Formal Methods（形式化方法）

Lecture 12. Formal Specification Examples（形式规格说明实例）

智能与计算学部 章衡

2021年上学期

Outline

Example 1: Vending machine (自动售货机)

Example (Types)

$$[Good, Report]$$

$$Report ::= \text{“Okay”} \mid \cdots$$

Example 1: Vending machine (自动售货机)

Example (Types)

$$[Good, Report]$$

$$Report ::= \text{“Okay”} \mid \cdots$$

Example (State space schema)

VendingMachine

$$coin : \mathbb{P}\,\mathbb{N}$$
$$cost : Good \rightarrow \mathbb{N}$$
$$stock : bag\,Good$$
$$float : bag\,\mathbb{N}$$

$$\mathrm{dom}\ stock \subseteq \mathrm{dom}\ cost$$
$$\mathrm{dom}\ float \subseteq coin$$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Initialization)

InitVendingMachine

$VendingMachine'$

$$coin' = \{\}$$
$$cost' = \{\}$$
$$stock' = [\![\,]\!]$$
$$float' = [\![\,]\!]$$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Price)

Price

$\Delta VendingMachine$
$item? : Good$
$price? : \mathbb{N}$

$coin' = coin$
$cost' = cost \oplus \{item? \mapsto price?\}$
$stock' = stock$
$float' = float$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Price)

Price

ΔVendingMachine
$item? : Good$
$price? : \mathbb{N}$

$coin' = coin$
$cost' = cost \oplus \{item? \mapsto price?\}$
$stock' = stock$
$float' = float$

Success

$rep! : Report$

$rep! = \text{“Okay”}$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Price)

$$DoPrice \mathrel{\widehat{=}} Price \land Success$$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Acceptable Coins)

Accept

ΔVendingMachine
$c? : \mathbb{N}$

$c? \notin coin$
$coin' = coin \cup \{c?\}$
$cost' = cost$
$stock' = stock$
$float' = float$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Acceptable Coins)

AlreadyAcceptable

Ξ VendingMachine
$c? : \mathbb{N}$
$rep! : Report$

$c? \in coin$
$rep! = $ "Coin already acceptable"

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Acceptable Coins)

AlreadyAcceptable

Ξ VendingMachine
$c? : \mathbb{N}$
$rep! : Report$

$c? \in coin$
$rep! = \text{“Coin already acceptable”}$

$$DoAccept \mathrel{\widehat{=}} (Accept \land Success) \lor AlreadyAcceptable$$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Restock)

Restock

ΔVendingMachine
$new? : \text{bag } Good$

$\text{dom } new? \subseteq \text{dom } cost$
$stock' = stock \uplus new?$
$coin' = coin$
$cost' = cost$
$float' = float$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Restock)

GoodsNotPriced

Ξ VendingMachine
$new? : \text{bag } Good$
$rep! : Report$

$\neg(\text{dom } new? \subseteq \text{dom } cost)$
$rep! = \text{"Some goods are unpriced"}$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Restock)

$$
\begin{array}{|l}
\textbf{GoodsNotPriced} \\
\hline
\Xi\,\mathit{VendingMachine} \\
\mathit{new?} : \mathrm{bag}\ \mathit{Good} \\
\mathit{rep!} : \mathit{Report} \\
\hline
\neg(\mathrm{dom}\ \mathit{new?} \subseteq \mathrm{dom}\ \mathit{cost}) \\
\mathit{rep!} = \text{“Some goods are unpriced”} \\
\hline
\end{array}
$$

$$
\mathit{DoRestock} \mathrel{\widehat=} (\mathit{Restock} \land \mathit{Success}) \lor \mathit{GoodsNotPriced}
$$

Example 1: Vending machine (自动售货机)

$$\mathrm{sum} : \mathrm{bag}\,\mathbb{N} \mathrel{+\!\!\!\rightarrow} \mathbb{N}$$

$$\mathrm{sum}[\![\,]\!] = 0$$
$$\forall\, i, j : \mathbb{N}; L : \mathrm{bag}\,\mathbb{N} \mid (\mathrm{dom}\, L) \in \mathbb{F}\,\mathbb{N} \bullet$$
$$\mathrm{sum}(\{i \mapsto j\} \cup L) = i * j + \mathrm{sum}\, L$$

Example

$$\mathrm{sum}\{2 \mapsto 8, 5 \mapsto 4\}$$

Example 1: Vending machine (自动售货机)

$$\mathrm{sum} : \mathrm{bag}\,\mathbb{N} \nrightarrow \mathbb{N}$$

$$\mathrm{sum}[\![\,]\!] = 0$$
$$\forall\, i, j : \mathbb{N}; L : \mathrm{bag}\,\mathbb{N} \mid (\mathrm{dom}\, L) \in \mathbb{F}\,\mathbb{N} \bullet$$
$$\quad \mathrm{sum}(\{i \mapsto j\} \cup L) = i * j + \mathrm{sum}\, L$$

Example

$$\mathrm{sum}\{2 \mapsto 8, 5 \mapsto 4\} = 2 * 8 +$$

Example 1: Vending machine (自动售货机)

$$sum : \mathrm{bag}\,\mathbb{N} \nrightarrow \mathbb{N}$$

$$sum[\![]\!] = 0$$
$$\forall\, i, j : \mathbb{N}; L : \mathrm{bag}\,\mathbb{N} \mid (\mathrm{dom}\, L) \in \mathbb{F}\,\mathbb{N} \bullet$$
$$sum(\{i \mapsto j\} \cup L) = i * j + sum\, L$$

Example

$$sum\{2 \mapsto 8, 5 \mapsto 4\} = 2 * 8 + 5 * 4$$

Example 1: Vending machine (自动售货机)

$$\begin{array}{|l}
\mathrm{sum} : \mathrm{bag}\,\mathbb{N} \nrightarrow \mathbb{N} \\
\hline
\mathrm{sum}[\![\,]\!] = 0 \\
\forall\, i, j : \mathbb{N}; L : \mathrm{bag}\,\mathbb{N} \mid (\mathrm{dom}\, L) \in \mathbb{F}\,\mathbb{N} \bullet \\
\qquad \mathrm{sum}(\{i \mapsto j\} \cup L) = i * j + \mathrm{sum}\, L
\end{array}$$

Example

$$\mathrm{sum}\{2 \mapsto 8, 5 \mapsto 4\} = 2 * 8 + 5 * 4 = 36$$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Buy)

Buy

$$\Delta VendingMachine$$
$$item? : Good$$
$$in?, out! : \mathrm{bag}\ \mathbb{N}$$

$$item? \in \mathrm{dom}\ stock$$
$$sum\,(in?) \geqslant cost\,(item?)$$
$$out! \sqsubseteq float$$
$$\mathrm{dom}\,(in?) \subseteq coin$$
$$sum\,(in?) = sum\,(out!) + cost\,(item?)$$
$$stock' \uplus \{item? \mapsto 1\} = stock$$
$$float' \uplus out! = float \uplus in?$$
$$coin' = coin$$
$$cost' = cost$$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Buy)

$$
\begin{array}{|l}
\textbf{Notinstock} \\
\hline
\Xi\, VendingMachine \\
item? : Good \\
rep! : Report \\
\hline
item? \notin \operatorname{dom} stock \\
rep! = \text{“Item not in stock”} \\
\hline
\end{array}
$$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Buy)

TooLittleMoney

Ξ VendingMachine
item? : Good
in? : bag $\mathbb{N}$
rep! : Report

---

sum (in?) < cost (item?)
rep! = "Insert more money"

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Buy)

ExactChangeUnavailable

Ξ VendingMachine
in? : bag $\mathbb{N}$
item? : Good
rep! : Report

$\neg \exists L : \text{bag } \mathbb{N} \bullet (L \sqsubseteq \text{float} \wedge \text{sum}(in?) = \text{sum}(L) + \text{cost}(item?))$
rep! = "Correct change unavaiable"

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Buy)

ForeignCoin

Ξ VendingMachine
$in? : \text{bag } \mathbb{N}$
$rep! : Report$

$\neg(\text{dom } in? \subseteq coin)$
$rep! = \text{“Unaccepable coin”}$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Buy)

$$DoBuy \mathrel{\widehat{=}} (Buy \land Success) \lor NotInstock \lor ToolittleMoney$$
$$\lor\ ExactChangeUnavailable \lor ForeignCoin$$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Remove money)

RemoveMoney

$\Delta VendingMachine$

$profit? : \mathrm{bag}\ \mathbb{N}$

$float' \uplus profit? = float$

$coin' = coin$

$cost' = cost$

$stock' = stock$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Remove money)

RemoveMoney

ΔVendingMachine
$profit? : \text{bag } \mathbb{N}$

$float' \uplus profit? = float$
$coin' = coin$
$cost' = cost$
$stock' = stock$

Profittering

Ξ VendingMachine
$profit? : \text{bag } \mathbb{N}$
$rep! : Report$

$\neg profit? \sqsubseteq float$
$rep! =$ "Such profit non-existent"

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Remove money)

$$\mathrm{DoRemoveMoney} \mathrel{\widehat=} (\mathrm{RemoveMoney} \land \mathrm{Success}) \lor \mathrm{Profiteering}$$

Outline

Example 2: Storage management (内存管理系统)

Example (Type and axiomatic definition)

- Basic type :

  [U]: the set of all possible users

Example 2: Storage management (内存管理系统)

Example (Type and axiomatic definition)

- Basic type:

  [U]: the set of all possible users

  [Report]: the set of all messages

Example 2: Storage management (内存管理系统)

Example (Type and axiomatic definition)

- Basic type:

  [U]: the set of all possible users

  [Report]: the set of all messages

- Axiomatic definition of block:

$$
\begin{array}{|l}
n : \mathbb{N} \\
B : \mathbb{P}\ \mathbb{N} \\
\hline
B = 1..n
\end{array}
$$

Example 2: Storage management

Example (State space schema)

SM

$\mathrm{dir} : \mathrm{B} \mathbin{+\!\!\!\!\rightarrow} \mathrm{U}$

$\mathrm{free} : \mathbb{P}\ \mathrm{B}$

$\mathrm{free} = \mathrm{B} \setminus (\mathrm{dom}\ \mathrm{dir})$

Example 2: Storage management

Example (State space schema)

SM

$dir : B \pfun U$

$free : \mathbb{P}\ B$

$free = B \setminus (\mathrm{dom}\ dir)$

$$\begin{aligned}
\Delta SM &\ \widehat{=}\ SM \land SM' \\
\Xi SM &\ \widehat{=}\ \Delta SM \mid dir' = dir \land free' = free
\end{aligned}$$

Example 2: Storage management

Example (State space schema)

SM

$$\mathrm{dir} : B \pfun U$$
$$\mathrm{free} : \mathbb{P}\, B$$

$$\mathrm{free} = B \setminus (\mathrm{dom}\ \mathrm{dir})$$

$$\Delta SM \quad \widehat{=} \quad SM \land SM'$$
$$\Xi SM \quad \widehat{=} \quad \Delta SM \mid \mathrm{dir}' = \mathrm{dir} \land \mathrm{free}' = \mathrm{free}$$

ΔSM

$$\mathrm{dir}, \mathrm{dir}' : B \pfun U$$
$$\mathrm{free}, \mathrm{free}' : \mathbb{P}\, B$$

$$\mathrm{free} = B \setminus (\mathrm{dom}\,\mathrm{dir})$$
$$\mathrm{free}' = B \setminus (\mathrm{dom}\,\mathrm{dir}')$$

Example 2: Storage management

Example (Operational schemas: Initilization)

InitSM

SM'

$dir' = \{\}$
$free' = B$

Example 2: Storage management

Example (Operational schemas: Request memory)

$$\text{Report} ::== \text{“Okay”} \mid \text{“Fail”} \mid \text{“BlockFree”} \mid \text{“NotOwner”}$$

Example 2: Storage management

Example (Operational schemas: Request memory)

$$\text{Report} ::== \text{“Okay”} \mid \text{“Fail”} \mid \text{“BlockFree”} \mid \text{“NotOwner”}$$

Request_0

ΔSM
$u? : U$
$b! : B$
$r! : \text{Report}$

$\text{free} \neq \{\}$
$b! \in \text{free}$
$\text{free}' = \text{free} \setminus \{b!\}$
$\text{dir}' = \text{dir} \cup \{b! \mapsto u?\}$
$r! = \text{“Okay”}$

Precondition

- Precondition: a condition or predicate that must always be true before an operation

Precondition

- Precondition: a condition or predicate that must always be true before an operation

Definition (Precondition)

OP

$x_1, x_1' : T_1; \ldots; x_n, x_n' : T_n$

$y_1! : S_1; \ldots; y_m! : S_m$

declarations

φ

Precondition

- Precondition: a condition or predicate that must always be true before an operation

Definition (Precondition)

OP

$x_1, x_1' : T_1; \ldots ; x_n, x_n' : T_n$

$y_1! : S_1; \ldots ; y_m! : S_m$

declarations

φ

preOP

$x_1 : T_1; \ldots ; x_n : T_n$

declarations

$\exists x_1' : T_1; \ldots ; x_n' : T_n; y_1! : S_1; \ldots ; y_m! : S_m \bullet \varphi$

Example 2: Storage management

Example (Precondition of Request_0)

$$
\begin{array}{|l}
\hline
\text{PreRequest}_0 \\
\text{SM} \\
u? : U \\
\hline
\exists\, SM'; b! : B; r! : Report \bullet \\
\quad (free \neq \{\} \wedge \\
\quad\ b! \in free \wedge \\
\quad\ free' = free \setminus \{b!\} \wedge \\
\quad\ dir' = dir \cup \{b! \mapsto u?\} \wedge \\
\quad\ r! = \text{“Okay”} \wedge \\
\quad\ free' = B \setminus (\mathrm{dom}\ dir')) \\
\hline
\end{array}
$$

Example 2: Storage management

Example (Precondition of $\mathrm{Request}_0$)

$$
\begin{array}{|l}
\hline
\mathrm{PreRequest}_0 \\
\mathrm{SM} \\
u? : U \\
\hline
\exists\ \mathrm{SM}'; b! : B; r! : \mathrm{Report} \bullet \\
\quad (\mathrm{free} \neq \{\} \land \\
\quad\ b! \in \mathrm{free} \land \\
\quad\ \mathrm{free}' = \mathrm{free} \setminus \{b!\} \land \\
\quad\ \mathrm{dir}' = \mathrm{dir} \cup \{b! \mapsto u?\} \land \\
\quad\ r! = \text{“Okay”} \land \\
\quad\ \mathrm{free}' = B \setminus (\mathrm{dom}\ \mathrm{dir}')) \\
\hline
\end{array}
$$

where $\exists\,\mathrm{SM}'$ denotes $\exists\,\mathrm{dir}' : B \rightarrow\!\!\!\!\!+ U; \mathrm{free}' : \mathbb{P}\,B$

Example 2: Storage management

Example (Precondition of $Request_0$)

$$
\begin{array}{|l}
\hline
\textbf{PreRequest}_0 \\
SM \\
u? : U \\
\hline
\exists\, SM'; b! : B; r! : Report \bullet \\
\qquad (free \neq \{\} \wedge \\
\qquad\; b! \in free \wedge \\
\qquad\; free' = free \setminus \{b!\} \wedge \\
\qquad\; dir' = dir \cup \{b! \mapsto u?\} \wedge \\
\qquad\; r! = \text{“Okay”} \wedge \\
\qquad\; free' = B \setminus (\mathrm{dom}\ dir')) \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
\textbf{PreRequest}_0 \\
SM \\
u? : U \\
\hline
free \neq \{\} \\
\hline
\end{array}
$$

where $\exists\, SM'$ denotes $\exists\, dir' : B \mathrel{+\!\!\!\!\to} U; free' : \mathbb{P}\, B$

Example 2: Storage management

Example (Operational schemas: Request memory)

$$
\begin{array}{|l}
\hline
\textit{Request}_0\textit{Err} \\
\Xi\, SM \\
r! : Report \\
\hline
free = \{\} \\
r! = \text{“Fail”} \\
\hline
\end{array}
$$

Example 2: Storage management

Example (Operational schemas: Request memory)

$$
\begin{array}{|l}
\hline
\textbf{Request}_0\textbf{Err} \\
\Xi\, SM \\
r! : Report \\
\hline
free = \{\} \\
r! = \text{“Fail”} \\
\hline
\end{array}
$$

$$Request \mathrel{\widehat{=}} Request_0 \lor Request_0Err$$

Example 2: Storage management

Example (Operational schemas: Release memory)

$\mathrm{Release}_0$

ΔSM
$u? : U$
$b? : B$
$r! : Report$

$(b? \mapsto u?) \in dir$
$free' = free \cup \{b?\}$
$dir' = \{b?\} \ntriangleleft dir$
$r! = \text{“Okay”}$

Example 2: Storage management

Example (Precondition of $\mathrm{Release}_0$)

$$
\begin{array}{|l}
\hline
\textbf{PreRelease}_0 \\
SM \\
u? : U \\
b? : B \\
\hline
\exists\, SM'; r! : Report \bullet \\
\qquad ((b? \mapsto u?) \in dir \wedge \\
\qquad\ free' = free \cup \{b?\} \wedge \\
\qquad\ dir' = \{b?\} \lhd dir \wedge \\
\qquad\ r! = \text{“Okay”}) \\
\hline
\end{array}
$$

Example 2: Storage management

Example (Precondition of $Release_0$)

$$
\begin{array}{|l}
\hline
\textbf{PreRelease}_0 \\
SM \\
u? : U \\
b? : B \\
\hline
\exists\ SM'; r! : Report \bullet \\
\qquad ((b? \mapsto u?) \in dir \land \\
\qquad\ free' = free \cup \{b?\} \land \\
\qquad\ dir' = \{b?\} \lhd dir \land \\
\qquad\ r! = \text{“Okay”}) \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
\textbf{PreRelease}_0 \\
SM \\
u? : U \\
b? : B \\
\hline
(b? \mapsto u?) \in dir \\
\hline
\end{array}
$$

Example 2: Storage management

Example (Operational schemas: Release memory)

RelFreeErr

$\Xi\, SM$

$u? : U$

$b? : B$

$r! : Report$

$b? \in free$

$r! = \text{“BlockFree”}$

Example 2: Storage management

Example (Operational schemas: Release memory)

RelFreeErr

$\Xi\, SM$

$u? : U$

$b? : B$

$r! : Report$

$b? \in free$

$r! = \text{“BlockFree”}$

PreRelFreeErr

SM

$u? : U$

$b? : B$

$b? \in free$

Example 2: Storage management

Example (Operational schemas: Release memory)

RelOwnerErr

$\Xi\, SM$

$u? : U$

$b? : B$

$r! : Report$

$b? \in \mathrm{dom}\ dir$

$dir\, b? \neq u?$

$r! = \text{“NotOwner”}$

Example 2: Storage management

Example (Operational schemas: Release memory)

RelOwnerErr

$\Xi\, SM$
$u? : U$
$b? : B$
$r! : Report$

$b? \in \mathrm{dom}\ dir$
$dir\, b? \neq u?$
$r! = \text{“NotOwner”}$

PreRelOwnerErr

SM
$u? : U$
$b? : B$

$b? \in \mathrm{dom}\ dir$
$dir\, b? \neq u?$

Example 2: Storage management

Example (Operational schemas: Release memory)

$$Release \mathrel{\widehat{=}} Release_0 \lor RelFreeErr \lor RelOwnerErr$$

Example 2: Storage management

Example (Request a set of blocks)

$$
\begin{array}{|l}
\textbf{ReqStore}_0 \\
\hline
\Delta SM \\
u? : U \\
n? : \mathbb{N} \\
b! : \mathbb{P}\, B \\
r! : Report \\
\hline
n? \in 1..\#B \\
\#b! = n? \\
\exists a, b : \mathbb{N} \bullet b! = a..b \\
dir' = dir \cup (b! \times \{u?\}) \\
free' = free \setminus b! \\
r! = \text{“Okay”} \\
\hline
\end{array}
$$

Example 2: Storage management

Example (Operational schemas: First-fit allocation)

FirstFit

SM

$n? : \mathbb{N}$

$b! : \mathbb{P}\ B$

$\exists\ S : \mathbb{P}\ B \bullet$

$(S = \{l, h : B \mid l..h \subseteq free \land l - 1 \notin free \land h - l + 1 \geqslant n? \bullet l\} \land$

$b! = (\min S)..(\min S) + n? - 1)$

Example 2: Storage management

Example (Operational schemas: First-fit allocation)

$$
\begin{array}{l}
\textbf{FirstFit} \\
\hline
SM \\
n? : \mathbb{N} \\
b! : \mathbb{P}\ B \\
\hline
\exists\ S : \mathbb{P}\ B \bullet \\
\quad (S = \{l, h : B \mid l..h \subseteq free \land l - 1 \notin free \land h - l + 1 \geqslant n? \bullet l\} \land \\
\quad\ b! = (\min S)..(\min S) + n? - 1)
\end{array}
$$

$$ReqStoreFF_0 \mathrel{\widehat=} ReqStore_0 \land FirstFit$$

Exercise

1. 假设在给用户分配存储块时增加限制：每个用户允许至多使用10个存储块。请修改操作模式request来实现这一需求变更（包括返回错误信息）。
   (English) Suppose we have a restriction in memory allocation: Each user can use at most 10 blocks. Please revise the operational schema request_0 to describe the new operation.
2. 称B中的一个连续空闲存储块S是极大的，若无法对S扩展得到一个更大的连续空闲存储块。请设计一个模式来描述存储块分配的最佳适应法，也就是说，若用户申请分配一个大小为n的连续存储块，存储管理系统将在B中寻找一个满足要求的最小的极大连续空闲存储块S，并将S中的前面n个连续存储块分配给该用户。