Lecture 19

Thursday November 24

$$\underbrace{\{x \mid x>0\}}_{1,\,2,\,3,\,\ldots} \subseteq \underbrace{\{x \mid x \geq 0\}}_{0,\,1,\,2,\,\ldots}$$

Does ① require more or less than ②?

① $x > 0$

$$x>0 \Rightarrow x \geq 0$$

② $x \geq 0$

∵ $x=0$ is not allowed by ① but is allowed by ②.

① requires more than ②

② requires less than ①

$$\underbrace{P_1}_{\text{precondition}} \Rightarrow \underbrace{P_2}_{\text{precondition}}$$

P_2 requires less than P_1

$\because$ P_2 allows more input values.

sort (input : ARRAY[INTEGER])

ensure

② ⇒ ①

↳ ② ensures more than ①.

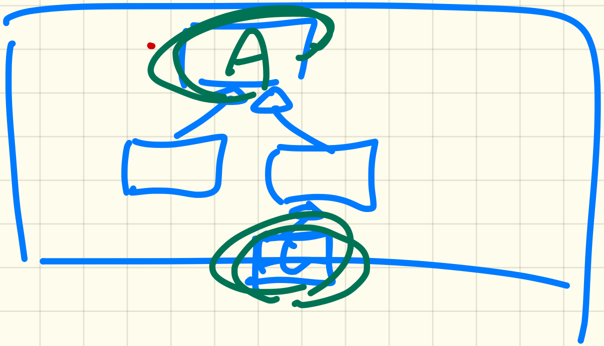
① $\forall i \mid 1 \leq i \leq \text{input.count} - 1 \bullet$

weaker $\boxed{\text{input}[i] \leq \text{input}[i+1]}$

② $\forall i \mid 1 \leq i \leq \text{input.count} - 1 \bullet$

stronger $\boxed{\text{input}[i] < \text{input}[i+1]}$

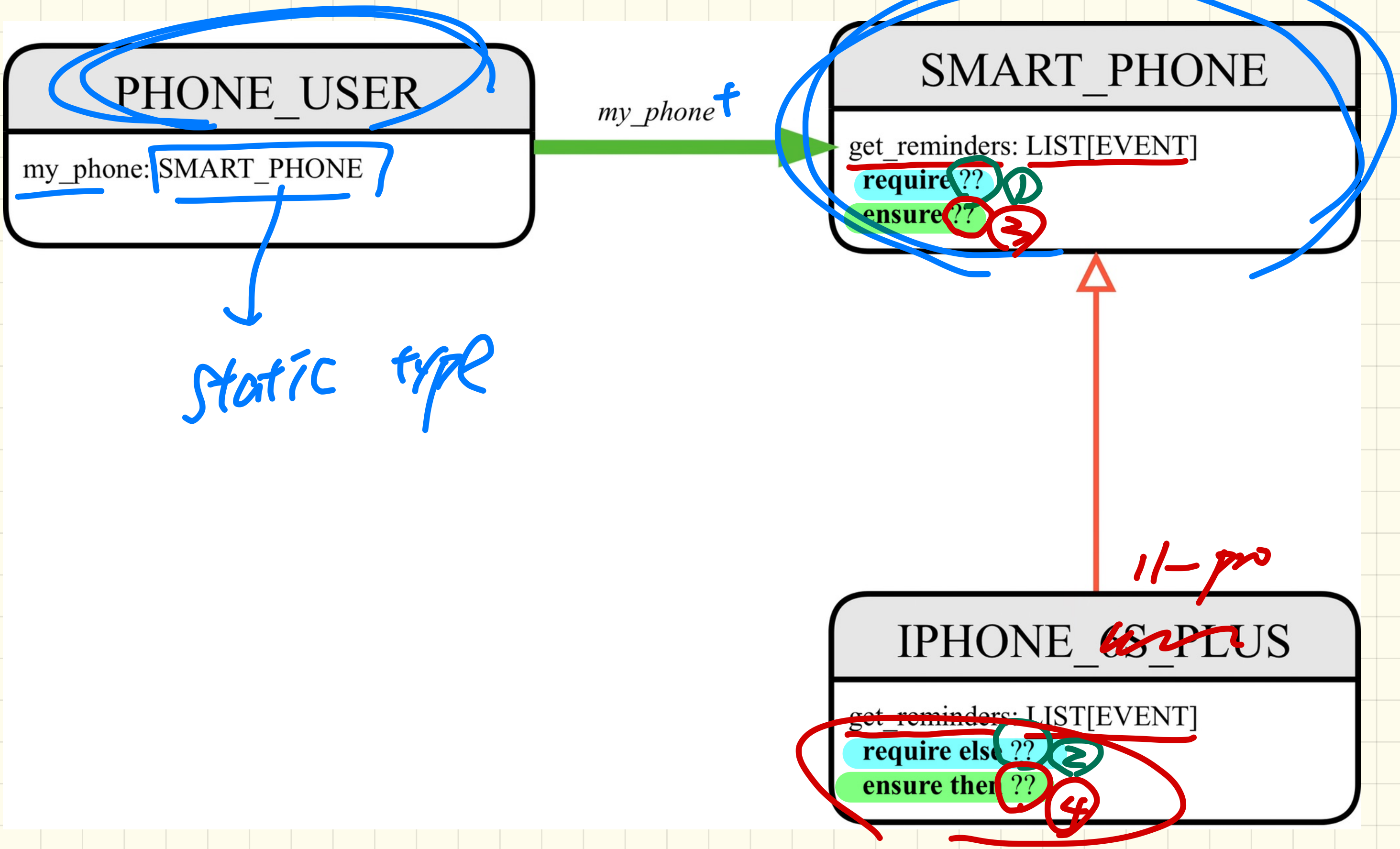
less output values will be able to satisfy.



✓ v1 := v2 ✓

ST of v2 can fulfill all expectations on the ST of v1.

Subcontracting: Architectural View



PHONE_USER

my_phone: SMART_PHONE

my_phone

SMART_PHONE

get_reminders: LIST[EVENT]

require ??

ensure ??

IPHONE_6S_PLUS

get_reminders: LIST[EVENT]

require else ??

ensure then ??

Subcontracting: Example (1)

```
class SMART_PHONE
  get_reminders: LIST[EVENT]
    require
      α: battery_level ≥ 0.1 -- 10%
    ensure
      β: ∀e : Result | e happens today
end
```

0.13 T

13

✓ α requires less than γ

γ ⇒ α

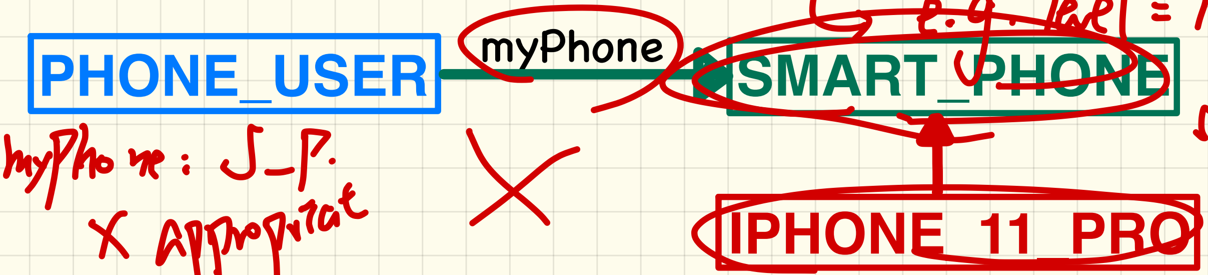
✓

```
class IPHONE_11_PRO
inherit SMART_PHONE redefine get_reminders end
  get_reminders: LIST[EVENT]
    require else
      γ: battery_level ≥ 0.15 -- 15%
    ensure then
      δ: ∀e : Result | e happens today or tomorrow
end
```

0.13 F

γ requires mor than α

e.g. level = 13%



↓ satisfies α but fail γ

myPhone : S_P.

X Appropriate

X

```
class SMART_PHONE
  get_reminders: LIST[EVENT]
    require
      α: battery_level ≥ 0.1 -- 10%
    ensure
      β: ∀e: Result | e happens today
end
```

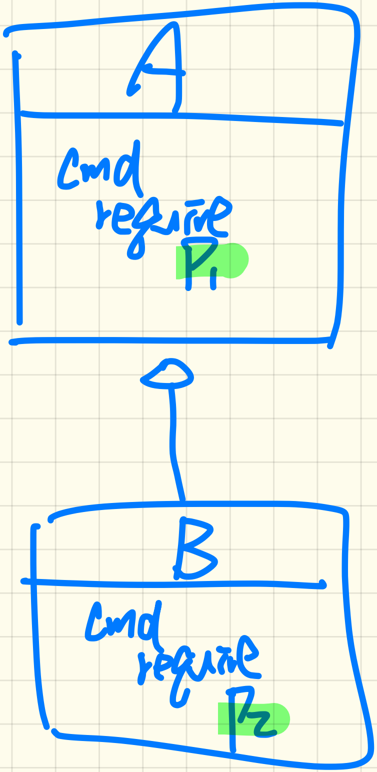
```
class IPHONE_11_PRO
inherit SMART_PHONE redefine get_reminders end
  get_reminders: LIST[EVENT]
    require else
      γ: battery_level ≥ ~~0.15~~ 0.05 -- ~~15%~~ 5%
    ensure then
      δ: ∀e: Result | e happens today or tomorrow
end
```

$\alpha \Rightarrow \underline{\gamma}$

level ≥ 10% ⇒ level ≥ 5%

$\{10, 11, 12, \ldots\} \subseteq \{5, 6, 7,$

Exam



Are the preconditions P_1 and P_2 design appropriately?

① To be appropriate:

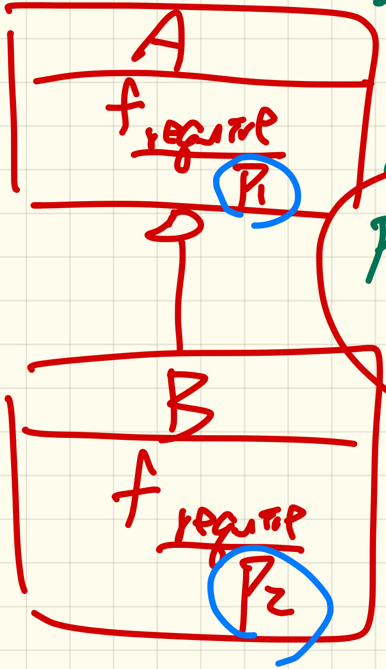
$P_1 \Rightarrow P_2$ (P_2 less strict)

② prove it (e.g. counter example)

P	Q	P ⇒ Q
F	F	T
F	T	T
T	F	F
T	T	T

T

$\neg(P1 \Rightarrow P2)$ means there is a witness 'X' that can mak P1 true but P2 false.



any input values that satisfy P_1 can also satisfy

$\boxed{P_1 \Rightarrow P_2}$ P_2

α : level $\geq 10\%$

10

γ : level $\geq 15\%$

allows less values ?

$$\{x \mid \alpha(x)\} = \{\underline{10\%}, \underline{11\%}, \underline{12}, \underline{13}, \underline{14}, \ldots\}$$

$$\{y \mid \underline{\gamma}(y)\} = \{15\%, 16\%, \ldots\}$$

Subcontracting: Example (2)

```
class SMART_PHONE
  get_reminders: LIST[EVENT]
    require
      α: battery_level ≥ 0.1 -- 10%
    ensure
      β: ∀e:Result | e happens today
end
```

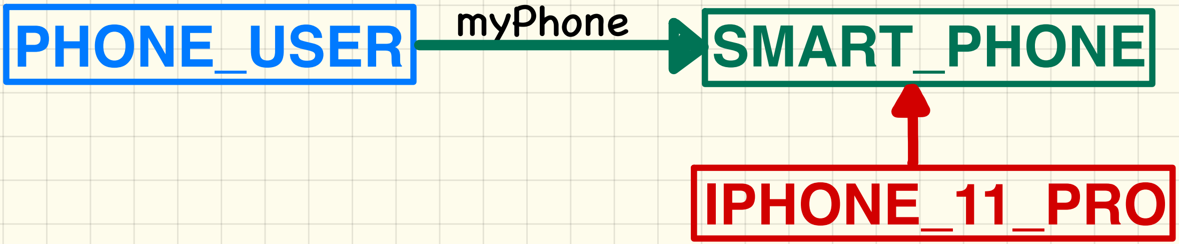
① If β and δ are appropriate, then:

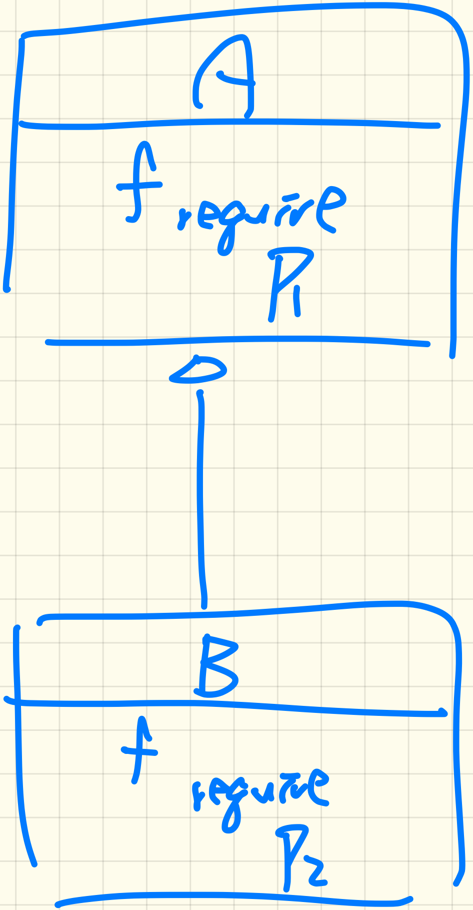
→ δ ⇒ β

→ not the case.

```
class IPHONE_11_PRO
inherit SMART_PHONE redefine get_reminders end
  get_reminders: LIST[EVENT]
    require else
      γ: battery_level ≥ 0.15 -- 15%
    ensure then
      δ: ∀e:Result | e happens today or tomorrow
end
```

Counter example: list of events containing only those tom.





Valid:

$$P_1 \Rightarrow P_2$$

Is it ever possible that our design requires $P_2 \Rightarrow P_1$?

↳ poor design ∵ it breaks substitutability.

$$S_1 \subseteq S_2$$

↳ to disprove it,

find x s.t.

$x \in S_1 \wedge x \notin S_2$.

```
class SMART_PHONE
  get_reminders: LIST[EVENT]
    require
      α: battery_level ≥ 0.1 -- 10%
    ensure
      β: ∀e : Result | e happens today
end
```

P1

not appropriate :(

13%

```
class IPHONE_11_PRO
inherit SMART_PHONE redefine get_reminders end
  get_reminders: LIST[EVENT]
    require else
      γ: battery_level ≥ 0.15 -- 15%
    ensure then
      δ: ∀e : Result | e happens today or tomorrow
end
```

P2 → should be weaker when IF not; IF has no effect

S : SMART_PHONE

Create {IP_11_PRO} S.make

S.get_reminders .

Run time check

level ≥ 10% ✓

level ≥ 15%

T

level $\geq$ 10% || or else level $\geq$ 15%

T

↯

X don't bother

short-circuit ~~effect~~.

13%

and then

p1	p2	p1 && p2	p2 || p1
F	F		
F	T		
T	F		
T	T		

p1 and then ~~p2~~

p1 or else ~~p2~~

T

```
class SMART_PHONE
  get_reminders: LIST[EVENT]
    require
      α: battery_level ≥ 0.1 -- 10%
    ensure
      β: ∀e : Result | e happens today
end
```

```
class IPHONE_11_PRO
inherit SMART_PHONE redefine get_reminders end
  get_reminders: LIST[EVENT]
    require ~~else~~
      γ: battery_level ≥ 0.15 -- 15%
    ensure ~~then~~
      δ: ∀e : Result | e happens today or tomorrow
end
```

won't compile

won't compile.

$\underline{x>1}$? $\underline{2y}$: $\underline{3z}$

↳ $\underline{\text{if}}$ $x>1$ $\underline{\text{then}}$

$2y$

$\underline{\text{else}}$ $3z$. end