Formal Methods（形式化方法）

Lecture 14. Reasoning about Specifications

智能与计算学部 章衡

2021年上学期

Motivation

Features of Z notation

- By using Z notations one can define the specification precisely, which could reduce the misunderstandings in requirement analyses largely
- The formal semantics of Z provides a way to reason about the specification

What can be done by reasoning

- How to assure the specification admitting a desired property?
- How to know whether a program meets the requirements stated in the specification?

Outline

1. Introduction by Example
2. Rigorous Proofs
3. Reasoning about Specifications

Example: Hobby club

- Basic type:

  [Person]

- Global variable:

  $$\mathrm{Max} : \mathbb{N}$$

- State space schema:

  HoClub

  $$s : \mathbb{P}\,\mathrm{Person}$$

  $$\#s \leqslant \mathrm{Max}$$

$$\Delta\mathrm{HoClub} \;\widehat{=}\; \mathrm{HoClub} \land \mathrm{HoClub}'$$

$$\Xi\mathrm{HoClub} \;\widehat{=}\; \Delta\mathrm{HoClub} \mid s' = s$$

Example: Hobby club

EnterClub

$\Delta HoClub$
$p? : Person$

$\#s < Max$
$p? \notin s$
$s' = s \cup \{p?\}$

LeaveClub

$\Delta HoClub$
$p? : Person$

$p? \in s$
$s' = s \setminus \{p?\}$

Example: Hobby club

$$\mathrm{EnterClub} \fatsemi \mathrm{LeaveClub} \vDash \#s < \mathrm{Max} \land s' = s$$

$$\mathrm{Alpha} \mathrel{\widehat{=}} \mathrm{EnterClub} \fatsemi \mathrm{LeaveClub}$$

Alpha

$s, s' : \mathbb{P}\,\mathrm{Person}$
$p? : \mathrm{Person}$

$$\begin{aligned}
&\exists\, s^{+} : \mathbb{P}\,\mathrm{Person} \bullet \\
&\qquad (\#s \leqslant \mathrm{Max} \land \\
&\qquad \#s^{+} \leqslant \mathrm{Max} \land \\
&\qquad \#s' \leqslant \mathrm{Max} \land \\
&\qquad \#s < \mathrm{Max} \land \\
&\qquad p? \notin s \land \\
&\qquad s^{+} = s \cup \{p?\} \land \\
&\qquad p? \in s^{+} \land \\
&\qquad s' = s^{+} \setminus \{p?\})
\end{aligned}$$

Example: Hobby club

If x does not occur in φ, then $\exists x : X \bullet (\varphi \wedge \psi) \equiv \varphi \wedge \exists x : X \bullet \psi$

$$\mathrm{Alpha} \models \mathrm{Alpha}_1$$

Alpha

$s, s' : \mathbb{P}\,\mathrm{Person}$
$p? : \mathrm{Person}$

$$\begin{aligned}
&\exists s^+ : \mathbb{P}\,\mathrm{Person} \bullet \\
&\qquad (\#s \leqslant \mathrm{Max} \wedge \\
&\qquad \#s^+ \leqslant \mathrm{Max} \wedge \\
&\qquad \#s' \leqslant \mathrm{Max} \wedge \\
&\qquad \#s < \mathrm{Max} \wedge \\
&\qquad p? \notin s \wedge \\
&\qquad s^+ = s \cup \{p?\} \wedge \\
&\qquad p? \in s^+ \wedge \\
&\qquad s' = s^+ \setminus \{p?\})
\end{aligned}$$

Alpha_1

$s, s' : \mathbb{P}\,\mathrm{Person}$
$p? : \mathrm{Person}$

$$\begin{aligned}
&\#s \leqslant \mathrm{Max} \\
&\#s' \leqslant \mathrm{Max} \\
&\#s < \mathrm{Max} \\
&p? \notin s \\
&\exists s^+ : \mathbb{P}\,\mathrm{Person} \bullet \\
&\qquad \#s^+ \leqslant \mathrm{Max} \wedge \\
&\qquad s^+ = s \cup \{p?\} \wedge \\
&\qquad p? \in s^+ \wedge \\
&\qquad s' = s^+ \setminus \{p?\})
\end{aligned}$$

Example: Hobby club

By applying the 1-point rule, we have

$$\mathrm{Alpha} \models \mathrm{Alpha}_1 \models \mathrm{Alpha}_2$$

$$\begin{array}{|l}
\textbf{Alpha}_1 \\
\hline
s, s' : \mathbb{P}\,\mathrm{Person} \\
p? : \mathrm{Person} \\
\hline
\#s \leqslant \mathrm{Max} \\
\#s' \leqslant \mathrm{Max} \\
\#s < \mathrm{Max} \\
p? \notin s \\
\exists\, s^+ : \mathbb{P}\,\mathrm{Person} \bullet \\
\qquad \#s^+ \leqslant \mathrm{Max} \land \\
\qquad s^+ = s \cup \{p?\} \land \\
\qquad p? \in s^+ \land \\
\qquad s' = s^+ \setminus \{p?\}) \\
\hline
\end{array}$$

$$\begin{array}{|l}
\textbf{Alpha}_2 \\
\hline
s, s' : \mathbb{P}\,\mathrm{Person} \\
p? : \mathrm{Person} \\
\hline
\#s \leqslant \mathrm{Max} \\
\#s' \leqslant \mathrm{Max} \\
\#s < \mathrm{Max} \\
p? \notin s \\
\#(s \cup \{p?\}) \leqslant \mathrm{Max} \\
p? \in (s \cup \{p?\}) \\
s' = (s \cup \{p?\}) \setminus \{p?\}) \\
\hline
\end{array}$$

Example: Hobby club

From $p? \notin s$, we know that $(s \cup \{p?\}) \setminus \{p?\} = s$. Consequently,

$$Alpha \models Alpha_1 \models Alpha_2 \models Alpha_3 \models \#s < Max \land s' = s$$

Alpha₂

$$\begin{array}{l} s, s' : \mathbb{P}\,Person \\ p? : Person \\ \hline \#s \leqslant Max \\ \#s' \leqslant Max \\ \#s < Max \\ p? \notin s \\ \#(s \cup \{p?\}) \leqslant Max \\ p? \in (s \cup \{p?\}) \\ s' = (s \cup \{p?\}) \setminus \{p?\} \end{array}$$

Alpha₃

$$\begin{array}{l} s, s' : \mathbb{P}\,Person \\ p? : Person \\ \hline \#s \leqslant Max \\ \#s' \leqslant Max \\ \#s < Max \\ p? \notin s \\ \#(s \cup \{p?\}) \leqslant Max \\ p? \in (s \cup \{p?\}) \\ s' = s \end{array}$$

Outline

1. Introduction by Example
2. Rigorous Proofs
3. Reasoning about Specifications

Formal proof vs. rigorous proof (严密证明)

- Formal proofs provide a procedure of rewriting to obtain theorems from inference rules
- The correctness of such proofs is easily checkable
- However, it is hard to construct a formal proof, and the proof maybe tediously long
- In many cases, mathematicians try to find a weaker form of formal proofs, called rigorous proofs
- They believe that every rigorous proof can be converted into a formal proof
- In a rigorous proof, one is allowed to use the properties in set theory and number theory, as well as the method of induction

Method of induction

Definition (Mathematical induction, 数学归纳法)

To prove "for every natural number n it holds that $P(n)$", it suffices to prove both of the following:

1. $P(0)$ holds;
2. $\forall i : \mathbb{N} \bullet (P(i) \Rightarrow P(i+1))$.

Definition (Structural induction, 结构归纳法)

To prove "for every sequence $s : \mathrm{seq}\, X$ it holds that $P(s)$", it suffices to prove both of the following:

1. $P(\langle\rangle)$ holds;
2. $\forall x : X; s : \mathrm{seq}\, X \bullet (P(s) \Rightarrow P(\langle x \rangle \frown s))$.

Method of induction: Example 1

Example

Please prove that, for all sequences s, t, u : seq X, we have

$$s \frown (t \frown u) = (s \frown t) \frown u.$$

Proof.

By definition, it is easy to see that $\langle\rangle \frown s = s$ and $(\langle x\rangle \frown s) \frown t = \langle x\rangle \frown (s \frown t)$. Next we prove the property by an induction on s.

Base case: $\langle\rangle \frown (t \frown u) = t \frown u = (\langle\rangle \frown t) \frown u$.

Inductive step: Assume as inductive hypothesis that $s \frown (t \frown u) = (s \frown t) \frown u$. We need to prove $(\langle x\rangle \frown s) \frown (t \frown u) = ((\langle x\rangle \frown s) \frown t) \frown u$. Note that

$$\begin{aligned}
(\langle x\rangle \frown s) \frown (t \frown u) &= \langle x\rangle \frown (s \frown (t \frown u)) \\
&= \langle x\rangle \frown ((s \frown t) \frown u) \\
&= (\langle x\rangle \frown (s \frown t)) \frown u \\
&= ((\langle x\rangle \frown s) \frown t) \frown u,
\end{aligned}$$

which completes the proof. □

Method of induction: Example 2

Example

Please prove that, for all sequences $s, t : \operatorname{seq} X$, we have

$$\operatorname{rev}(s \frown t) = (\operatorname{rev} t) \frown (\operatorname{rev} s)$$

Proof.

By definition, it is easy to see that $\langle\rangle \frown s = s = s \frown \langle\rangle$ and $\operatorname{rev}(\langle x\rangle \frown t) = (\operatorname{rev} t) \frown \langle x\rangle$. Next we prove the desired property by an induction on s.

Base case: $\operatorname{rev}(\langle\rangle \frown t) = \operatorname{rev} t = (\operatorname{rev} t) \frown \langle\rangle = (\operatorname{rev} t) \frown \operatorname{rev} \langle\rangle$.

Inductive step: Assume as inductive hypothesis that $\operatorname{rev}(s \frown t) = (\operatorname{rev} t) \frown (\operatorname{rev} s)$. We need to prove that $\operatorname{rev}((\langle x\rangle \frown s) \frown t) = (\operatorname{rev} t) \frown \operatorname{rev}(\langle x\rangle \frown s)$. Note that

$$\begin{aligned}
\operatorname{rev}((\langle x\rangle \frown s) \frown t) &= \operatorname{rev}(\langle x\rangle \frown (s \frown t)) \\
&= \operatorname{rev}(s \frown t) \frown \langle x\rangle \\
&= ((\operatorname{rev} t) \frown (\operatorname{rev} s)) \frown \langle x\rangle \\
&= (\operatorname{rev} t) \frown ((\operatorname{rev} s) \frown \langle x\rangle) \\
&= (\operatorname{rev} t) \frown \operatorname{rev}(\langle x\rangle \frown s),
\end{aligned}$$

which completes the proof. □

Exercise

Prove the following by induction: for every sequence s, we have that $\mathrm{rev}(\mathrm{rev}\, \mathrm{s}) = \mathrm{s}$.

Outline

1. Introduction by Example
2. Rigorous Proofs
3. Reasoning about Specifications

Example: Fan ID management

- Basic types:

  [Person, ID]

- State space schema:

$$\begin{array}{|l}\hline \textbf{FID} \\ members : ID \rightarrowtail\!\!\!\!\twoheadrightarrow Person \\ banned : \mathbb{P}\, ID \\ \hline banned \subseteq \operatorname{dom} members \\ \hline \end{array}$$

$$\begin{array}{|l}\hline \textbf{FID}' \\ members' : ID \rightarrowtail\!\!\!\!\twoheadrightarrow Person \\ banned' : \mathbb{P}\, ID \\ \hline banned' \subseteq \operatorname{dom} members' \\ \hline \end{array}$$

- $\Delta FID \mathrel{\widehat{=}} FID \wedge FID'$

  $\Xi FID \mathrel{\widehat{=}} \Delta FID \mid members' = members \wedge banned' = banned$

The initialization theorem (初始化定理)

- Operational schemas: Initialization

$$\begin{array}{|l}
\text{InitFID} \\
\hline
\text{FID}' \\
\hline
\text{members}' = \emptyset \\
\text{banned}' = \emptyset \\
\hline
\end{array}$$

- The initialization theorem: $\models \exists\, \text{FID}' \bullet \text{InitFID}$

  The above is an abbreviation of the following theorem:

$$\models \exists\, \text{members}' : \text{Person} \rightarrowtail\!\!\!\!\!\!\twoheadrightarrow \text{ID}; \text{banned}' : \mathbb{P}\,\text{ID} \bullet$$
$$(\text{banned}' \subseteq \operatorname{dom} \text{members}' \land \text{members}' = \emptyset \land \text{banned}' = \emptyset)$$

Prove the initialization theorem

$$\models \exists\, members' : Person \rightarrowtail\!\!\!\!\!\!\!\rightarrow ID; banned' : \mathbb{P}\, ID \bullet$$
$$(banned' \subseteq \mathrm{dom}\, members' \land members' = \emptyset \land banned' = \emptyset) \quad (1)$$

1-point rule (bidirection)

$$\frac{\Sigma \models \exists x : S \bullet (\varphi \land x = t)}{\Sigma \models t \in S \land \varphi[t/x]} \quad [1\text{-point}] \quad \text{<x does not occur in t>}$$

- By applying the above rule, (1) can be simplified as

$$\models \emptyset \in Person \rightarrowtail\!\!\!\!\!\!\!\rightarrow ID \land \emptyset \in \mathbb{P}\, ID \land \emptyset \subseteq \mathrm{dom}\, \emptyset \quad (2)$$

- To prove this, it is equivalent to prove all of the following:

$$\models \emptyset \in Person \rightarrowtail\!\!\!\!\!\!\!\rightarrow ID,$$
$$\models \emptyset \in \mathbb{P}\, ID,$$
$$\models \emptyset \subseteq \mathrm{dom}\, \emptyset.$$

Precondition of an operation

AddMember

ΔFID
$applicant? : Person$
$id! : ID$

$applicant? \notin \operatorname{ran} members$
$id! \notin \operatorname{dom} members$
$members' = members \cup \{id! \mapsto applicant?\}$
$banned' = banned$

- We need to know when the operation can be executed.
- If such a condition is not true, we need to report an error.

Precondition of an operation

PreAddMember

FID
applicant? : Person

$$
\begin{aligned}
&\exists\, FID'; id! : ID \bullet\\
&\qquad (applicant? \notin \operatorname{ran} members \land\\
&\qquad id! \notin \operatorname{dom} members \land\\
&\qquad members' = members \cup \{id! \mapsto applicant?\} \land\\
&\qquad banned' = banned)
\end{aligned}
$$

- Unfolding the predicate of the above schema, we have

$$
\begin{aligned}
&\exists\, members' : ID \rightarrowtail\!\!\!\!\!\!\rightarrow Person; banned' : \mathbb{P}\, ID; id! : ID \bullet\\
&\qquad (banned' \subseteq \operatorname{dom} members' \land applicant? \notin \operatorname{ran} members \land\\
&\qquad id! \notin \operatorname{dom} members \land members' = members \cup \{id! \mapsto applicant?\} \land\\
&\qquad banned' = banned)
\end{aligned}
$$

Simplification of precondition

Most often used rules for precondition simplification

$$\frac{\Sigma \vDash \exists x : S \bullet (\varphi \wedge x = t)}{\Sigma \vDash t \in S \wedge \varphi[t/x]} \quad [\text{1-point}] \quad <\text{x does not occur in t}>$$

$$\frac{\Sigma \vDash \varphi \wedge \psi}{\Sigma \vDash \varphi} \quad [\wedge] \quad <\Sigma, \varphi \vDash \psi>$$

$$\frac{\Sigma \vDash \varphi}{\Sigma \vDash \varphi'} \quad [=] \quad <\Sigma \vDash t_1 = t_2 \text{ and } \varphi' \text{ is obtained from } \varphi \text{ by substituting } t_2 \text{ for some occurrence of } t_1>$$

Simplification of precondition

$$\begin{aligned}\exists\, & members' : ID \rightarrowtail\!\!\!\!\twoheadrightarrow Person; banned' : \mathbb{P}\, ID; id! : ID \bullet \\ & (banned' \subseteq \mathrm{dom}\, members' \wedge applicant? \notin \mathrm{ran}\, members \wedge \\ & id! \notin \mathrm{dom}\, members \wedge members' = members \cup \{id! \mapsto applicant?\} \wedge \\ & banned' = banned)\end{aligned} \tag{3}$$

- By applying 1-point rule for variable banned′, (3) can be simplified as

$$\begin{aligned}\exists\, & members' : ID \rightarrowtail\!\!\!\!\twoheadrightarrow Person; id! : ID \bullet \\ & (banned \subseteq \mathrm{dom}\, members' \wedge applicant? \notin \mathrm{ran}\, members \wedge \\ & id! \notin \mathrm{dom}\, members \wedge members' = members \cup \{id! \mapsto applicant?\} \wedge \\ & banned \in \mathbb{P}\, ID)\end{aligned} \tag{4}$$

- By applying 1-point rule for variable members′, (4) can be simplified as

$$\begin{aligned}\exists\, id! : ID \bullet\ & (banned \subseteq \mathrm{dom}(members \cup \{id! \mapsto applicant?\}) \wedge applicant? \notin \mathrm{ran}\, members \wedge \\ & id! \notin \mathrm{dom}\, members \wedge members \cup \{id! \mapsto applicant?\} \in ID \rightarrowtail\!\!\!\!\twoheadrightarrow Person \wedge \\ & banned \in \mathbb{P}\, ID)\end{aligned} \tag{5}$$

Simplification of precondition

$$\exists\, id! : ID \bullet (banned \subseteq \mathrm{dom}(members \cup \{id! \mapsto applicant?\}) \land applicant? \notin \mathrm{ran}\, members \land id! \notin \mathrm{dom}\, members \land members \cup \{id! \mapsto applicant?\} \in ID \rightarrowtail\!\!\!\!\!\rightarrow Person \land banned \in \mathbb{P}\, ID) \tag{6}$$

- By the declaration $banned : \mathbb{P}\, ID$ we know $banned \in \mathbb{P}\, ID$. Consequently, (6) can be equivalently rewritten as

$$\exists\, id! : ID \bullet (banned \subseteq \mathrm{dom}(members \cup \{id! \mapsto applicant?\}) \land applicant? \notin \mathrm{ran}\, members \land id! \notin \mathrm{dom}\, members \land members \cup \{id! \mapsto applicant?\} \in ID \rightarrowtail\!\!\!\!\!\rightarrow Person) \tag{7}$$

- By $members : ID \rightarrowtail\!\!\!\!\!\rightarrow Person; id! : ID; applicant? : Person$ and $id! \notin \mathrm{dom}\, members$, we have that $members \cup \{id! \mapsto applicant?\} \in ID \twoheadrightarrow\!\!\!\!\!\!\shortmid\;\, Person$. By $applicant? \notin \mathrm{ran}\, members$, we obtain that $members \cup \{id! \mapsto applicant?\} \in ID \rightarrowtail\!\!\!\!\!\rightarrow Person$. Thus, (7) can be simplified as

$$\exists\, id! : ID \bullet (banned \subseteq \mathrm{dom}(members \cup \{id! \mapsto applicant?\}) \land applicant? \notin \mathrm{ran}\, members \land id! \notin \mathrm{dom}\, members) \tag{8}$$

Simplification of precondition

$$\exists\, id! : ID \bullet (banned \subseteq \mathrm{dom}(members \cup \{id! \mapsto applicant?\}) \land applicant? \notin \mathrm{ran}\, members \land id! \notin \mathrm{dom}\, members) \tag{9}$$

- By properties $\mathrm{dom}(A \cup B) = \mathrm{dom}\, A \cup \mathrm{dom}\, B$ and $\mathrm{dom}\{id! \mapsto applicant?\} = \{id!\}$, we conclude that $\mathrm{dom}(members \cup \{id! \mapsto applicant?\} = (\mathrm{dom}\, members) \cup \{id!\}$. Thus, (9) can be simplified as

$$\exists\, id! : ID \bullet (banned \subseteq (\mathrm{dom}\, members) \cup \{id!\} \land applicant? \notin \mathrm{ran}\, members \land id! \notin \mathrm{dom}\, members) \tag{10}$$

- By the definition of FID we know that $banned \subseteq \mathrm{dom}\, members$. Thus, (10) can be simplified as

$$\exists\, id! : ID \bullet (applicant? \notin \mathrm{ran}\, members \land id! \notin \mathrm{dom}\, members) \tag{11}$$

$$\equiv \quad applicant? \notin \mathrm{ran}\, members \land \exists\, id! : ID \bullet id! \notin \mathrm{dom}\, members \tag{12}$$

$$\equiv \quad applicant? \notin \mathrm{ran}\, members \land \mathrm{dom}\, members \neq ID \tag{13}$$

Simplification of precondition

$$
\begin{array}{|l}
\hline
\textit{PreAddMember} \\
\textit{FID} \\
applicant? : Person \\
\hline
applicant? \notin \operatorname{ran} members \\
\operatorname{dom} members \neq ID \\
\hline
\end{array}
$$

Simplified precondition schema PreAddMember

Properties of the specification

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{BanMember}} \\
\hline
\Delta FID \\
ban? : ID \\
\hline
ban? \in \mathrm{dom}\, members \\
banned' = banned \cup \{ban?\} \\
members' = members \\
\hline
\end{array}
$$

- Property to be verified: To execute the operation BanMember on some banned member, the state of the system will not changed.
- Such a property can be stated as follows:

$$BanMember \mid ban? \in banned \models \Xi FID$$

Properties of the specification

- Be definition, the above statement is equivalent to the following one:

$$
\begin{aligned}
&\Delta FID; ban? : ID \mid (ban? \in \mathrm{dom}\, members \land \\
&\qquad banned' = banned \cup \{ban?\} \land members' = members \land \\
&\qquad ban? \in banned) \\
&\models \\
&\qquad \Delta FID \mid members' = members \land banned' = banned
\end{aligned}
$$

- From $ban? \in banned$ and $banned' = banned \cup \{ban?\}$, we know $banned' = banned$, which completes the proof.

Exercises

$$
\begin{array}{|l}
\hline
\textsf{SM} \\
dir : B \pfun U \\
free : \mathbb{P}\ B \\
\hline
free = B \setminus (\mathrm{dom}\ dir) \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
\textsf{InitSM} \\
SM' \\
\hline
dir' = \{\} \\
free' = B \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
\textsf{Release}_0 \\
\Delta\ SM \\
u? : U \\
b? : B \\
r! : Report \\
\hline
(b? \mapsto u?) \in dir \\
free' = free \cup \{b?\} \\
dir' = \{b?\} \ndres dir \\
r! = \text{“Okay”} \\
\hline
\end{array}
$$

Ex. 1

What is the initialization theorem of the above specification? Write it down, and prove it.

Ex. 2

What is the schema of precondition of Release_0? Write it down, and simplify it.