# Summary from James' visit

**A: how to reason about external calls and how to reason about protection in the open world,**

Here the call of an external function. Note that I changed the notation for "$y$ is protected from $x$ by module $M$" to be

[ext-call]

$$M \vdash \{\, A \wedge \text{Ext}\ x \wedge\ y ⦚ x \,\}\ x.m()\ \{\ \text{Ret}(A \wedge \text{Inside}(y), HS_M)\ \}$$

In the above, $HS_M$ is the holistic specification of $M$. And we define "$y$ is protected from $x$ by module $M$" as below

$$M, \sigma \vDash y ⦚ x \ \triangleq\ \forall n, f_1, \ldots f_n.\ [\ \sigma(x.f_1\ldots f_n) = y \implies \exists\ k<n.\ \sigma(x.f_1\ldots f_k) \in M\ ]$$

Note that the above definition does not preclude tat the path once it went through M, can go outside again. Here it is possible that $j>k \wedge \sigma(x.f_1\ldots f_j) \notin M$.

And we need some HL rules for the preservation of $y ⦚ x$. For example, something like

[prot-1]

$$M \vdash \{\, x \neq u \wedge\ y ⦚ x \,\}\ u=v\ \{\ y ⦚ x \,\}$$

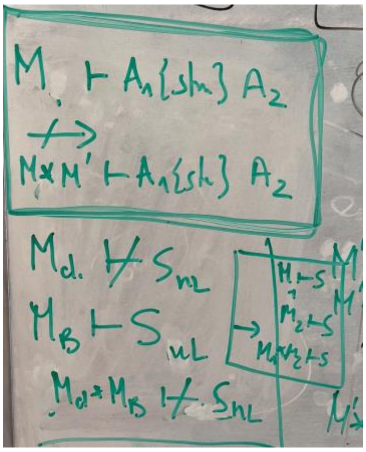
[prot-2]

$$M \vdash \{\, u ⦚ x \wedge\ y ⦚ x \,\}\ u=v.f\ \{\ y ⦚ x \,\}$$

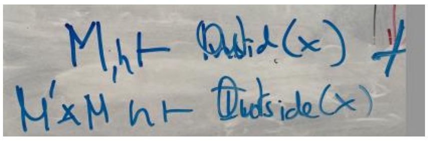
## B: Obtaining holistic specs of more than one module

**Lack of monotonicity**

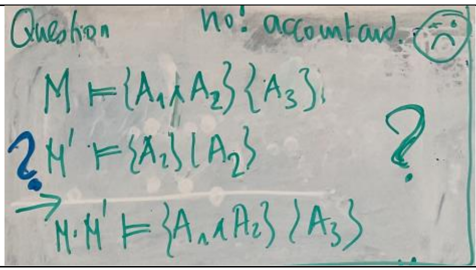
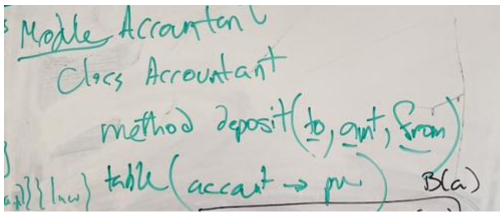
Here some implications which do not hold in general (btw, check in how far they would hold in the closed world)



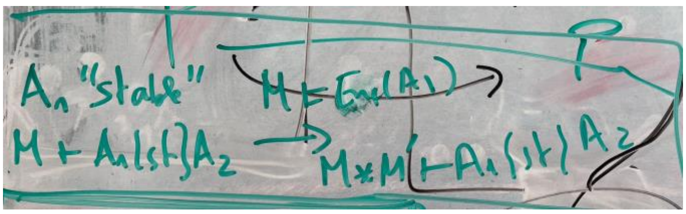
For example:




And similarly, the following implication does not hold, eg “Accountant” gives a counter-example

| | |
|---|---|
|   | Here the code of accountant   |

However, some assertions are “stable”, and then, more implications hold

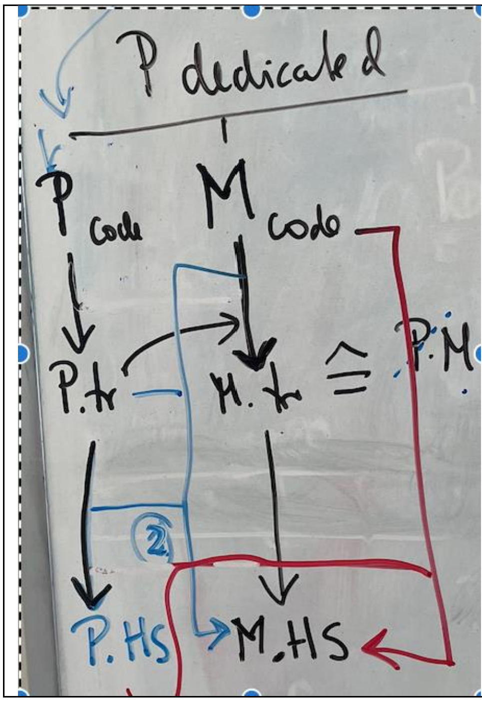


**How do we combine modules into larger ones?**

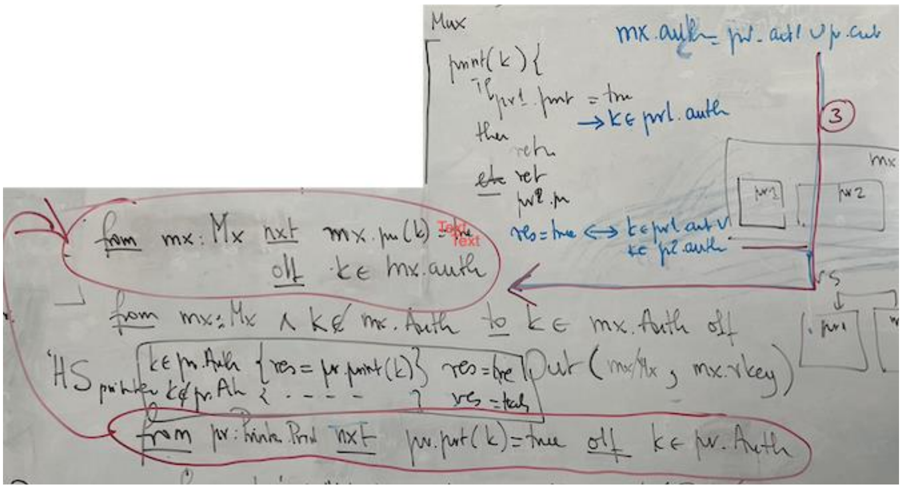
When we combine modules, we should distinguish between

1) M1[M2]  M1 encapsulates M2, and M2 is not visible outside M1
2) M1 || M2  M1 and M2 are not aware of each other,
   and are both visible to outside
3) M1[M2] || M2
   M1 uses M2, and they are both visible to outside
4) M1[M2] || M2[M1]
   M1 and M2 use each ither, and are both visible to outside

**Three avenues to obtain holistic specs from several modules**

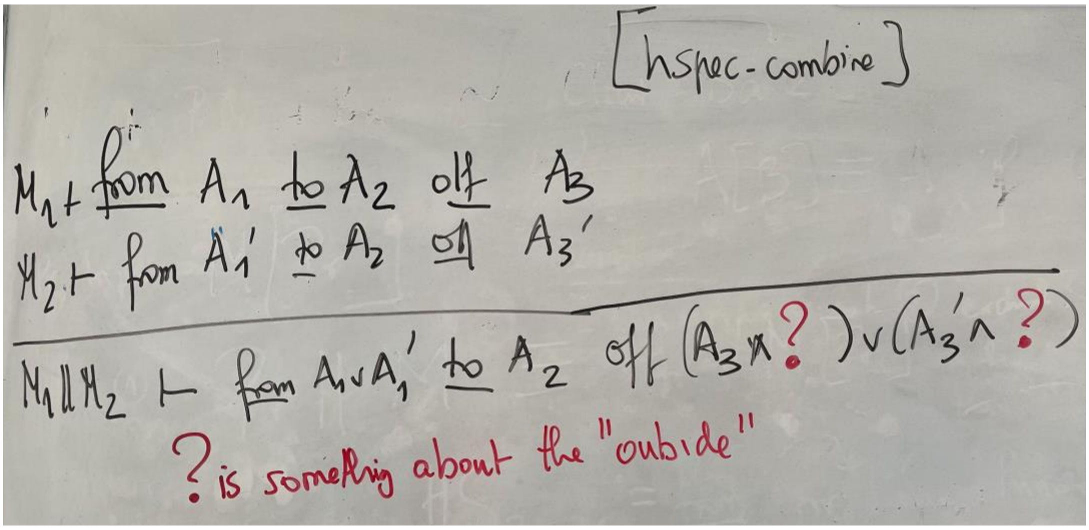
| | |
|---|---|
|  | P is dedicated, ie we have M[P]<br>Q::= P \| M<br><br>Q.tr : classical triples for Q<br>Q.HS : Holistic Spec Q<br><br>Black arrows:<br>as we did in OOPSLA'23<br><br>Blue Arrow (new)<br>Combine Holistic Specs<br><br>Red Arrow (new)<br>Combine Code and HS |

An Example of the "red avenue" from above. Here, printers keep authorization tokens, and only print if the call print(k) passes a k which is one of the authorization tokens. In that case the call print)k) returns true.

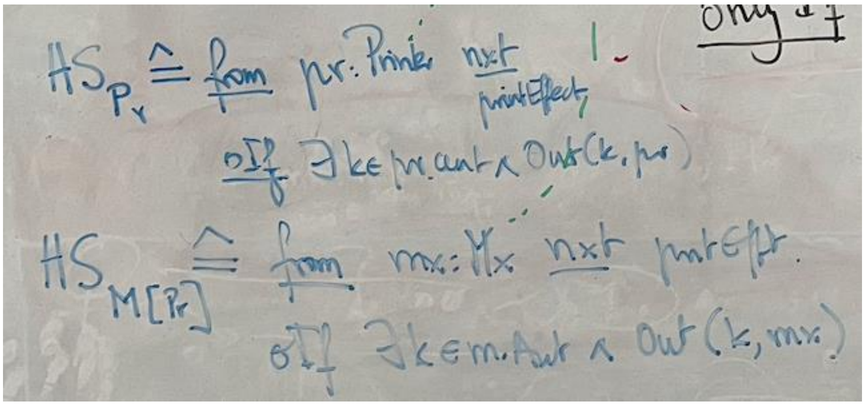



An Example of the “Blue Avenue” from above

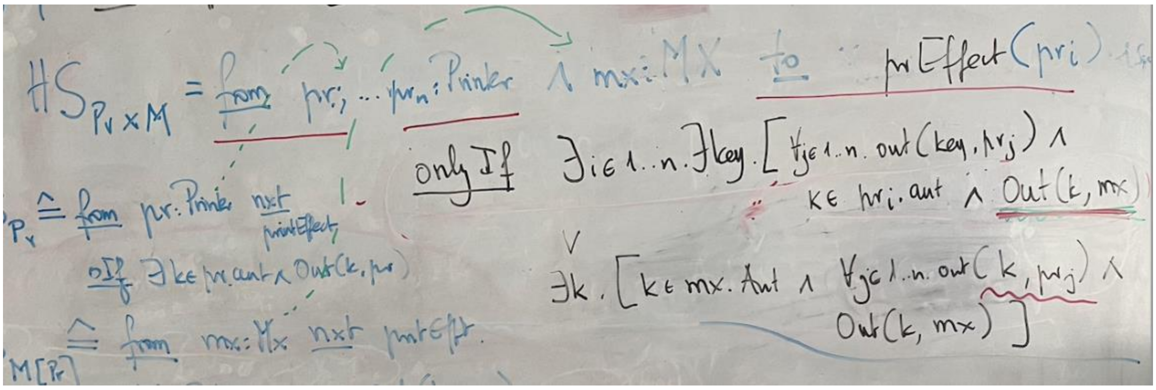
Here the general rule



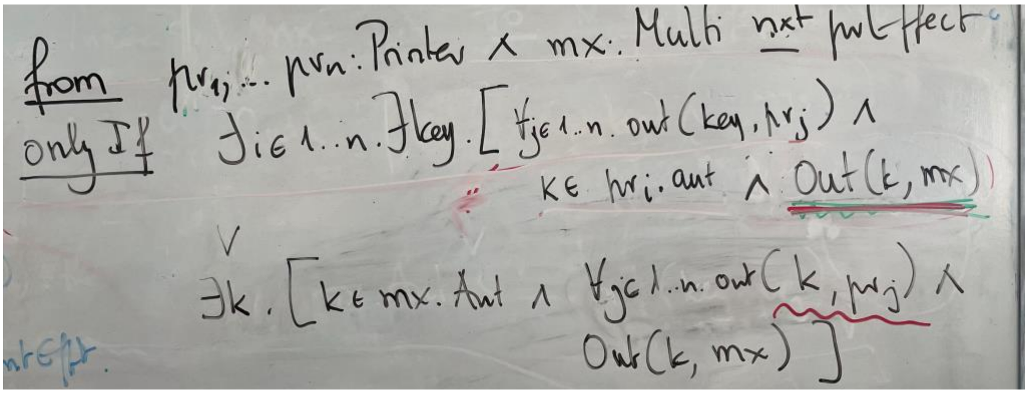

Here its application



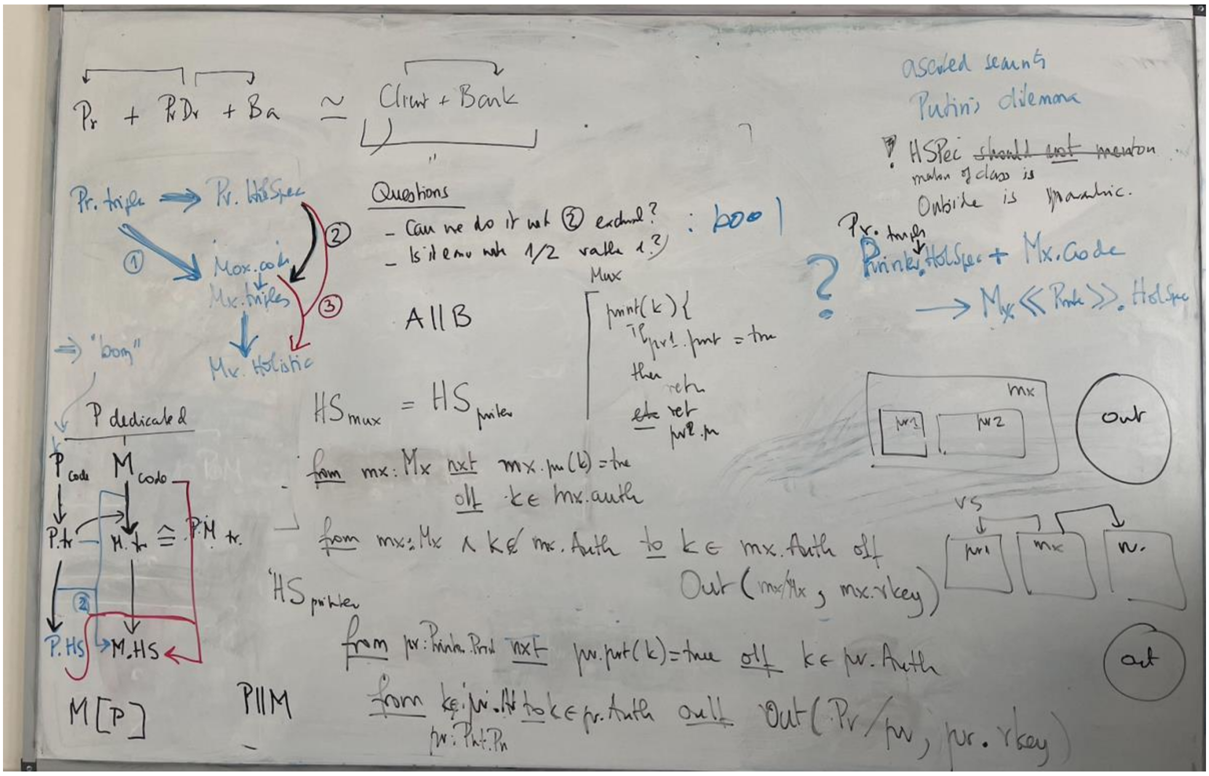
The combined holistic spec should be




Or perhaps should be




**Example: Printer, Printer Multiplexer and Bank**




**Things that I have not yet fully “deciphered”**

I think that G stands for obeys, P for pays, but B?