

Application of Mathematical Logic in Functional Program Verification

Nikolaj Popov and Tudor Jebelean

Research Institute for Symbolic Computation, Linz

{popov, jebelean}@risc.uni-linz.ac.at

Outline

Functional Program Verification
Total Correctness
Building up Correct Programs
Coherent Programs. Recursion
Soundness and Completeness

Conclusion and Discussions

Preconditions and Postconditions. Total Correctness

Given the triple

$\{I\}F\{O\}$ (Input condition, Function definition, Output condition)

Total Correctness Formula

$(\forall n : I[n]) (F[n] \downarrow \wedge O[n, F[n]])$

Example

$\{x \in \mathbb{R} \wedge n \in \mathbb{N}\}$

$pow[x, n] = \text{If } n = 0 \text{ then } 1 \text{ else } x * pow[x, n - 1]$

$\{x^n = pow[x, n]\}$

$(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (pow[x, n] \downarrow \wedge x^n = pow[x, n])$

Preconditions and Postconditions. Total Correctness

Given the triple

$\{I\}F\{O\}$ (Input condition, Function definition, Output condition)

Total Correctness Formula

$(\forall n : I[n]) (F[n] \downarrow \wedge O[n, F[n]])$

Example

$\{x \in \mathbb{R} \wedge n \in \mathbb{N}\}$

$pow[x, n] = \text{If } n = 0 \text{ then } 1 \text{ else } x * pow[x, n - 1]$

$\{x^n = pow[x, n]\}$

$(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (pow[x, n] \downarrow \wedge x^n = pow[x, n])$

Preconditions and Postconditions. Total Correctness

Given the triple

$\{I\}F\{O\}$ (Input condition, Function definition, Output condition)

Total Correctness Formula

$(\forall n : I[n]) (F[n] \downarrow \wedge O[n, F[n]])$

Example

$\{x \in \mathbb{R} \wedge n \in \mathbb{N}\}$

$pow[x, n] = \text{If } n = 0 \text{ then } 1 \text{ else } x * pow[x, n - 1]$

$\{x^n = pow[x, n]\}$

$(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (pow[x, n] \downarrow \wedge x^n = pow[x, n])$

Building up Correct Programs

Basic Functions e.g. +, -, *, etc.

New Functions in Terms of Already Known Functions

► Input and output predicates;

► Recursive definitions;

Modularity. After proving correctness, use only the specification.

$\{x \in \mathbb{R} \wedge n \in \mathbb{N}\}$ *Input condition*

$pow[x, n] = \dots$

$\{x^n = pow[x, n]\}$ *Output condition*

Building up Correct Programs

Basic Functions e.g. +, -, *, etc.

New Functions in Terms of Already Known Functions

- ▶ Input and output predicates;
- ▶ Prove total correctness;

Modularity. After proving correctness, use only the specification.

$\{x \in \mathbb{R} \wedge n \in \mathbb{N}\}$ *Input condition*

$pow[x, n] = \dots$

$\{x^n = pow[x, n]\}$ *Output condition*

Building up Correct Programs

Basic Functions e.g. +, -, *, etc.

New Functions in Terms of Already Known Functions

- ▶ Input and output predicates;
- ▶ Prove total correctness;

Modularity. After proving correctness, use only the specification.

$\{x \in \mathbb{R} \wedge n \in \mathbb{N}\}$ *Input condition*

$pow[x, n] = \dots$

$\{x^n = pow[x, n]\}$ *Output condition*

Building up Correct Programs

Basic Functions e.g. +, -, *, etc.

New Functions in Terms of Already Known Functions

- ▶ Input and output predicates;
- ▶ Prove total correctness;

Modularity. After proving correctness, use only the specification.

$\{x \in \mathbb{R} \wedge n \in \mathbb{N}\}$ *Input condition*

$pow[x, n] = \dots$

$\{x^n = pow[x, n]\}$ *Output condition*

Building up Correct Programs

Basic Functions e.g. +, -, *, etc.

New Functions in Terms of Already Known Functions

- ▶ Input and output predicates;
- ▶ Prove total correctness;

Modularity. After proving correctness, use only the specification.

$\{x \in \mathbb{R} \wedge n \in \mathbb{N}\}$ *Input condition*

$pow[x, n] = \dots$

$\{x^n = pow[x, n]\}$ *Output condition*

Building up Correct Programs

Appropriate values for the auxiliary functions

No input condition of an auxiliary function will be violated

Example

$F[x] = \text{If } Q[x] \text{ then } H[x] \text{ else } G[x]$

Condition: $Q[x] \rightarrow H[x]$

Condition: $\neg Q[x] \rightarrow G[x]$

Building up Correct Programs

Appropriate values for the auxiliary functions

No input condition of an auxiliary function will be violated

Example

$F[x] = \text{If } Q[x] \text{ then } H[x] \text{ else } G[x]$

$\vdash (\forall x : F[x]) \ (Q[x] \rightarrow H[x])$

$\vdash (\forall x : F[x]) \ (Q[x] \rightarrow G[x])$

Building up Correct Programs

Appropriate values for the auxiliary functions

No input condition of an auxiliary function will be violated

Example

$F[x] = \text{If } Q[x] \text{ then } H[x] \text{ else } G[x]$

- ▶ $(\forall x : I_F[x]) (Q[x] \Rightarrow I_H[x])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_G[x])$

Building up Correct Programs

Appropriate values for the auxiliary functions

No input condition of an auxiliary function will be violated

Example

$F[x] = \text{If } Q[x] \text{ then } H[x] \text{ else } G[x]$

- ▶ $(\forall x : I_F[x]) \ (Q[x] \Rightarrow I_H[x])$
- ▶ $(\forall x : I_F[x]) \ (\neg Q[x] \Rightarrow I_G[x])$

Building up Correct Programs

Appropriate values for the auxiliary functions

No input condition of an auxiliary function will be violated

Example

$F[x] = \text{If } Q[x] \text{ then } H[x] \text{ else } G[x]$

- ▶ $(\forall x : I_F[x]) \ (Q[x] \Rightarrow I_H[x])$
- ▶ $(\forall x : I_F[x]) \ (\neg Q[x] \Rightarrow I_G[x])$

Coherent Programs

Simple Recursive Programs

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R[x]]]$

Conditions for coherency

$\neg (\forall x \in A[x]) (Q[x] \rightarrow I[x])$

$\neg (\exists x \in A[x]) (Q[x] \wedge I[x])$

$\neg (\exists x \in A[x]) (I[x] \wedge \neg Q[x])$

$\neg (\exists x \in A[x]) (I[x] \wedge \exists y \in A[y] (y \neq x \wedge Q[y] \wedge I[y]))$

Coherent Programs

Simple Recursive Programs

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R[x]]]$

Conditions for coherency

- ▶ $(\forall x : I_F[x]) (Q[x] \Rightarrow I_S[x])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_F[R[x]])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_R[x])$
- ▶ $(\forall x, y : I_F[x]) (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow I_C[x, y])$

Coherent Programs

Simple Recursive Programs

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R[x]]]$

Conditions for coherency

- ▶ $(\forall x : I_F[x]) (Q[x] \Rightarrow I_S[x])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_F[R[x]])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_R[x])$
- ▶ $(\forall x, y : I_F[x]) (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow I_C[x, y])$

Coherent Programs

Simple Recursive Programs

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R[x]]]$

Conditions for coherency

- ▶ $(\forall x : I_F[x]) (Q[x] \Rightarrow I_S[x])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_F[R[x]])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_R[x])$
- ▶ $(\forall x, y : I_F[x]) (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow I_C[x, y])$

Coherent Programs

Simple Recursive Programs

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R[x]]]$

Conditions for coherency

- ▶ $(\forall x : I_F[x]) (Q[x] \Rightarrow I_S[x])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_F[R[x]])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_R[x])$
- ▶ $(\forall x, y : I_F[x]) (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow I_C[x, y])$

Coherent Programs

Simple Recursive Programs

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R[x]]]$

Conditions for coherency

- ▶ $(\forall x : I_F[x]) (Q[x] \Rightarrow I_S[x])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_F[R[x]])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_R[x])$
- ▶ $(\forall x, y : I_F[x]) (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow I_C[x, y])$

Verification Conditions Generation

Simple Recursive Program

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R[x]]]$

is correct if the verification conditions hold

$$\vdash (\forall x : I[x]) \ (Q[x] \rightarrow C[x, S[x]])$$

and the recursive condition holds

$\vdash (\forall x : I[x]) \ (S[x] \rightarrow F[R[x]])$

and the base case

$\vdash (\forall x : I[x]) \ (C[x, S[x]] \wedge F[R[x]])$

Verification Conditions Generation

Simple Recursive Program

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R[x]]]$

is correct if the verification conditions hold

- ▶ $(\forall x : I_F[x]) \ (Q[x] \Rightarrow O_F[x, S[x]])$
- ▶ $(\forall x, y : I_F[x]) \ (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow O_F[x, C[x, y]])$
- ▶ $(\forall x : I_F[x]) \ (\neg Q[x] \Rightarrow F'[R[x]] = 0)$
- ▶ where:

$F'[x] = \text{If } Q[x] \text{ then } 0 \text{ else } F'[R[x]]$

Verification Conditions Generation

Simple Recursive Program

$$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R[x]]]$$

is correct if the verification conditions hold

- ▶ $(\forall x : I_F[x]) \ (Q[x] \Rightarrow O_F[x, S[x]])$
- ▶ $(\forall x, y : I_F[x]) \ (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow O_F[x, C[x, y]])$
- ▶ $(\forall x : I_F[x]) \ (\neg Q[x] \Rightarrow F'[R[x]] = 0)$
- ▶ where:

$$F'[x] = \text{If } Q[x] \text{ then } 0 \text{ else } F'[R[x]]$$

Verification Conditions Generation

Simple Recursive Program

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R[x]]]$

is correct if the verification conditions hold

- ▶ $(\forall x : I_F[x]) (Q[x] \Rightarrow O_F[x, S[x]])$
- ▶ $(\forall x, y : I_F[x]) (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow O_F[x, C[x, y]])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow F'[R[x]] = 0)$
- ▶ where:

$F'[x] = \text{If } Q[x] \text{ then } 0 \text{ else } F'[R[x]]$

Verification Conditions Generation

Simple Recursive Program

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R[x]]]$

is correct if the verification conditions hold

- ▶ $(\forall x : I_F[x]) (Q[x] \Rightarrow O_F[x, S[x]])$
- ▶ $(\forall x, y : I_F[x]) (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow O_F[x, C[x, y]])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow F'[R[x]] = 0)$
- ▶ where:

$F'[x] = \text{If } Q[x] \text{ then } 0 \text{ else } F'[R[x]]$

Verification Conditions Generation

Simple Recursive Program

$$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R[x]]]$$

is correct if the verification conditions hold

- ▶ $(\forall x : I_F[x]) \ (Q[x] \Rightarrow O_F[x, S[x]])$
- ▶ $(\forall x, y : I_F[x]) \ (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow O_F[x, C[x, y]])$
- ▶ $(\forall x : I_F[x]) \ (\neg Q[x] \Rightarrow F'[R[x]] = 0)$
- ▶ where:

$$F'[x] = \text{If } Q[x] \text{ then } 0 \text{ else } F'[R[x]]$$

Soundness and Completeness

$\langle \text{Program}, \text{Specification} \rangle \xrightarrow{\text{VCG}} \text{VerificationConditions}$

$\langle F[x], \langle I_F[x], O_F[x, F[x]] \rangle \rangle \xrightarrow{\text{VCG}} \text{VerificationConditions}$

Soundness

if $\models \varphi_1[x] \wedge \cdots \wedge \varphi_n[x]$
then $\forall n (I[n] \Rightarrow F[n] \downarrow \wedge O[n, F[n]])$

Completeness

if $\forall n (I[n] \Rightarrow F[n] \downarrow \wedge O[n, F[n]])$
then $\models \varphi_1[x] \wedge \cdots \wedge \varphi_n[x]$

Soundness and Completeness

$\langle \text{Program}, \text{Specification} \rangle \xrightarrow{\text{VCG}} \text{VerificationConditions}$

$\langle F[x], \langle I_F[x], O_F[x, F[x]] \rangle \rangle \xrightarrow{\text{VCG}} \text{VerificationConditions}$

Soundness

if $\models \varphi_1[x] \wedge \dots \wedge \varphi_n[x]$
then $\forall n (I[n] \Rightarrow F[n] \downarrow \wedge O[n, F[n]])$

Completeness

if $\forall n (I[n] \Rightarrow F[n] \downarrow \wedge O[n, F[n]])$
then $\models \varphi_1[x] \wedge \dots \wedge \varphi_n[x]$

Soundness and Completeness

$\langle \text{Program}, \text{Specification} \rangle \xrightarrow{\text{VCG}} \text{VerificationConditions}$

$\langle F[x], \langle I_F[x], O_F[x, F[x]] \rangle \rangle \xrightarrow{\text{VCG}} \text{VerificationConditions}$

Soundness

if $\models \varphi_1[x] \wedge \cdots \wedge \varphi_n[x]$
then $\forall n (I[n] \Rightarrow F[n] \downarrow \wedge O[n, F[n]])$

Completeness

if $\forall n (I[n] \Rightarrow F[n] \downarrow \wedge O[n, F[n]])$
then $\models \varphi_1[x] \wedge \cdots \wedge \varphi_n[x]$

Soundness and Completeness

$\langle \text{Program}, \text{Specification} \rangle \xrightarrow{\text{VCG}} \text{VerificationConditions}$

$\langle F[x], \langle I_F[x], O_F[x, F[x]] \rangle \rangle \xrightarrow{\text{VCG}} \text{VerificationConditions}$

Soundness

if $\models \varphi_1[x] \wedge \cdots \wedge \varphi_n[x]$
then $\forall n (I[n] \Rightarrow F[n] \downarrow \wedge O[n, F[n]])$

Completeness

if $\forall n (I[n] \Rightarrow F[n] \downarrow \wedge O[n, F[n]])$
then $\models \varphi_1[x] \wedge \cdots \wedge \varphi_n[x]$

Example

Sum $(\forall n : \mathbb{N}) (Sum[n] = \frac{n(n+1)}{2})$

$Sum[n] = \begin{cases} \text{If } n = 0 \text{ then } 0 \\ \text{else } n + Sum[n - 1]. \end{cases}$

is coherent if

$\sim (\forall n : \mathbb{N}) (n \neq 0 \rightarrow n \in \mathbb{N})$

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

Example

Sum $(\forall n : \mathbb{N}) (Sum[n] = \frac{n(n+1)}{2})$

$Sum[n] = \begin{array}{l} \textbf{if } n = 0 \textbf{ then } 0 \\ \textbf{else } n + Sum[n - 1]. \end{array}$

is coherent if

- ▶ $(\forall n : \mathbb{N}) (n \neq 0 \Rightarrow n \in \mathbb{N})$
- ▶ $(\forall n : \mathbb{N}) (n = 0 \Rightarrow T)$

Example

Sum $(\forall n : \mathbb{N}) (Sum[n] = \frac{n(n+1)}{2})$

$Sum[n] = \begin{array}{l} \textbf{if } n = 0 \textbf{ then } 0 \\ \textbf{else } n + Sum[n - 1]. \end{array}$

is coherent if

- ▶ $(\forall n : \mathbb{N}) (n \neq 0 \Rightarrow n \in \mathbb{N})$
- ▶ $(\forall n : \mathbb{N}) (n = 0 \Rightarrow T)$

Example

Sum $(\forall n : \mathbb{N}) (Sum[n] = \frac{n(n+1)}{2})$

$Sum[n] = \begin{array}{l} \textbf{if } n = 0 \textbf{ then } 0 \\ \textbf{else } n + Sum[n - 1]. \end{array}$

is coherent if

- ▶ $(\forall n : \mathbb{N}) (n \neq 0 \Rightarrow n \in \mathbb{N})$
- ▶ $(\forall n : \mathbb{N}) (n = 0 \Rightarrow T)$

Example

Sum $(\forall n : \mathbb{N}) (Sum[n] = \frac{n(n+1)}{2})$

$Sum[n] = \begin{cases} \text{If } n = 0 \text{ then } 0 \\ \text{else } n + Sum[n - 1]. \end{cases}$

is correct if and only if

- ▶ $(\forall n : \mathbb{N}) (n = 0 \Rightarrow 0 = \frac{n(n+1)}{2})$
- ▶ $(\forall n, m : \mathbb{N}) (n \neq 0 \wedge m = \frac{(n-1)((n-1)+1)}{2} \Rightarrow n + m = \frac{n(n+1)}{2})$
- ▶ $(\forall n : \mathbb{N}) (Sum'[n] = 0)$
- ▶ where:

$Sum'[n] = \begin{cases} \text{If } n = 0 \text{ then } 0 \\ \text{else } Sum'[n - 1] \end{cases}$

Example

Sum $(\forall n : \mathbb{N}) (Sum[n] = \frac{n(n+1)}{2})$

$Sum[n] = \begin{array}{l} \textbf{if } n = 0 \textbf{ then } 0 \\ \textbf{else } n + Sum[n - 1]. \end{array}$

is correct if and only if

- ▶ $(\forall n : \mathbb{N}) (n = 0 \Rightarrow 0 = \frac{n(n+1)}{2})$
- ▶ $(\forall n, m : \mathbb{N}) (n \neq 0 \wedge m = \frac{(n-1)((n-1)+1)}{2} \Rightarrow n + m = \frac{n(n+1)}{2})$
- ▶ $(\forall n : \mathbb{N}) (Sum'[n] = 0)$
- ▶ where:

$Sum'[n] = \text{If } n = 0 \text{ then } 0 \text{ else } Sum'[n - 1]$

Example

Sum $(\forall n : \mathbb{N}) (Sum[n] = \frac{n(n+1)}{2})$

$Sum[n] = \begin{array}{l} \textbf{if } n = 0 \textbf{ then } 0 \\ \textbf{else } n + Sum[n - 1]. \end{array}$

is correct if and only if

- ▶ $(\forall n : \mathbb{N}) (n = 0 \Rightarrow 0 = \frac{n(n+1)}{2})$
- ▶ $(\forall n, m : \mathbb{N}) (n \neq 0 \wedge m = \frac{(n-1)((n-1)+1)}{2} \Rightarrow n + m = \frac{n(n+1)}{2})$
- ▶ $(\forall n : \mathbb{N}) (Sum'[n] = 0)$
- ▶ where:

$Sum'[n] = \text{If } n = 0 \text{ then } 0 \text{ else } Sum'[n - 1]$

Example

Sum $(\forall n : \mathbb{N}) (Sum[n] = \frac{n(n+1)}{2})$

$Sum[n] = \begin{array}{l} \textbf{if } n = 0 \textbf{ then } 0 \\ \textbf{else } n + Sum[n - 1]. \end{array}$

is correct if and only if

- ▶ $(\forall n : \mathbb{N}) (n = 0 \Rightarrow 0 = \frac{n(n+1)}{2})$
- ▶ $(\forall n, m : \mathbb{N}) (n \neq 0 \wedge m = \frac{(n-1)((n-1)+1)}{2} \Rightarrow n + m = \frac{n(n+1)}{2})$
- ▶ $(\forall n : \mathbb{N}) (Sum'[n] = 0)$
- ▶ where:

$Sum'[n] = \text{If } n = 0 \text{ then } 0 \text{ else } Sum'[n - 1]$

Example

Sum $(\forall n : \mathbb{N}) (Sum[n] = \frac{n(n+1)}{2})$

$Sum[n] = \begin{cases} \text{If } n = 0 \text{ then } 0 \\ \text{else } n + Sum[n - 1]. \end{cases}$

is correct if and only if

- ▶ $(\forall n : \mathbb{N}) (n = 0 \Rightarrow 0 = \frac{n(n+1)}{2})$
- ▶ $(\forall n, m : \mathbb{N}) (n \neq 0 \wedge m = \frac{(n-1)((n-1)+1)}{2} \Rightarrow n + m = \frac{n(n+1)}{2})$
- ▶ $(\forall n : \mathbb{N}) (Sum'[n] = 0)$
- ▶ where:

$Sum'[n] = \begin{cases} \text{If } n = 0 \text{ then } 0 \\ \text{else } Sum'[n - 1] \end{cases}$

Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] =$ **If** $n = 0$ **then** 1
 elseif Even[n] **then** $P[x * x, n/2]$
 else $x * P[x * x, (n - 1)/2]$.

is coherent if

$$\sim (\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \rightarrow \text{Even}[n])$$

and the recursive step is valid for all even numbers.

Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] = \begin{aligned} & \textbf{If } n = 0 \textbf{ then } 1 \\ & \textbf{elseif } \text{Even}[n] \textbf{ then } P[x * x, n/2] \\ & \textbf{else } x * P[x * x, (n - 1)/2]. \end{aligned}$

is coherent if

- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \Rightarrow \text{Even}[n])$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \Rightarrow \text{Even}[n - 1])$

Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] =$ **If** $n = 0$ **then** 1
 elseif Even[n] **then** $P[x * x, n/2]$
 else $x * P[x * x, (n - 1)/2]$.

is coherent if

- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \Rightarrow \text{Even}[n])$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \Rightarrow \text{Even}[n - 1])$

Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] =$ **If** $n = 0$ **then** 1
 elseif Even[n] **then** $P[x * x, n/2]$
 else $x * P[x * x, (n - 1)/2]$.

is coherent if

- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \Rightarrow \text{Even}[n])$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \Rightarrow \text{Even}[n - 1])$

Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] =$ **If** $n = 0$ **then** 1
 elseif Even[n] **then** $P[x * x, n/2]$
 else $x * P[x * x, (n - 1)/2]$.

is correct if and only if

- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n = 0 \Rightarrow 1 = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \Rightarrow n/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \wedge m = (x * x)^{n/2} \Rightarrow m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \Rightarrow (n - 1)/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \wedge m = (x * x)^{(n-1)/2} \Rightarrow x * m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (P'[x, n] = 0)$

Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] =$ **If** $n = 0$ **then** 1
 elseif Even[n] **then** $P[x * x, n/2]$
 else $x * P[x * x, (n - 1)/2]$.

is correct if and only if

- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n = 0 \Rightarrow 1 = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \Rightarrow n/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \wedge m = (x * x)^{n/2} \Rightarrow m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \Rightarrow (n - 1)/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \wedge m = (x * x)^{(n-1)/2} \Rightarrow x * m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (P'[x, n] = 0)$

Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] =$ **If** $n = 0$ **then** 1
 elseif Even[n] **then** $P[x * x, n/2]$
 else $x * P[x * x, (n - 1)/2]$.

is correct if and only if

- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n = 0 \Rightarrow 1 = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \Rightarrow n/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \wedge m = (x * x)^{n/2} \Rightarrow m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \Rightarrow (n - 1)/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \wedge m = (x * x)^{(n-1)/2} \Rightarrow x * m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (P'[x, n] = 0)$

Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] =$ **If** $n = 0$ **then** 1
 elseif Even[n] **then** $P[x * x, n/2]$
 else $x * P[x * x, (n - 1)/2]$.

is correct if and only if

- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n = 0 \Rightarrow 1 = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \Rightarrow n/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \wedge m = (x * x)^{n/2} \Rightarrow m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \Rightarrow (n - 1)/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \wedge m = (x * x)^{(n-1)/2} \Rightarrow x * m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (P'[x, n] = 0)$

Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] =$ **If** $n = 0$ **then** 1
 elseif Even[n] **then** $P[x * x, n/2]$
 else $x * P[x * x, (n - 1)/2]$.

is correct if and only if

- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n = 0 \Rightarrow 1 = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \Rightarrow n/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \wedge m = (x * x)^{n/2} \Rightarrow m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \Rightarrow (n - 1)/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \wedge m = (x * x)^{(n-1)/2} \Rightarrow x * m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (P'[x, n] = 0)$

Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] =$ **If** $n = 0$ **then** 1
 elseif Even[n] **then** $P[x * x, n/2]$
 else $x * P[x * x, (n - 1)/2]$.

is correct if and only if

- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n = 0 \Rightarrow 1 = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \Rightarrow n/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \wedge m = (x * x)^{n/2} \Rightarrow m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \Rightarrow (n - 1)/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \wedge m = (x * x)^{(n-1)/2} \Rightarrow x * m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (P'[x, n] = 0)$

Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] =$ **If** $n = 0$ **then** 1
 elseif Even[n] **then** $P[x * x, n/2]$
 else $x * P[x * x, (n - 1)/2]$.

is correct if and only if

- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n = 0 \Rightarrow 1 = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \Rightarrow n/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \wedge m = (x * x)^{n/2} \Rightarrow m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \Rightarrow (n - 1)/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \wedge m = (x * x)^{(n-1)/2} \Rightarrow x * m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (P'[x, n] = 0)$

Counter-Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] =$ **If** $n = 0$ **then** **0**
 elseif Even[n] **then** $P[x * x, n/2]$
 else $x * P[x * x, (n - 1)/2]$.

is correct if and only if

- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n = 0 \Rightarrow 0 = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \Rightarrow n/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \wedge m = (x * x)^{n/2} \Rightarrow m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \Rightarrow (n - 1)/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \wedge m = (x * x)^{(n-1)/2} \Rightarrow x * m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (P'[x, n] = 0)$

Counter-Example

Binary powering $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) P[x, n] = x^n$

$P[x, n] =$ **If** $n = 0$ **then** **0**
 elseif Even[n] **then** $P[x * x, n/2]$
 else $x * P[x * x, (n - 1)/2]$.

is correct if and only if

- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n = 0 \Rightarrow \text{0} = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \Rightarrow n/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \text{Even}[n] \wedge m = (x * x)^{n/2} \Rightarrow m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \Rightarrow (n - 1)/2 \in \mathbb{N})$
- ▶ $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \neg \text{Even}[n] \wedge m = (x * x)^{(n-1)/2} \Rightarrow x * m = x^n)$
- ▶ $(\forall x : \mathbb{R})(\forall n : \mathbb{N}) (P'[x, n] = 0)$

Outline

Functional Program Verification
Total Correctness
Building up Correct Programs
Coherent Programs. Recursion
Soundness and Completeness

Conclusion and Discussions

Conclusions and Discussion

- ▶ The problem of proving program correctness is translated into a problem of proving first order formulae;
- ▶ Prove by hand;
- ▶ Prove by an automatic theorem prover.

Conclusions and Discussion

- ▶ The problem of proving program correctness is translated into a problem of proving first order formulae;
- ▶ Prove by hand;
- ▶ Prove by an automatic theorem prover.

Conclusions and Discussion

- ▶ The problem of proving program correctness is translated into a problem of proving first order formulae;
- ▶ Prove by hand;
- ▶ Prove by an automatic theorem prover.