Experiment 'p4_6_find_minimum' Results

December 18, 2025

Experiment outcome: SUCCESS
Bad responses: 0
Responses containing `assume :` 0
Resolution attempts: 3
Hard fails (resolution): 0
Soft fails (resolution): 1
Verification attempts: 2

Problem Specification

Problem name: p4_6_find_minimum
Natural language statement: Write a method that finds the minimum value from a list of inputs into a method.
Method signature: p4_6_find_minimum(inputs: array<int>) returns (min: int)

Ensures

- `forall i :: 0 <= i < inputs.Length ==> min <= inputs[i]`
- `exists i :: 0 <= i < inputs.Length && min == inputs[i]`

Requires

- `0 < inputs.Length`

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 finds the minimum value from a list of inputs into a method.

The signature should be:

method p4_6_find_minimum(inputs: array<int>) returns (min: int)

The method should respect the following contract:

requires 0 < inputs.Length, ensures forall i :: 0 <= i < inputs.Length ==> min <= inputs[
    ↪ i], ensures exists i :: 0 <= i < inputs.Length && min == inputs[i]

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-Co4ejFoVBGtDTV1FajCrsDenhnaSW
Model: gpt-5.1-chat-latest
Created at: 1766050333
Finish reason: stop
Usage: 322 tokens in, and 197 tokens out

```
{
  var i := 0;
  min := inputs[0];
  // Loop to scan all elements
  while i < inputs.Length
    invariant 0 <= i <= inputs.Length
    invariant forall j :: 0 <= j < i ==> min <= inputs[j]
    invariant exists j :: 0 <= j < i ==> min == inputs[j]
  {
    if inputs[i] < min {
      min := inputs[i];
    }
    i := i + 1;
  }
}
```

Program → GenAI

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

```
problem_data/p4_6_find_minimum/program.dfy(12,14): Warning: the quantifier has the form '
    ↪ exists x :: A ==> B', which most often is a typo for 'exists x :: A && B'; if you
    ↪ think otherwise, rewrite as 'exists x :: (A ==> B)' or 'exists x :: !A || B' to
    ↪ suppress this warning
   |
12 | invariant exists j :: 0 <= j < i ==> min == inputs[j]
   | ^^^^^^

Compilation failed because warnings were found and --allow-warnings is false

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-Co4en7gJ7UOQjNdudt5JRfJ2tmQ9u
Model: gpt-5.1-chat-latest
Created at: 1766050337
Finish reason: stop
Usage: 679 tokens in, and 132 tokens out

```
{
  var i := 0;
  min := inputs[0];
  while i < inputs.Length
    invariant 0 <= i <= inputs.Length
    invariant forall j :: 0 <= j < i ==> min <= inputs[j]
    invariant i > 0 ==> (exists j :: 0 <= j < i && min == inputs[j])
  {
    if inputs[i] < min {
      min := inputs[i];
    }
    i := i + 1;
  }
}
```

Program → GenAI

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

problem_data/p4_6_find_minimum/program.dfy(11,25): Error: this invariant could not be
    ↪ proved to be maintained by the loop
 Related message: loop invariant violation
   |
11 | invariant i > 0 ==> (exists j :: 0 <= j < i && min == inputs[j])
   | ^^^^^^
```

```
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-Co4erR7qkPrJHIRJYbrmU945N3uyb
Model: gpt-5.1-chat-latest
Created at: 1766050341
Finish reason: stop
Usage: 931 tokens in, and 206 tokens out

```
{
  var minIndex := 0;
  min := inputs[0];
  var i := 1;
  while i < inputs.Length
    invariant 1 <= i <= inputs.Length
    invariant 0 <= minIndex < i
    invariant min == inputs[minIndex]
    invariant forall j :: 0 <= j < i ==> min <= inputs[j]
  {
    if inputs[i] < min {
      min := inputs[i];
      minIndex := i;
    }
    i := i + 1;
  }
}
```

Final Program

```
method p4_6_find_minimum(inputs: array<int>) returns (min: int)
       requires 0 < inputs.Length
       ensures forall i :: 0 <= i < inputs.Length ==> min <= inputs[i]
       ensures exists i :: 0 <= i < inputs.Length && min == inputs[i]
{
  var minIndex := 0;
  min := inputs[0];
  var i := 1;
  while i < inputs.Length
    invariant 1 <= i <= inputs.Length
    invariant 0 <= minIndex < i
    invariant min == inputs[minIndex]
    invariant forall j :: 0 <= j < i ==> min <= inputs[j]
  {
    if inputs[i] < min {
```

```
    min := inputs[i];
    minIndex := i;
   }
   i := i + 1;
  }
}
```

Total Token Usage

Input tokens: 1932
Output tokens: 535
Reasoning tokens: 128
Sum of 'total tokens': 2467

Experiment Timings

Overall Experiment started at 1766050335307, ended at 1766050347630, lasting 12323ms (12.32 seconds)
Iteration #1 started at 1766050335309, ended at 1766050339083, lasting 3774ms (3.77 seconds)
Iteration #2 started at 1766050339083, ended at 1766050342814, lasting 3731ms (3.73 seconds)
Iteration #3 started at 1766050342814, ended at 1766050347630, lasting 4816ms (4.82 seconds)