Functional Program Verification in Theorema. Soundness and Completeness

Nikolaj Popov and Tudor Jebelean

Research Institute for Symbolic Computation, Linz

popov@risc.uni-linz.ac.at

Outline

Outline

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

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

$pow[x, n] =$ **If** $n = 0$ **then** 1 **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

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

$pow[x, n] =$ **If** $n = 0$ **then** 1 **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

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

$pow[x, n] =$ **If** $n = 0$ **then** 1 **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;
- Prove total correctness;

Modularity. After proving correctness, use only the specification.

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

$pow[x, n] = \ldots$

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

$pow[x, n] = \ldots$

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

$pow[x, n] = \ldots$

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

$pow[x, n] = \ldots$

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

$pow[x, n] = \ldots$

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

$F[x] =$ **If** $Q[x]$ **then** $H[x]$ **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 *Composition*

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

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

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*

$F[x] =$ **If** $Q[x]$ **then** $H[x]$ **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 *Composition*

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

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

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*

$F[x] =$ **If** $Q[x]$ **then** $H[x]$ **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 *Composition*

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

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

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*

$F[x] =$ **If** $Q[x]$ **then** $H[x]$ **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 *Composition*

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

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

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*

$F[x] =$ **If** $Q[x]$ **then** $H[x]$ **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 *Composition*

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

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

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*

$F[x] =$ **If** $Q[x]$ **then** $H[x]$ **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 *Composition*

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

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

Coherent Programs

Simple Recursive Programs

$F[x] =$ **If** $Q[x]$ **then** $S[x]$ **else** $C[x, F[R[x]]]$

Conditions for Coherence

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

Coherent Programs

Simple Recursive Programs

$F[x] =$ **If** $Q[x]$ **then** $S[x]$ **else** $C[x, F[R[x]]]$

Conditions for Coherence

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

Verification Conditions Generation

Simple Recursive Program

$F[x] =$ **If** $Q[x]$ **then** $S[x]$ **else** $C[x, F[R[x]]]$

Conditions for Total Correctness

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

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

Verification Conditions Generation

Simple Recursive Program

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

Conditions for Total Correctness

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

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

Soundness and Completeness

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

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

Soundness

if $\models \varphi_1 \wedge \cdots \wedge \varphi_n$

then $\forall x \, (I[x] \Rightarrow F[x]\downarrow \wedge O[x, F[x]])$

Completeness

if $\forall x \, (I[x] \Rightarrow F[x]\downarrow \wedge O[x, F[x]])$

then $\models \varphi_1 \wedge \cdots \wedge \varphi_n$

Soundness and Completeness

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

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

Soundness

if $\models \varphi_1 \wedge \cdots \wedge \varphi_n$

then $\forall x \, (I[x] \Rightarrow F[x]\downarrow \wedge O[x, F[x]])$

Completeness

if $\forall x \, (I[x] \Rightarrow F[x]\downarrow \wedge O[x, F[x]])$

then $\models \varphi_1 \wedge \cdots \wedge \varphi_n$

Soundness and Completeness

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

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

Soundness

if $\models \varphi_1 \wedge \cdots \wedge \varphi_n$

then $\forall x \, (I[x] \Rightarrow F[x]\downarrow \wedge O[x, F[x]])$

Completeness

if $\forall x \, (I[x] \Rightarrow F[x]\downarrow \wedge O[x, F[x]])$

then $\models \varphi_1 \wedge \cdots \wedge \varphi_n$

Soundness and Completeness

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

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

Soundness

if $\models \varphi_1 \wedge \cdots \wedge \varphi_n$

then $\forall x \, (I[x] \Rightarrow F[x]\downarrow \wedge O[x, F[x]])$

Completeness

if $\forall x \, (I[x] \Rightarrow F[x]\downarrow \wedge O[x, F[x]])$

then $\models \varphi_1 \wedge \cdots \wedge \varphi_n$

Example

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

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

is coherent if and only if

- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \ldots \Rightarrow \mathbb{T})$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \mathrm{Even}[n] \Rightarrow \mathrm{Even}[n])$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \neg\mathrm{Even}[n] \Rightarrow \mathrm{Even}[n-1])$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \mathrm{Even}[n] \Rightarrow n/2 \in \mathbb{N})$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \neg\mathrm{Even}[n] \Rightarrow (n-1)/2 \in \mathbb{N})$

Example

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

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

is coherent if and only if

- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \ \ldots \ \Rightarrow \mathbb{T})$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \mathrm{Even}[n] \ \Rightarrow \mathrm{Even}[n])$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \neg\mathrm{Even}[n] \ \Rightarrow \mathrm{Even}[n-1])$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \mathrm{Even}[n] \ \Rightarrow n/2 \in \mathbb{N})$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \neg\mathrm{Even}[n] \ \Rightarrow (n-1)/2 \in \mathbb{N})$

Example

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

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

is correct if and only if

- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n = 0 \ \Rightarrow 1 = x^n)$
- $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \mathrm{Even}[n] \wedge m = (x * x)^{n/2} \ \Rightarrow m = x^n)$
- $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \neg\mathrm{Even}[n] \wedge m = (x * x)^{(n-1)/2} \ \Rightarrow x * m = x^n)$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (P'[x, n] = \mathbb{T})$

Example

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

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

is correct if and only if

- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n = 0\ \Rightarrow 1 = x^n)$
- $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \mathrm{Even}[n] \wedge m = (x * x)^{n/2}\ \Rightarrow m = x^n)$
- $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \neg\mathrm{Even}[n] \wedge m = (x * x)^{(n-1)/2}\ \Rightarrow x * m = x^n)$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (P'[x, n] = \mathbb{T})$

Counter-Example

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

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

is correct if and only if

- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n = 0 \ \Rightarrow \mathbf{0} = x^n)$
- $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \mathrm{Even}[n] \wedge m = (x * x)^{n/2} \ \Rightarrow m = x^n)$
- $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \neg\mathrm{Even}[n] \wedge m = (x * x)^{(n-1)/2} \ \Rightarrow x * m = x^n)$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (P'[x, n] = \mathbb{T})$

Counter-Example

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

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

is correct if and only if

- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n = 0\ \Rightarrow \mathbf{0} = x^n)$
- $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \mathrm{Even}[n] \wedge m = (x * x)^{n/2}\ \Rightarrow m = x^n)$
- $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \neg\mathrm{Even}[n] \wedge m = (x * x)^{(n-1)/2}\ \Rightarrow x * m = x^n)$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (P'[x, n] = \mathbb{T})$

Counter-Example

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

$$
\begin{array}{lll}
P[x,n] = & \textbf{If } n = 0 \textbf{ then } 1 & \\
& \textbf{elseif } \mathrm{Even}[n] \textbf{ then } P[x, n/2] & \% \textit{ but not } x * x \\
& \textbf{else } x * P[x * x, (n-1)/2]. &
\end{array}
$$

is correct if and only if

- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n = 0 \ \Rightarrow 1 = x^n)$
- $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \mathrm{Even}[n] \wedge m = (x)^{n/2} \ \Rightarrow m = x^n)$
- $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \neg\mathrm{Even}[n] \wedge m = (x * x)^{(n-1)/2} \ \Rightarrow x * m = x^n)$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (P'[x, n] = \mathbb{T})$

Counter-Example

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

$$
\begin{array}{lll}
P[x,n] = & \textbf{If } n = 0 \textbf{ then } 1 & \\
& \textbf{elseif } \mathrm{Even}[n] \textbf{ then } P[x, n/2] & \% \textit{ but not } x * x \\
& \textbf{else } x * P[x * x, (n-1)/2]. &
\end{array}
$$

is correct if and only if

- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (n = 0\ \Rightarrow 1 = x^n)$
- $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \mathrm{Even}[n] \wedge m = (x)^{n/2}\ \Rightarrow m = x^n)$
- $(\forall x, m : \mathbb{R})(\forall n : \mathbb{N})\ (n \neq 0 \wedge \neg\mathrm{Even}[n] \wedge m = (x * x)^{(n-1)/2}\ \Rightarrow x * m = x^n)$
- $(\forall x : \mathbb{R})(\forall n : \mathbb{N})\ (P'[x, n] = \mathbb{T})$

General Recursive Schema

$F[x] =$ **If** $Q[x]$ **then** $S[x]$ **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 Schema

$F[x] =$ **If** $Q[x]$ **then** $S[x]$ **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 Schema

$F[x] =$ **If** $Q[x]$ **then** $S[x]$ **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 Schema

$F[x] =$ **If** $Q[x]$ **then** $S[x]$ **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 Schema

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

Condition for Termination

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

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

General Recursive Schema

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

Condition for Termination

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

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

Mutual Recursion Program Schema

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

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

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

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 Schema

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

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

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

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 Schema

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

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

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

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 Schema

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

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

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

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 Schema

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

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

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

Condition for Termination

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

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

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

Mutual Recursion Program Schema

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

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

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

Condition for Termination

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

$$F_1'[x] = \textbf{If } Q_1[x] \textbf{ then } \mathbb{T} \textbf{ else } F_2'[R_1[x]]$$

$$F_2'[x] = \textbf{If } Q_2[x] \textbf{ then } \mathbb{T} \textbf{ else } F_1'[R_2[x]]$$

Example Even and Odd

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

$F[x] = EV[x]$

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

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

is coherent if and only if

- $(\forall x : x \in \mathbb{N})\ (\ldots \Longrightarrow \mathbb{T})$
- $(\forall x : x \in \mathbb{N})\ (x \neq 0 \Longrightarrow x - 1 \in \mathbb{N})$

Example Even and Odd

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

$F[x] = EV[x]$

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

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

is coherent if and only if

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

Example Even and Odd

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

$F[x] = EV[x]$

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

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

is partially correct if an only if

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

Example Even and Odd

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

$F[x] = EV[x]$

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

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

is partially correct if an only if

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

Example Even and Odd

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

$F[x] = EV[x]$

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

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

terminates if and only 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 and Odd

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

$F[x] = EV[x]$

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

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

terminates if and only if

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

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

Outline

Example Factorial

Fact $(\forall n : \mathbb{N})\ (Fact[n] = n!)$

$$Fact[n] = \quad \textbf{If } n = 0 \textbf{ then } 1 \textbf{ else } n * Fact[n-1].$$

terminates if and only if

- $(\forall n : \mathbb{N})\ (Fact'[n] = \mathbf{T})$
- where:

$$Fact'[n] = \quad \textbf{If } n = 0 \textbf{ then } \mathbf{T} \textbf{ else } Fact'[n-1].$$

Example Factorial

Fact $(\forall n : \mathbb{N})\ (Fact[n] = n!)$

$$Fact[n] = \quad \textbf{If } n = 0 \textbf{ then } 1 \textbf{ else } n * Fact[n-1].$$

terminates if and only if

- $(\forall n : \mathbb{N})\ (Fact'[n] = \mathbf{T})$
- where:

$$Fact'[n] = \quad \textbf{If } n = 0 \textbf{ then } \mathbf{T} \textbf{ else } Fact'[n-1].$$

Example Sum

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

$$Sum[n] = \quad \textbf{If } n = 0 \textbf{ then } 0 \\ \textbf{else } n + Sum[n-1].$$

terminates if and only if

- $(\forall n : \mathbb{N})\ (Sum'[n] = \mathbf{T})$
- where:

$$Sum'[n] = \quad \textbf{If } n = 0 \textbf{ then } \mathbf{T} \\ \textbf{else } Sum'[n-1].$$

Example Sum

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

$$Sum[n] = \quad \textbf{If } n = 0 \textbf{ then } 0 \\ \textbf{else } n + Sum[n-1].$$

terminates if and only if

- $(\forall n : \mathbb{N})\ (Sum'[n] = \mathbf{T})$
- where:

$$Sum'[n] = \quad \textbf{If } n = 0 \textbf{ then } \mathbf{T} \\ \textbf{else } Sum'[n-1].$$

Example Even and Odd

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

$F[x] = EV[x]$

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

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

terminates if and only if

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

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

Example Even and Odd

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

$F[x] = EV[x]$

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

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

terminates if and only if

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

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

Outline

Conclusions and Discussion

- The problem of proving program correctness is translated into a problem of proving first order formulae;
- We obtain automatically all the necessary and sufficient conditions for the algorithm to be correct;
- Prove the conditions by an automatic theorem prover;
- Any counter-example for the conditions becomes a counter-example for the algorithm;
- The termination proofs may in many cases be omitted.

Conclusions and Discussion

- The problem of proving program correctness is translated into a problem of proving first order formulae;
- We obtain automatically all the necessary and sufficient conditions for the algorithm to be correct;
- Prove the conditions by an automatic theorem prover;
- Any counter-example for the conditions becomes a counter-example for the algorithm;
- The termination proofs may in many cases be omitted.

Conclusions and Discussion

- The problem of proving program correctness is translated into a problem of proving first order formulae;
- We obtain automatically all the necessary and sufficient conditions for the algorithm to be correct;
- Prove the conditions by an automatic theorem prover;
- Any counter-example for the conditions becomes a counter-example for the algorithm;
- The termination proofs may in many cases be omitted.

Conclusions and Discussion

- The problem of proving program correctness is translated into a problem of proving first order formulae;
- We obtain automatically all the necessary and sufficient conditions for the algorithm to be correct;
- Prove the conditions by an automatic theorem prover;
- Any counter-example for the conditions becomes a counter-example for the algorithm;
- The termination proofs may in many cases be omitted.

Conclusions and Discussion

- The problem of proving program correctness is translated into a problem of proving first order formulae;
- We obtain automatically all the necessary and sufficient conditions for the algorithm to be correct;
- Prove the conditions by an automatic theorem prover;
- Any counter-example for the conditions becomes a counter-example for the algorithm;
- The termination proofs may in many cases be omitted.