Experiment ‘p4_1_c_powers_of_two’ Results

December 18, 2025

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

Problem Specification

Problem name: p4_1_c_powers_of_two
Natural language statement: Write a method with loops that computes all powers of 2 from 2^0 up to 2^20.
Method signature: p4_1_c_powers_of_two() returns (powers: seq<int>)

Ensures

- `|powers| == 21`
- `forall i :: 0 <= i < |powers| ==> powers[i] == pow(2, i)`

Functional Code Given

```
function pow(base: int, exp: int): int
  requires 0 <= exp
  decreases exp
{
  if exp == 0 then 1 else base * pow(base, exp - 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:

Write a method with loops that computes all powers of 2 from 2^0 up to 2^20.

The signature should be:

method p4_1_c_powers_of_two() returns (powers: seq<int>)

The method should respect the following contract:

ensures |powers| == 21, ensures forall i :: 0 <= i < |powers| ==> powers[i] == pow(2, i)

The contract uses the following dafny code:

function pow(base: int, exp: int): int
  requires 0 <= exp
  decreases exp
{
  if exp == 0 then 1 else base * pow(base, exp - 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-Co4ZTgqw85F6Ctgz5Z0clsiQ5mgFm
Model: gpt-5.1-chat-latest
Created at: 1766050007
Finish reason: stop
Usage: 458 tokens in, and 202 tokens out

```
{
  var res: seq<int> := [];
  var cur: int := 1;
  var i: int := 0;
  while i <= 20
    invariant 0 <= i <= 21
    invariant |res| == i
    invariant forall k :: 0 <= k < i ==> res[k] == (1 << k)
  {
    res := res + [cur];
    cur := cur * 2;
    i := i + 1;
  }
  powers := res;
}
```

Program → GenAI

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

problem_data/p4_1_c_powers_of_two/program.dfy(18,54): Error: type of << must be a
    ↪ bitvector type (instead got int)
   |
18 | invariant forall k :: 0 <= k < i ==> res[k] == (1 << k)
   | ^^

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-Co4ZXNJ2y2FVDnuhi9WA1M4GFkEr9
Model: gpt-5.1-chat-latest
Created at: 1766050011
Finish reason: stop
Usage: 660 tokens in, and 314 tokens out

```
{
  var res: seq<int> := [];
  var cur: int := 1;
  var i: int := 0;
  while i <= 20
    invariant 0 <= i <= 21
    invariant |res| == i
    invariant forall k :: 0 <= k < i ==> (k == 0 ==> res[k] == 1) && (k > 0 ==> res[k] ==
        ↪ 2 * res[k-1])
    invariant cur == (if i == 0 then 1 else 2 * res[i-1])
  {
    res := res + [cur];
    cur := cur * 2;
    i := i + 1;
  }
  powers := res;
}
```

Program → GenAI

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

problem_data/p4_1_c_powers_of_two/program.dfy(11,0): Error: a postcondition could not be
    ↪ proved on this return path
   |
11 | {
   | ^

problem_data/p4_1_c_powers_of_two/program.dfy(10,9): Related location: this is the
    ↪ postcondition that could not be proved
   |
10 |  ensures forall i :: 0 <= i < |powers| ==> powers[i] == pow(2, i)
   | ^^^^^^


Dafny program verifier finished with 2 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-Co4ZdK6t6hmf23msuhiwXa0DP4CJs
Model: gpt-5.1-chat-latest
Created at: 1766050017
Finish reason: stop
Usage: 1090 tokens in, and 150 tokens out

```
{
  var res: seq<int> := [];
  var cur: int := 1;
  var i: int := 0;
  while i <= 20
    invariant 0 <= i <= 21
    invariant |res| == i
    invariant cur == pow(2, i)
    invariant forall k :: 0 <= k < i ==> res[k] == pow(2, k)
  {
    res := res + [cur];
    cur := cur * 2;
    i := i + 1;
  }
  powers := res;
}
```

Final Program

```
function pow(base: int, exp: int): int
  requires 0 <= exp
  decreases exp
{
  if exp == 0 then 1 else base * pow(base, exp - 1)
}

method p4_1_c_powers_of_two() returns (powers: seq<int>)
       ensures |powers| == 21
       ensures forall i :: 0 <= i < |powers| ==> powers[i] == pow(2, i)
{
  var res: seq<int> := [];
  var cur: int := 1;
  var i: int := 0;
  while i <= 20
    invariant 0 <= i <= 21
    invariant |res| == i
    invariant cur == pow(2, i)
    invariant forall k :: 0 <= k < i ==> res[k] == pow(2, k)
  {
    res := res + [cur];
    cur := cur * 2;
    i := i + 1;
  }
  powers := res;
}
```

Total Token Usage

Input tokens: 2208
Output tokens: 666
Reasoning tokens: 192
Sum of 'total tokens': 2874

Experiment Timings

Overall Experiment started at 1766050009776, ended at 1766050023333, lasting 13557ms (13.56 seconds)
Iteration #1 started at 1766050009776, ended at 1766050013310, lasting 3534ms (3.53 seconds)
Iteration #2 started at 1766050013310, ended at 1766050019052, lasting 5742ms (5.74 seconds)
Iteration #3 started at 1766050019052, ended at 1766050023333, lasting 4281ms (4.28 seconds)