | invariant alt_sum == sum_array(arr[..i], 1)
| ^^

```

Dafny program verifier finished with 1 verified, 2 errors

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-Co5BLAKzjpu0uZdqE8P0DM1GWIWLN

Model: gpt-5.1-chat-latest

Created at: 1766052355

Finish reason: stop

Usage: 2013 tokens in, and 200 tokens out

```

{

```

```

alt_sum := 0;
var i := 0;
var sign := 1;
while i < |arr|
  invariant 0 <= i <= |arr|
  invariant alt_sum == sum_array(arr[..i], 1)
  invariant sign == (if i % 2 == 0 then 1 else -1)
  {
    alt_sum := alt_sum + arr[i] * sign;
    i := i + 1;
    sign := -sign;
  }
}

```

Program → GenAI

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

```

problem_data/p6_6_alternating_sum/program.dfy(9,0): Error: a postcondition could not be
  ↳ proved on this return path
|
9 | {
| ~

problem_data/p6_6_alternating_sum/program.dfy(8,17): Related location: this is the
  ↳ postcondition that could not be proved
|
8 | ensures alt_sum == sum_array(arr, 1)
| ^^

problem_data/p6_6_alternating_sum/program.dfy(15,22): Error: this invariant could not be
  ↳ proved to be maintained by the loop
Related message: loop invariant violation
|
15 | invariant alt_sum == sum_array(arr[..i], 1)
| ^^

Dafny program verifier finished with 1 verified, 2 errors

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-Co5BQNpgiQsg8qD9ADeYBP7POpoOZ

Model: gpt-5.1-chat-latest

Created at: 1766052360

Finish reason: stop

Usage: 2388 tokens in, and 525 tokens out

```
{
  alt_sum := 0;
  var i := 0;
  while i < |arr|
    invariant 0 <= i <= |arr|
    invariant alt_sum + sum_array(arr[i..], if i % 2 == 0 then 1 else -1) == sum_array(
      ↪ arr, 1)
    {
      var sign := if i % 2 == 0 then 1 else -1;
      alt_sum := alt_sum + arr[i] * sign;
      i := i + 1;
    }
  }
}
```

Final Program

```
function sum_array(arr : seq<int>, i : int) : int
{
  if |arr| == 0 then 0
  else arr[0] * i + sum_array(arr[1..], i * -1)
}

method p6_6_alternating_sum(arr: seq<int>) returns (alt_sum: int)
  ensures alt_sum == sum_array(arr, 1)
{
  alt_sum := 0;
  var i := 0;
  while i < |arr|
    invariant 0 <= i <= |arr|
    invariant alt_sum + sum_array(arr[i..], if i % 2 == 0 then 1 else -1) == sum_array(
      ↪ arr, 1)
    {
      var sign := if i % 2 == 0 then 1 else -1;
      alt_sum := alt_sum + arr[i] * sign;
      i := i + 1;
    }
  }
}
```

Total Token Usage

Input tokens: 8798

Output tokens: 1892

Reasoning tokens: 1088

Sum of 'total tokens': 10690

Experiment Timings

Overall Experiment started at 1766052334436, ended at 1766052371157, lasting 36721ms (36.72 seconds)

Iteration #4 started at 1766052352427, ended at 1766052357458, lasting 5031ms (5.03 seconds)

Iteration #5 started at 1766052357458, ended at 1766052361800, lasting 4342ms (4.34 seconds)

Iteration #6 started at 1766052361800, ended at 1766052371157, lasting 9357ms (9.36 seconds)

Iteration #1 started at 1766052334437, ended at 1766052342244, lasting 7807ms (7.81 seconds)

Iteration #2 started at 1766052342244, ended at 1766052345558, lasting 3314ms (3.31 seconds)

Iteration #3 started at 1766052345558, ended at 1766052352427, lasting 6869ms (6.87 seconds)