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

1 Example 1: Vending Machine

2 Example 2: Storage Management

Outline

1 Example 1: Vending Machine

2 Example 2: Storage Management

Example 1: Vending machine (自动售货机)

Example (Types)

$$[\mathrm{Good}, \mathrm{Report}]$$

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

Example (State space schema)

VendingMachine

$$\begin{aligned} &\mathrm{coin} : \mathbb{P}\,\mathbb{N} \\ &\mathrm{cost} : \mathrm{Good} \to \mathbb{N} \\ &\mathrm{stock} : \mathrm{bag}\,\mathrm{Good} \\ &\mathrm{float} : \mathrm{bag}\,\mathbb{N} \\ \hline &\mathrm{dom}\ \mathrm{stock} \subseteq \mathrm{dom}\ \mathrm{cost} \\ &\mathrm{dom}\ \mathrm{float} \subseteq \mathrm{coin} \end{aligned}$$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Initialization)

InitVendingMachine

$$\text{VendingMachine}'$$

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

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Price)

Price

ΔVendingMachine

item? : Good

price? : $\mathbb{N}$

$\text{coin}' = \text{coin}$

$\text{cost}' = \text{cost} \oplus \{\text{item?} \mapsto \text{price?}\}$

$\text{stock}' = \text{stock}$

$\text{float}' = \text{float}$

Success

rep! : Report

---

rep! = “Okay”

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Price)

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

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Acceptable Coins)

Accept

$\Delta \text{VendingMachine}$

$c? : \mathbb{N}$

$c? \notin \text{coin}$

$\text{coin}' = \text{coin} \cup \{c?\}$

$\text{cost}' = \text{cost}$

$\text{stock}' = \text{stock}$

$\text{float}' = \text{float}$

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"

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

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Restock)

Restock

ΔVendingMachine

new? : bag Good

---

dom new? $\subseteq$ 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? : bag Good
rep! : Report

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

$$\text{DoRestock} \mathrel{\widehat{=}} (\text{Restock} \wedge \text{Success}) \vee \text{GoodsNotPriced}$$

Example 1: Vending machine (自动售货机)

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

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

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

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

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Buy)

Notinstock

Ξ VendingMachine
item? : Good
rep! : Report

---

item? $\notin$ dom stock
rep! = "Item not in stock"

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

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Buy)

ForeignCoin

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

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

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Buy)

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

$$\lor \text{ExactChangeUnavailable} \lor \text{ForeignCoin}$$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Remove money)

RemoveMoney

$\Delta VendingMachine$

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

$float' \uplus profit? = float$

$coin' = coin$

$cost' = cost$

$stock' = stock$

Profittering

$\Xi VendingMachine$

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

$rep! : Report$

$\neg profit? \sqsubseteq float$

$rep! = \text{“Such profit non-existent”}$

Example 1: Vending machine (自动售货机)

Example (Operational schemas: Remove money)

$$\text{DoRemoveMoney} \mathrel{\widehat{=}} (\text{RemoveMoney} \wedge \text{Success}) \vee \text{Profiteering}$$

Outline

1 Example 1: Vending Machine

2 Example 2: Storage Management

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

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

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

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

ΔSM

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

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

Example 2: Storage management

Example (Operational schemas: Initilization)

InitSM

SM'

$dir' = \{\}$

$free' = B$

Example 2: Storage management

Example (Operational schemas: Request memory)

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

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

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)

PreRequest_0

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

PreRequest_0

$$
\begin{array}{l}
\text{SM} \\
\text{u?} : \text{U} \\
\hline
\text{free} \neq \{\}
\end{array}
$$

where $\exists\, \text{SM}'$ denotes $\exists\, \text{dir}' : \text{B} \pfun \text{U}; \text{free}' : \mathbb{P}\, \text{B}$

Example 2: Storage management

Example (Operational schemas: Request memory)

$\mathrm{Request}_0\mathrm{Err}$

$\Xi \mathrm{SM}$

$r! : \mathrm{Report}$

$\mathrm{free} = \{\}$

$r! = \text{“Fail”}$

$$\mathrm{Request} \mathrel{\widehat{=}} \mathrm{Request}_0 \lor \mathrm{Request}_0\mathrm{Err}$$

Example 2: Storage management

Example (Operational schemas: Release memory)

Release_0

ΔSM

$u? : U$

$b? : B$

$r! : \text{Report}$

$(b? \mapsto u?) \in \text{dir}$

$\text{free}' = \text{free} \cup \{b?\}$

$\text{dir}' = \{b?\} \lhd \text{dir}$

$r! = \text{“Okay”}$

Example 2: Storage management

Example (Precondition of Release_0)

PreRelease_0

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

PreRelease_0

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

Example 2: Storage management

Example (Operational schemas: Release memory)

RelFreeErr

ΞSM

$u? : U$

$b? : B$

$r! : Report$

$b? \in free$

$r! = \text{“BlockFree”}$

PreRelFreeErr

SM

$u? : U$

$b? : B$

$b? \in free$

Example 2: Storage management

Example (Operational schemas: Release memory)

RelOwnerErr

ΞSM

$u? : U$

$b? : B$

$r! : Report$

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

$dir\, b? \neq u?$

$r! = \text{“NotOwner”}$

PreRelOwnerErr

SM

$u? : U$

$b? : B$

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

$dir\, b? \neq u?$

Example 2: Storage management

Example (Operational schemas: Release memory)

$$\text{Release} \mathrel{\widehat{=}} \text{Release}_0 \vee \text{RelFreeErr} \vee \text{RelOwnerErr}$$

Example 2: Storage management

Example (Request a set of blocks)

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

Example 2: Storage management

Example (Operational schemas: First-fit allocation)

FirstFit

$$SM$$
$$n? : \mathbb{N}$$
$$b! : \mathbb{P}\ B$$

$$\exists\ S : \mathbb{P}\ B \bullet$$
$$(S = \{l, h : B \mid l..h \subseteq free \land l - 1 \notin free \land h - l + 1 \geqslant n? \bullet l\} \land$$
$$b! = (\min S)..(\min S) + n? - 1)$$

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

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。