

Functional Program Verification in Theorema. Recent Achievements and Perspectives

Nikolaj Popov and Tudor Jebelean

Research Institute for Symbolic Computation, Linz

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

Outline

Functional Program Verification
Total Correctness
Building up Correct Programs
Coherent Programs. Recursion
Soundness and Completeness
Double (Multiple) Recursion Program Schemata
Mutual Recursion Program Schemata

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

Coherence conditions for *if-then-else*

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

Input condition of H is valid

Input condition of G is valid

Coherence conditions for *Superposition*

$$F[x] = H[G_1[x], G_2[x]]$$

Input condition of G_1 is valid

Input condition of G_2 is valid

Building up Correct Programs

Appropriate values for the auxiliary functions

No input condition of an auxiliary function will be violated

Coherence conditions for *if-then-else*

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

- $(\forall x : I[x]) \ (Q[x] \rightarrow I_H[x])$
- $(\forall x : I[x]) \ (\neg Q[x] \rightarrow I_G[x])$

Coherence conditions for *Superposition*

$F[x] = H[G_1[x], G_2[x]]$

$G_1[x] \in \mathcal{I}_1, G_2[x] \in \mathcal{I}_2$

$I_1 \cap I_2 = \emptyset$

Building up Correct Programs

Appropriate values for the auxiliary functions

No input condition of an auxiliary function will be violated

Coherence conditions for *if-then-else*

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

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

Coherence conditions for *Superposition*

$F[x] = H[G_1[x], G_2[x]]$

Condition 1: $I_F[x] \Rightarrow I_{G_1}[x]$

Condition 2: $I_F[x] \Rightarrow I_{G_2}[x]$

Building up Correct Programs

Appropriate values for the auxiliary functions

No input condition of an auxiliary function will be violated

Coherence conditions for *if-then-else*

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

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

Coherence conditions for *Superposition*

$F[x] = H[G_1[x], G_2[x]]$

- ▶ $(\forall x : I_F[x]) (G_1[x] \wedge G_2[x])$
- ▶ $(\forall x : I_F[x]) (\forall y_1, y_2) (G_1(x, y_1) \wedge G_2(x, y_2) \rightarrow H(y_1, y_2))$

Building up Correct Programs

Appropriate values for the auxiliary functions

No input condition of an auxiliary function will be violated

Coherence conditions for *if-then-else*

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

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

Coherence conditions for *Superposition*

$F[x] = H[G_1[x], G_2[x]]$

- ▶ $(\forall x : I_F[x]) (I_{G_1}[x] \wedge I_{G_2}[x])$
- ▶ $(\forall x : I_F[x]) (\forall y_1, y_2) (O_{G_1}[x, y_1] \wedge O_{G_2}[x, y_2] \implies I_H[y_1, y_2])$

Building up Correct Programs

Appropriate values for the auxiliary functions

No input condition of an auxiliary function will be violated

Coherence conditions for *if-then-else*

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

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

Coherence conditions for *Superposition*

$F[x] = H[G_1[x], G_2[x]]$

- ▶ $(\forall x : I_F[x]) (I_{G_1}[x] \wedge I_{G_2}[x])$
- ▶ $(\forall x : I_F[x]) (\forall y_1, y_2) (O_{G_1}[x, y_1] \wedge O_{G_2}[x, y_2] \implies I_H[y_1, y_2])$

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 call $F[R[x]]$ satisfies the same condition

inductively

base case

recursion step

induction hypothesis

recursion step

induction hypothesis

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

- ▶ $(\forall x : I_F[x]) \ (Q[x] \Rightarrow O_F[x, S[x]])$
- ▶ $(\forall x, y : I_F[x]) \ (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow O_F[x, C[x, y]])$
- ▶ $(\forall x : I_F[x]) \ (F'[x] = \mathbb{T})$
- ▶ where:

$F'[x] = \text{If } Q[x] \text{ then } \mathbb{T} \text{ else } F'[R[x]]$

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

- ▶ $(\forall x : I_F[x]) \ (Q[x] \Rightarrow O_F[x, S[x]])$
- ▶ $(\forall x, y : I_F[x]) \ (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow O_F[x, C[x, y]])$
- ▶ $(\forall x : I_F[x]) \ (F'[x] = \mathbb{T})$
- ▶ where:

$F'[x] = \text{If } Q[x] \text{ then } \mathbb{T} \text{ else } F'[R[x]]$

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

- ▶ $(\forall x : I_F[x]) (Q[x] \Rightarrow O_F[x, S[x]])$
- ▶ $(\forall x, y : I_F[x]) (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow O_F[x, C[x, y]])$
- ▶ $(\forall x : I_F[x]) (F'[x] = \mathbb{T})$
- ▶ where:

$F'[x] = \text{If } Q[x] \text{ then } \mathbb{T} \text{ else } F'[R[x]]$

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

- ▶ $(\forall x : I_F[x]) \ (Q[x] \Rightarrow O_F[x, S[x]])$
- ▶ $(\forall x, y : I_F[x]) \ (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow O_F[x, C[x, y]])$
- ▶ $(\forall x : I_F[x]) \ (F'[x] = \mathbb{T})$
- ▶ where:

$F'[x] = \text{If } Q[x] \text{ then } \mathbb{T} \text{ else } F'[R[x]]$

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

- ▶ $(\forall x : I_F[x]) (Q[x] \Rightarrow O_F[x, S[x]])$
- ▶ $(\forall x, y : I_F[x]) (\neg Q[x] \wedge O_F[R[x], y] \Rightarrow O_F[x, C[x, y]])$
- ▶ $(\forall x : I_F[x]) (F'[x] = \mathbb{T})$
- ▶ where:

$F'[x] = \text{If } Q[x] \text{ then } \mathbb{T} \text{ else } F'[R[x]]$

Soundness and Completeness

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

$\langle F[x], \langle I_F[x], O_F[x, F[x]] \rangle \rangle \xrightarrow{\text{VCG}} \varphi_1[x] \wedge \dots \wedge \varphi_n[x]$

Soundness

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

Completeness

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

Soundness and Completeness

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

$\langle F[x], \langle I_F[x], O_F[x, F[x]] \rangle \rangle \xrightarrow{\text{VCG}} \varphi_1[x] \wedge \dots \wedge \varphi_n[x]$

Soundness

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

Completeness

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

Soundness and Completeness

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

$\langle F[x], \langle I_F[x], O_F[x, F[x]] \rangle \rangle \xrightarrow{\text{VCG}} \varphi_1[x] \wedge \dots \wedge \varphi_n[x]$

Soundness

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

Completeness

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

Soundness and Completeness

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

$\langle F[x], \langle I_F[x], O_F[x, F[x]] \rangle \rangle \xrightarrow{\text{VCG}} \varphi_1[x] \wedge \dots \wedge \varphi_n[x]$

Soundness

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

Completeness

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

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

$\vdash (\forall x : \mathbb{R})(\forall n : \mathbb{N}) (n \neq 0 \wedge \dots \rightarrow \dots)$

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

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

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

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

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

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

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

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

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

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

Counter-Example

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

$P[x, n] =$ **If** $n = 0$ **then** 1
 elseif Even[n] **then** $P[\textcolor{red}{x}, n/2]$ % but not $x * x$
 else $x * P[x * x, (n - 1)/2]$.

is correct if and only if

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

Counter-Example

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

$P[x, n] =$ **If** $n = 0$ **then** 1
 elseif Even[n] **then** $P[\textcolor{red}{x}, n/2]$ % but not $x * x$
 else $x * P[x * x, (n - 1)/2]$.

is correct if and only if

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

General Recursive Schemata

Double (Multiple) Recursion Program Schemata

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R_1[x]], F[R_2[x]]]$

Conditions for coherence

$$\Rightarrow (\forall x : I[x]) (Q[x] \rightarrow I[x])$$

$$\Rightarrow (\forall x : I[x]) (\neg Q[x] \rightarrow I[R_1[x]])$$

$$\Rightarrow (\forall x : I[x]) (\neg Q[x] \rightarrow I[R_2[x]])$$

$$\Rightarrow (\forall x : I[x]) (\neg Q[x] \rightarrow I_n[x])$$

$$\Rightarrow (\forall x : I[x]) (\neg Q[x] \rightarrow I_n[x])$$

$$\Rightarrow (\forall x, y, z : I[x]) (\neg Q[x] \wedge O_R[R_1[x], y] \wedge O_R[R_2[x], z] \Rightarrow I[x, y, z])$$

General Recursive Schemata

Double (Multiple) Recursion Program Schemata

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R_1[x]], F[R_2[x]]]$

Conditions for coherence

- ▶ $(\forall x : I_F[x]) (Q[x] \Rightarrow I_S[x])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_F[R_1[x]])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_F[R_2[x]])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_{R_1}[x])$
- ▶ $(\forall x : I_F[x]) (\neg Q[x] \Rightarrow I_{R_2}[x])$
- ▶ $(\forall x, y, z : I_F[x]) (\neg Q[x] \wedge O_F[R_1[x], y] \wedge O_F[R_2[x], z] \Rightarrow I_C[x, y, z])$

General Recursive Schemata

Double (Multiple) Recursion Program Schemata

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R_1[x]], F[R_2[x]]]$

Conditions for Partial Correctness

- ▶ $(\forall x : I_F[x]) \ (Q[x] \Rightarrow O_F[x, S[x]])$

- ▶ $(\forall x, y, z : I_F[x]) \ (\neg Q[x] \wedge O_F[R_1[x], y] \wedge O_F[R_2[x], z] \Rightarrow O_F[x, C[x, y, z]])$

General Recursive Schemata

Double (Multiple) Recursion Program Schemata

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R_1[x]], F[R_2[x]]]$

Condition for Termination

- ▶ $(\forall x : I_F[x]) (F'[x] = \mathbb{T})$
- ▶ where:

$F'[x] = \text{If } Q[x] \text{ then } \mathbb{T} \text{ else } F'[R_1[x]] \wedge F'[R_2[x]]$

General Recursive Schemata

Double (Multiple) Recursion Program Schemata

$F[x] = \text{If } Q[x] \text{ then } S[x] \text{ else } C[x, F[R_1[x]], F[R_2[x]]]$

Condition for Termination

- ▶ $(\forall x : I_F[x]) \ (F'[x] = \mathbb{T})$
- ▶ where:

$F'[x] = \text{If } Q[x] \text{ then } \mathbb{T} \text{ else } F'[R_1[x]] \wedge F'[R_2[x]]$

Mutual Recursion Program Schemata

$$F[x] = F_1[x]$$

$$F_1[x] = \text{If } Q_1[x] \text{ then } S_1[x] \text{ else } C_1[x, F_2[R_1[x]]]$$

$$F_2[x] = \text{If } Q_2[x] \text{ then } S_2[x] \text{ else } C_2[x, F_1[R_2[x]]]$$

Conditions for Coherence

$$\rightarrow (\forall x : I_1[x]) (Q_1[x] \Rightarrow I_2[x])$$

$$\rightarrow (\forall x : I_1[x]) (\neg Q_1[x] \Rightarrow I_2[R_1[x]])$$

$$\rightarrow (\forall x : I_1[x]) (\neg Q_1[x] \Rightarrow I_1[x])$$

$$\rightarrow (\forall x, y : I_2[x]) (\neg Q_1[x] \wedge Q_2[R_1[x], y] \Rightarrow I_2[x, y])$$

Mutual Recursion Program Schemata

$$F[x] = F_1[x]$$

$$F_1[x] = \text{If } Q_1[x] \text{ then } S_1[x] \text{ else } C_1[x, F_2[R_1[x]]]$$

$$F_2[x] = \text{If } Q_2[x] \text{ then } S_2[x] \text{ else } C_2[x, F_1[R_2[x]]]$$

Conditions for Coherence

- ▶ $(\forall x : I_{F_1}[x]) \ (Q_1[x] \Rightarrow I_{S_1}[x])$
- ▶ $(\forall x : I_{F_1}[x]) \ (\neg Q_1[x] \Rightarrow I_{F_2}[R_1[x]])$
- ▶ $(\forall x : I_{F_1}[x]) \ (\neg Q_1[x] \Rightarrow I_{R_1}[x])$
- ▶ $(\forall x, y : I_{F_1}[x]) \ (\neg Q_1[x] \wedge O_{F_2}[R_1[x], y] \Rightarrow I_{C_1}[x, y])$

Mutual Recursion Program Schemata

$$F[x] = F_1[x]$$

$$F_1[x] = \text{If } Q_1[x] \text{ then } S_1[x] \text{ else } C_1[x, F_2[R_1[x]]]$$

$$F_2[x] = \text{If } Q_2[x] \text{ then } S_2[x] \text{ else } C_2[x, F_1[R_2[x]]]$$

Conditions for Partial Correctness

$$\rightarrow (\forall x : I_1[x]) (Q_1[x] \Rightarrow O_2[x, S_1[x]])$$

$$\rightarrow (\forall x, y : I_1[x]) (\neg Q_1[x] \wedge O_2[R_1[x], y] \Rightarrow O_2[x, G(x, y)])$$

$$\rightarrow (\forall x : I_2[x]) (Q_2[x] \Rightarrow O_1[x, S_2[x]])$$

$$\rightarrow (\forall x, y : I_2[x]) (\neg Q_2[x] \wedge O_1[R_2[x], y] \Rightarrow O_1[x, G(x, y)])$$

Mutual Recursion Program Schemata

$$F[x] = F_1[x]$$

$$F_1[x] = \text{If } Q_1[x] \text{ then } S_1[x] \text{ else } C_1[x, F_2[R_1[x]]]$$

$$F_2[x] = \text{If } Q_2[x] \text{ then } S_2[x] \text{ else } C_2[x, F_1[R_2[x]]]$$

Conditions for Partial Correctness

- ▶ $(\forall x : I_{F_1}[x]) \ (Q_1[x] \Rightarrow O_{F_1}[x, S_1[x]])$
- ▶ $(\forall x, y : I_{F_1}[x]) \ (\neg Q_1[x] \wedge O_{F_2}[R_1[x], y] \Rightarrow O_{F_1}[x, C_1[x, y]])$
- ▶ $(\forall x : I_{F_2}[x]) \ (Q_2[x] \Rightarrow O_{F_2}[x, S_2[x]])$
- ▶ $(\forall x, y : I_{F_2}[x]) \ (\neg Q_2[x] \wedge O_{F_1}[R_2[x], y] \Rightarrow O_{F_2}[x, C_2[x, y]])$

Mutual Recursion Program Schemata

$$F[x] = F_1[x]$$

$$F_1[x] = \text{If } Q_1[x] \text{ then } S_1[x] \text{ else } C_1[x, F_2[R_1[x]]]$$

$$F_2[x] = \text{If } Q_2[x] \text{ then } S_2[x] \text{ else } C_2[x, F_1[R_2[x]]]$$

Condition for Termination

$$\vdash (\forall x : I_1[x]) \ (F[x] = T)$$

where:

$$R_1[x] = \text{If } Q_1[x] \text{ then } T \text{ else } F_2[R_1[x]]$$

$$R_2[x] = \text{If } Q_2[x] \text{ then } T \text{ else } F_1[R_2[x]]$$

Mutual Recursion Program Schemata

$$F[x] = F_1[x]$$

$$F_1[x] = \text{If } Q_1[x] \text{ then } S_1[x] \text{ else } C_1[x, F_2[R_1[x]]]$$

$$F_2[x] = \text{If } Q_2[x] \text{ then } S_2[x] \text{ else } C_2[x, F_1[R_2[x]]]$$

Condition for Termination

- ▶ $(\forall x : I_{F_1}[x]) (F'_1[x] = \mathbb{T})$
- ▶ where:

$$F'_1[x] = \text{If } Q_1[x] \text{ then } \mathbb{T} \text{ else } F'_2[R_1[x]]$$

$$F'_2[x] = \text{If } Q_2[x] \text{ then } \mathbb{T} \text{ else } F'_1[R_2[x]]$$

Mutual Recursion Program Schemata

$$F[x] = F_1[x]$$

$$F_1[x] = \text{If } Q_1[x] \text{ then } S_1[x] \text{ else } C_1[x, F_2[R_1[x]]]$$

$$F_2[x] = \text{If } Q_2[x] \text{ then } S_2[x] \text{ else } C_2[x, F_1[R_2[x]]]$$

Condition for Termination

- ▶ $(\forall x : I_{F_1}[x]) \ (F'_1[x] = \mathbb{T})$
- ▶ where:

$$F'_1[x] = \text{If } Q_1[x] \text{ then } \mathbb{T} \text{ else } F'_2[R_1[x]]$$

$$F'_2[x] = \text{If } Q_2[x] \text{ then } \mathbb{T} \text{ else } F'_1[R_2[x]]$$

Example Even Numbers

Even numbers $(\forall x : \mathbb{N}) (Even[x] \wedge F[x] = \mathbb{T}) \vee (Odd[x] \wedge F[x] = \mathbb{F})$

$F[x] = EV[x]$

$EV[x] = \text{If } x = 0 \text{ then } \mathbb{T} \text{ else } OD[x - 1]$

$OD[x] = \text{If } x = 0 \text{ then } \mathbb{F} \text{ else } EV[x - 1]$

Coherence Conditions

$\rightarrow (\forall x : x \in \mathbb{N}) (x \neq 0 \longrightarrow \mathbb{T})$

$\rightarrow (\forall x : x \in \mathbb{N}) (x \neq 0 \longrightarrow x - 1 \in \mathbb{N})$

Example Even Numbers

Even numbers $(\forall x : \mathbb{N}) (Even[x] \wedge F[x] = \mathbb{T}) \vee (Odd[x] \wedge F[x] = \mathbb{F})$

$$F[x] = EV[x]$$

$$EV[x] = \text{If } x = 0 \text{ then } \mathbb{T} \text{ else } OD[x - 1]$$

$$OD[x] = \text{If } x = 0 \text{ then } \mathbb{F} \text{ else } EV[x - 1]$$

Coherence Conditions

- ▶ $(\forall x : x \in \mathbb{N}) (\dots \implies \mathbb{T})$
- ▶ $(\forall x : x \in \mathbb{N}) (x \neq 0 \implies x - 1 \in \mathbb{N})$

Example Even Numbers

Even numbers $(\forall x : \mathbb{N}) (Even[x] \wedge F[x] = \mathbb{T}) \vee (Odd[x] \wedge F[x] = \mathbb{F})$

$F[x] = EV[x]$

$EV[x] = \text{If } x = 0 \text{ then } \mathbb{T} \text{ else } OD[x - 1]$

$OD[x] = \text{If } x = 0 \text{ then } \mathbb{F} \text{ else } EV[x - 1]$

Partial Correctness Conditions

$\vdash (\forall x : x \in \mathbb{N}) (x = 0 \rightarrow (Even[x] \wedge T = \mathbb{T}) \vee (Odd[x] \wedge T = \mathbb{F}))$

Partial correctness conditions are used to verify that the program satisfies certain properties.

For example, we can check if the program correctly handles even numbers.

Partial correctness conditions are used to verify that the program satisfies certain properties.

Example Even Numbers

Even numbers $(\forall x : \mathbb{N}) (Even[x] \wedge F[x] = \mathbb{T}) \vee (Odd[x] \wedge F[x] = \mathbb{F})$

$F[x] = EV[x]$

$EV[x] = \text{If } x = 0 \text{ then } \mathbb{T} \text{ else } OD[x - 1]$

$OD[x] = \text{If } x = 0 \text{ then } \mathbb{F} \text{ else } EV[x - 1]$

Partial Correctness Conditions

- ▶ $(\forall x : x \in \mathbb{N}) (x = 0 \implies (Even[x] \wedge T = \mathbb{T}) \vee (Odd[x] \wedge T = \mathbb{F}))$
- ▶ $(\forall x, y : x \in \mathbb{N})$
 $(x \neq 0 \wedge (Even[x - 1] \wedge y = \mathbb{F}) \vee (Odd[x - 1] \wedge y = \mathbb{T}))$
 \implies
 $(Even[x] \wedge y = \mathbb{T}) \vee (Odd[x] \wedge y = \mathbb{F}))$

Example Even Numbers

Even numbers $(\forall x : \mathbb{N}) (Even[x] \wedge F[x] = \mathbb{T}) \vee (Odd[x] \wedge F[x] = \mathbb{F})$

$F[x] = EV[x]$

$EV[x] = \text{If } x = 0 \text{ then } \mathbb{T} \text{ else } OD[x - 1]$

$OD[x] = \text{If } x = 0 \text{ then } \mathbb{F} \text{ else } EV[x - 1]$

Partial Correctness Conditions

- ▶ $(\forall x : x \in \mathbb{N}) (x = 0 \implies (Even[x] \wedge \mathbb{T} = \mathbb{T}) \vee (Odd[x] \wedge \mathbb{T} = \mathbb{F}))$
- ▶ $(\forall x, y : x \in \mathbb{N})$
 $(x \neq 0 \wedge (Even[x - 1] \wedge y = \mathbb{F}) \vee (Odd[x - 1] \wedge y = \mathbb{T}))$
 \implies
 $(Even[x] \wedge y = \mathbb{T}) \vee (Odd[x] \wedge y = \mathbb{F}))$

Example Even Numbers

Even numbers $(\forall x : \mathbb{N}) (Even[x] \wedge F[x] = \mathbb{T}) \vee (Odd[x] \wedge F[x] = \mathbb{F})$

$F[x] = EV[x]$

$EV[x] = \text{If } x = 0 \text{ then } \mathbb{T} \text{ else } OD[x - 1]$

$OD[x] = \text{If } x = 0 \text{ then } \mathbb{F} \text{ else } EV[x - 1]$

Partial Correctness Conditions

- ▶ $(\forall x : x \in \mathbb{N}) (x = 0 \implies (Even[x] \wedge \mathbb{T} = \mathbb{T}) \vee (Odd[x] \wedge \mathbb{T} = \mathbb{F}))$
- ▶ $(\forall x, y : x \in \mathbb{N})$
 $(x \neq 0 \wedge (Even[x - 1] \wedge y = \mathbb{F}) \vee (Odd[x - 1] \wedge y = \mathbb{T}))$
 \implies
 $(Even[x] \wedge y = \mathbb{T}) \vee (Odd[x] \wedge y = \mathbb{F}))$

Example Even Numbers

Even numbers $(\forall x : \mathbb{N}) (Even[x] \wedge F[x] = \mathbb{T}) \vee (Odd[x] \wedge F[x] = \mathbb{F})$

$F[x] = EV[x]$

$EV[x] = \text{If } x = 0 \text{ then } \mathbb{T} \text{ else } OD[x - 1]$

$OD[x] = \text{If } x = 0 \text{ then } \mathbb{F} \text{ else } EV[x - 1]$

terminates if

- ▶ $(\forall x : \mathbb{N}) (F'[x] = \mathbb{T})$
- ▶ where:

$F'[x] = \text{If } x = 0 \text{ then } \mathbb{T}$
 $\text{else } F'[x - 1].$

Example Even Numbers

Even numbers $(\forall x : \mathbb{N}) (Even[x] \wedge F[x] = \mathbb{T}) \vee (Odd[x] \wedge F[x] = \mathbb{F})$

$F[x] = EV[x]$

$EV[x] = \text{If } x = 0 \text{ then } \mathbb{T} \text{ else } OD[x - 1]$

$OD[x] = \text{If } x = 0 \text{ then } \mathbb{F} \text{ else } EV[x - 1]$

terminates if

- ▶ $(\forall x : \mathbb{N}) (F'[x] = \mathbb{T})$
- ▶ where:

$F'[x] = \begin{cases} \mathbb{T} & \text{if } x = 0 \\ F'[x - 1]. & \text{else} \end{cases}$

Outline

Functional Program Verification

Total Correctness

Building up Correct Programs

Coherent Programs. Recursion

Soundness and Completeness

Double (Multiple) Recursion Program Schemata

Mutual Recursion Program Schemata

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.