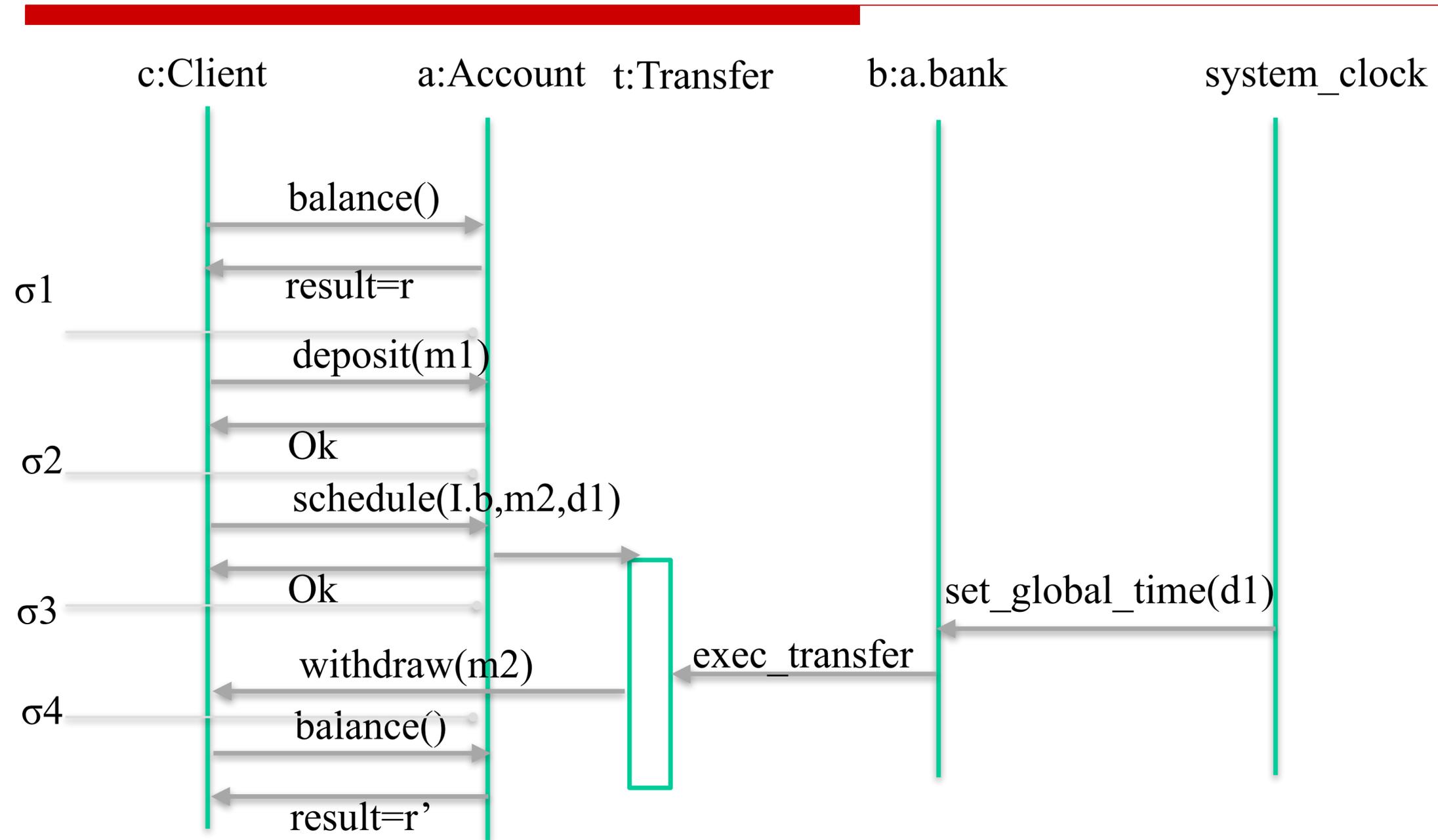


# **TD GLA 3**

## **Introduction to Test**

**Bu 3.10.20**

- Exo 1:
  - Developper un scenario de test abstrait pour un client, le systeme, et les operations deposit, balance schedule\_transfer and exec\_transfer.
  - Developper un scenario de test abstrait pour deux clients, et les operations deposit, balance withdraw pour les deux clients acceder leurs comptes cheques.
  - Developper un scenario de test abstrait pour deux clients, et les operations deposit, balance withdraw pour les deux clients acceder leurs comptes d'epargne.



- On construit la conjonction de tous les contraintes, et cela donne:

$$\langle \dots \implies r + m1 = r' + m2 \rangle$$

- La somme après le deposit doit etre la somme entre le solde final plus la somme viré.

- A) Evidemment, cette exo est basé sur la spec du TD1 et TD 2, donc les predicats definis la seront la base ici. Additionnellement, on complete la specification sur `schedule_transfer`:

```
<definition pre_schedule_transfer(a,i,b,m,d1) ≡ d1 ≥ a.bank.global_time >  
<definition post_schedule_transfer(c,i,b,m,d1,result) ≡  
  ∃ t::Transfer. isNew(t) ∧ t.target = i ∧ t.target_bank=b ∧ t.amount = m ∧  
    modifiesOnly({t})  
>
```

- On définit pour simplicité une invariante "global" consistant de tous les invariants de classe du TD1, donc:

```
<definition INV(σ) ≡ inv_Account1 (σ) ∧ inv_Bank (σ) ∧ ... >
```

- On suppose que le diagramme de sequence représente une série des transitions valides. Selon la semantique de MOAL (Revoir slides "UML with MOAL-Contracts (Part III), page 20.), cela implique:

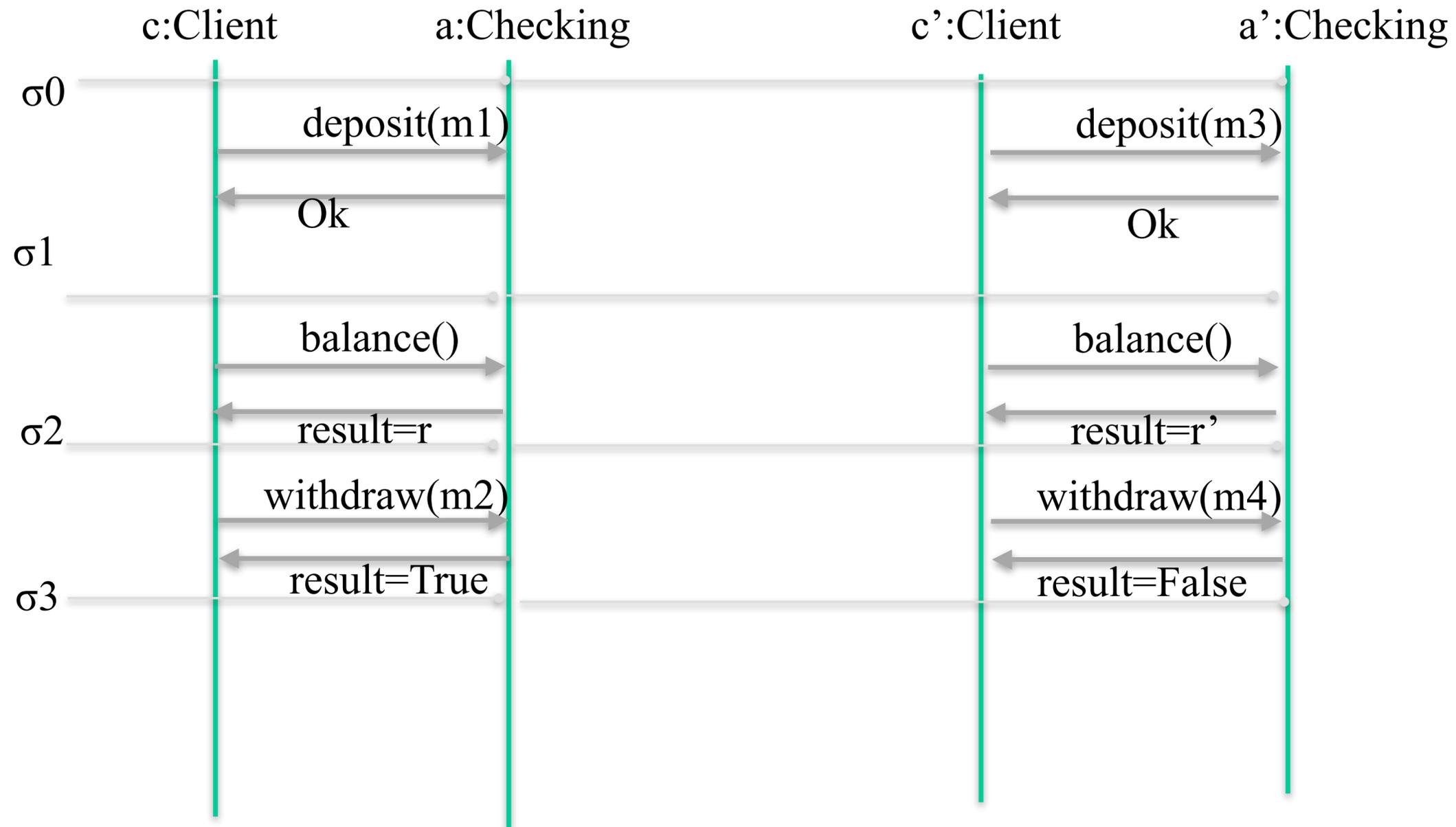
```
INV( $\sigma_1$ )  $\wedge$  r = a.balance( $\sigma_1$ )
 $\wedge$  pre_deposit(a::Account, m1::FPN)( $\sigma_1$ )
 $\wedge$  post_deposit(a::Account, m1::FPN, dummy)( $\sigma_1, \sigma_2$ )
 $\wedge$  (* compte target doit exister *)
 $\wedge$  INV( $\sigma_2$ )
 $\wedge$  pre_schedule_transfer(a, i, b, m2, d1)( $\sigma_2$ )
 $\wedge$  post_schedule_transfer(c, i, b, m2, d1, result)( $\sigma_2, \sigma_3$ )
 $\wedge$  INV( $\sigma_3$ )
 $\wedge$  pre_exec_transfer(t)( $\sigma_3$ )
 $\wedge$  post_exec_transfer(t, result)( $\sigma_3, \sigma_4$ )
 $\wedge$  INV( $\sigma_4$ )
 $\wedge$  pre_withdraw(a, m2)( $\sigma_4$ )
 $\wedge$  post_withdraw(a, m2, result)( $\sigma_4, \sigma_5$ )  $\wedge$ 
 $\wedge$  r' = a.balance( $\sigma_5$ )
 $\wedge$  INV( $\sigma_5$ )
```

- De cette conjonction des contraintes, on peut conclure la relation entre  $\langle r \rangle$  et  $\langle r' \rangle$ :

$$\langle \dots \rangle \implies r + m1 = r' + m2$$

- La somme après le deposit doit etre la somme entre le solde final plus la somme viré.
- Ceci est un test pour tous les  $\langle r \rangle$ ,  $\langle m1 \rangle$ ,  $\langle r' \rangle$  et  $\langle m2 \rangle$ , donc un test abstrait qui est valide dans n'importe quel contexte d'evaluation et n'importe quel valeurs concrets pour  $r$ ,  $m1$ ,  $r'$  et  $m2$  ...

- B) deposit balance withdraw (independence test of two accounts)



- De cette conjonction des contraintes, on peut conclure un cas de test abstrait pour client a:

```

text< Abstract test case for Account <a> (Objectif: le client a mis suffisamment de sous
      pour le withdraw est un success) :
  □< ...  ⇒ a.balance(σ0) + m1 = r ∧ a.balance(σ3) = r - m2 ∧
          ( (a.balance(σ2) - m2 ≥ 0 ∧ a.balance(σ3) = a.balance(σ2) - m2)
            ∨ (a.balance(σ2) - m2 < 0 ∧
              a.balance(σ2) - m2 - a.overdraft_interest > c.max_overdraft ∧
              a.balance(σ3) = a.balance(σ2) - m2 - a.overdraft_interest))
  >

```

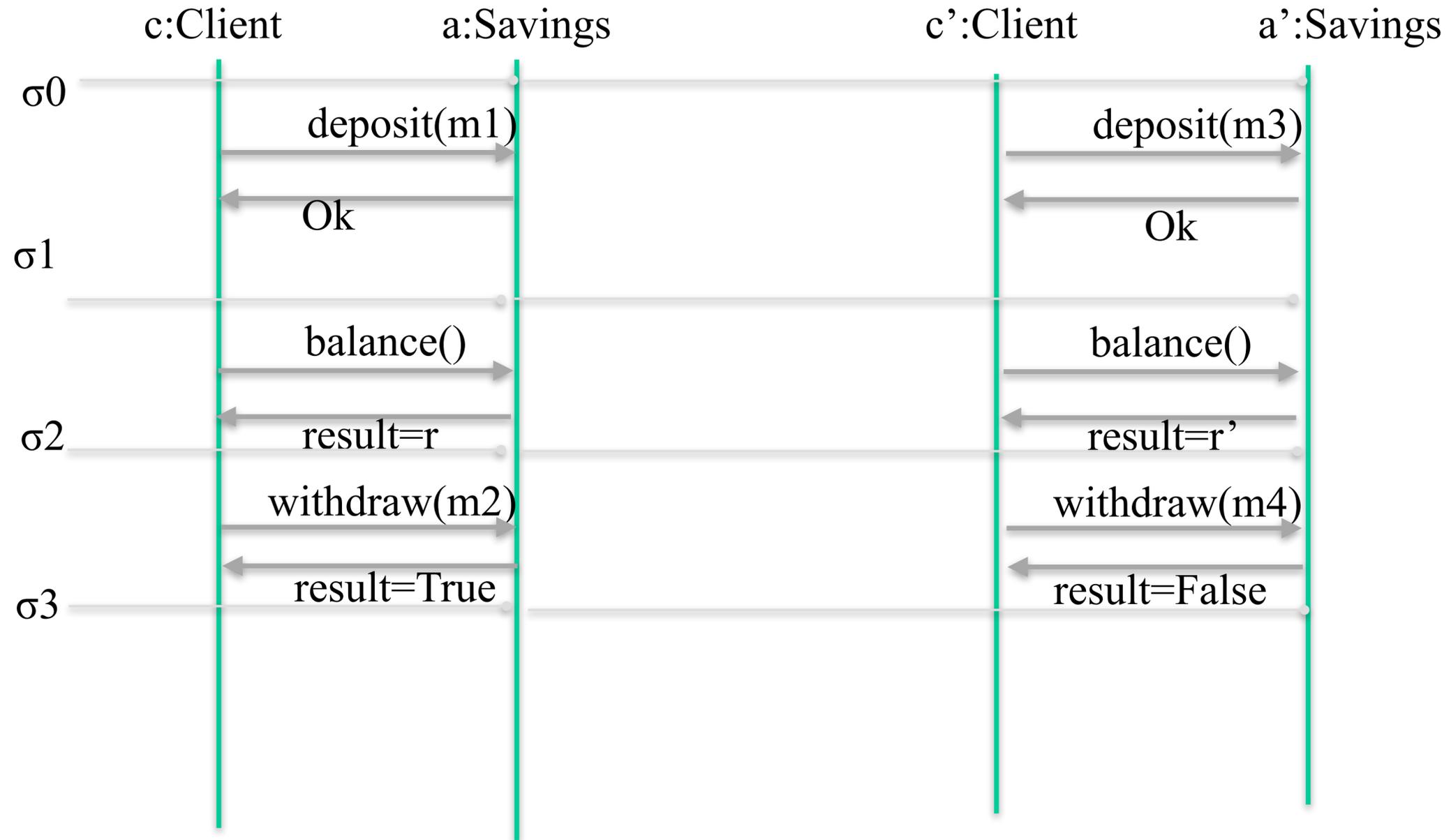
- De la meme conjonction des contraintes, on peut conclure pour client a':

```

Abstract test case for Account <a'> (recall that the withdraw fails ! Objectif: le client n'a pas
mis suffisamment de sous et le withdraw est un echec.):
  □< ...  ⇒ a'.balance(σ0) + m3 = r' ∧ a.balance(σ3) = r ∧
          (a'.balance(σ2) - m4 - a.overdraft_interest ≤ c.max_overdraft)
  >

```

- B) deposit balance withdraw (independence test of two accounts)



- HOMEWORK !!!

## Exo 2.1

	Objectif de test	Formule du résultat attendu	Données d'entrée		Résultat attendu
			A	B	
1	Prix sans réduction : $5A + 10B < 200$ et $A < 45$	$5A + 10B$	3	5	65
2	Réduction de 5% : $200 \leq 5A + 10B < 1000$ et $A < 45$	$(5A + 10B) \times 0,95$	20	20	285
3	Réduction de 20% : $5A + 10B \geq 1000$ et $A < 45$	$(5A + 10B) \times 0,8$	20	100	880
4	Réduction de 5% puis 10% : $200 \leq 5A + 10B < 1000$ et $A \geq 45$	$(5A + 10B) \times 0,95 \times 0,9$	60	10	342
5	Réduction de 20% puis 10% : $5A + 10B \geq 1000$ et $A \geq 45$	$(5A + 10B) \times 0,8 \times 0,9$	60	100	936

## Exo 2.2 (Borderline Test)

<i>Limite</i>	<i>Objectif</i>	$5A + 10B$	$A$	$B$	<i>Résultat attendu</i>
<i>Entre 1 et 2</i>	$5A + 10B < 200$ et $A < 45$	<b>195</b>	1	19	195
	$200 \leq 5A + 10B < 1000$ et $A < 45$	<b>200</b>	2	19	190
<i>Entre 2 et 3</i>	$200 \leq 5A + 10B < 1000$ et $A < 45$	<b>995</b>	1	99	945,25
	$5A + 10B \geq 1000$ et $A < 45$	<b>1000</b>	2	99	800
<i>Entre 4 et 5</i>	$200 \leq 5A + 10B < 1000$ et $A \geq 45$	<b>995</b>	99	50	850,725
	$5A + 10B \geq 1000$ et $A \geq 45$	<b>1000</b>	100	50	720
<i>Entre 2 et 4</i>	$5A + 10B \geq 1000$ et $A \geq 45$	300	<b>44</b>	8	285
	$200 \leq 5A + 10B < 1000$ et $A \geq 45$	305	<b>45</b>	8	260,775
<i>Entre 3 et 5</i>	$5A + 10B \geq 1000$ et $A < 45$	1220	<b>44</b>	100	976
	$5A + 10B \geq 1000$ et $A \geq 45$	1225	<b>45</b>	100	882