

PROJET COPS

Deuxième réunion le 5 Octobre 2006

Composition des services Web dans une communauté de confiance

Fahima Cheikh

Lilac

IRIT

Plan

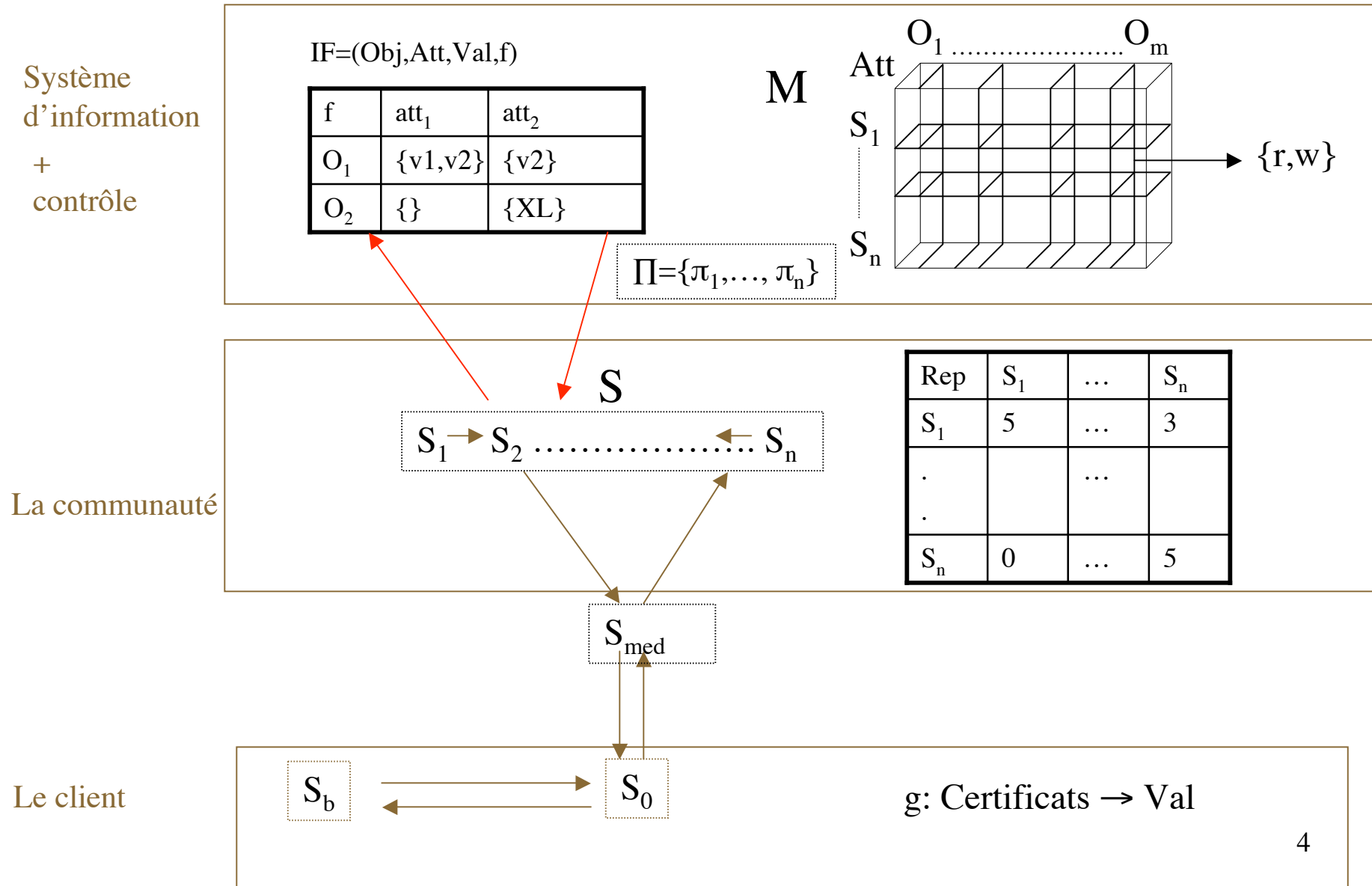
- Modèles d'une communauté de confiance
- Problème de la composition de services
- Complexité du problème
- Travaux futurs

Modèle de la communauté de confiance

→ Communication

→ Lecture et écriture

Un modèle de communauté de confiance

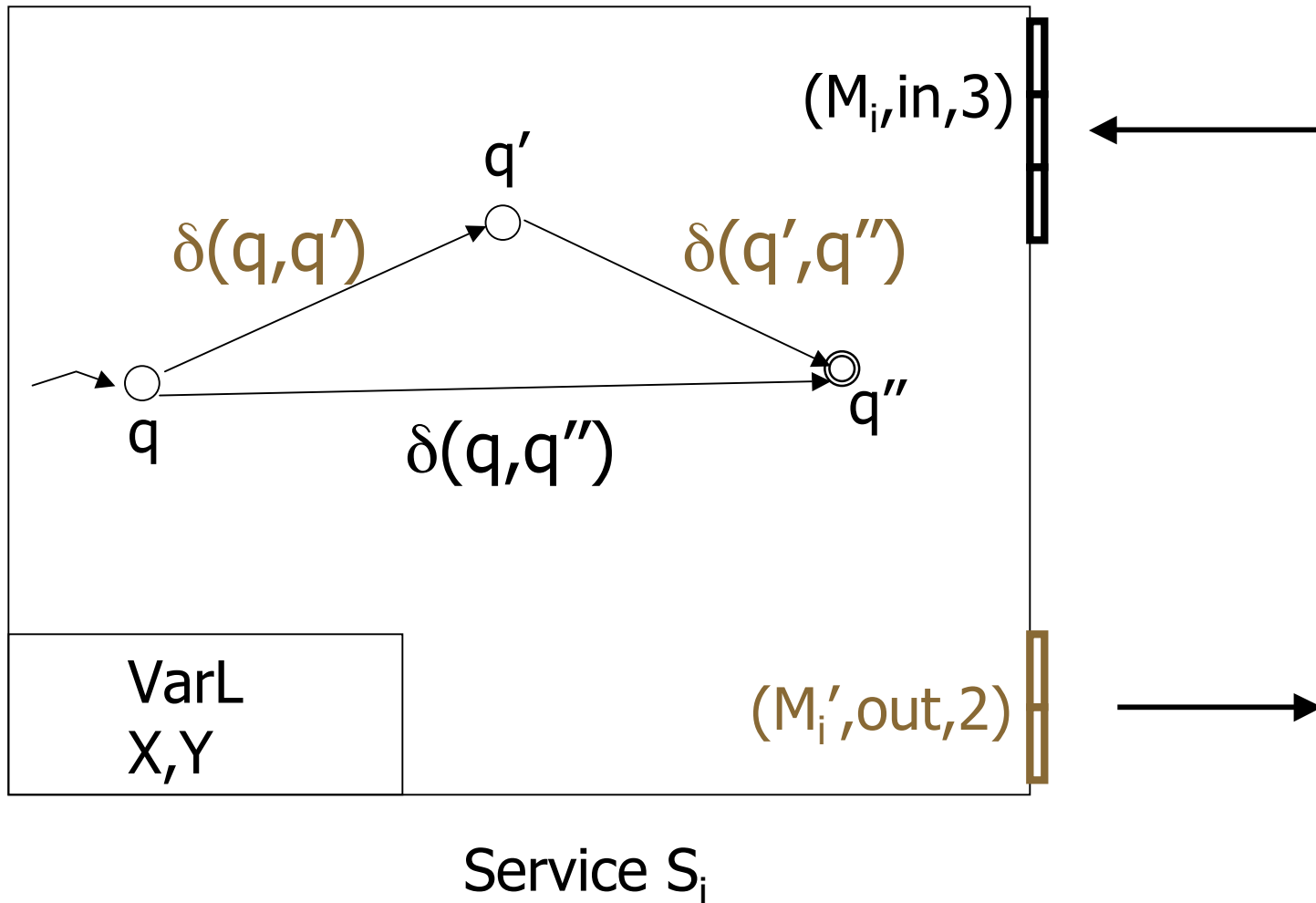


Exemple: Gestion d'un magasin

IF=(Obj,Att,Val,f)

f	Att-Nom	Att-Prix
O ₁	{Jupe}	{20}
O ₂	{Robe}	{50}

Service Web



Transition

- Fonction de transition (relative à S_i)

$\delta(q, q') = \{(C, \alpha) \mid C \text{ est une condition, } \alpha \text{ une séquence d'opérations primitives et de commande de } \Pi\}$.

- Condition

$C := T \mid (\theta_1 = \theta_2) \mid (\theta \in f(z, a)) \mid \text{Rep}(S_i, \text{Cert.Issuer}) \geq k \mid$
 $\text{Cert.att} \geq k \mid \neg C \mid (C_1 \wedge C_2) \mid \exists z C$

θ : une variable locale ou une constante de type Val

z : une variable qui reçoit des valeurs dans Obj

a : un attribut de Att

Cert: un des certificats du client

Cert.Issuer: l'indice du service qui a émis le certificat

k : une constante de type Val

Cert.att: la valeur de l'attribut du certificat

Transition

- Opérations primitives

- créer l'objet z
- détruire l'objet z
- ajouter θ à $f(z,a)$
- retrancher θ de $f(z,a)$



Mise à jour de IF

- $x := \theta$

- $?P(\theta_1, \dots, \theta_n)$

recevoir un message

- $!P'(\theta_1, \dots, \theta_m)$

transmettre un message

- $\text{Cert.Issuer} := S_i$

- $\text{Cert.att} := \text{cert.att} \pm k$

- $\text{Cert.att} := k$



Mise à jour des certificats

Transition

- Commandes de Π

Si cond **alors** séquence d'opérations primitives sur M

- $\text{cond} ::= T \mid d \in M(y,z,a) \mid \neg C \mid (C_1 \wedge C_2) \mid \exists z C$

- Opérations primitives

- introduire r dans $M(y,z,a)$

- supprimer r de $M(y,z,a)$

d : droit de lecture ou d'écriture ($d \in \{r,w\}$)

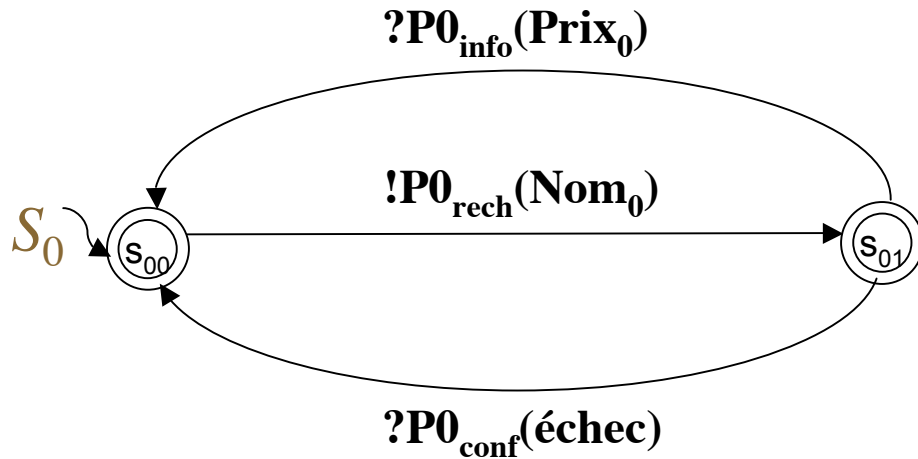
M: Matrice des droits d'accès

y: variable qui reçoit des valeurs dans S

z: variable qui reçoit des valeurs dans Obj

a: un attribut dans Att

Exemple: Gestion d'un magasin

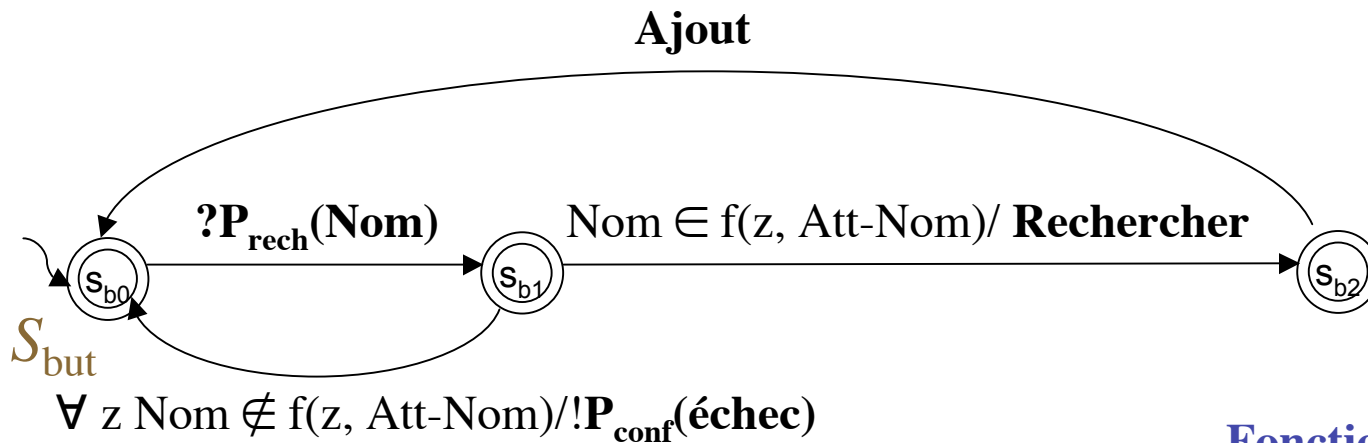


Rechercher

Prix:=f(z, Att-prix); !P_info(Prix)

Ajout

créer l'objet z;
 ajouter Nom f(z, Att-Nom)
 ajouter Prix f(z, Att-Prix)



Fonction g:

$g(\text{cred.Issuer})=1$

$g(\text{cred.Montant})=3$

Le client

Exemple: Gestion d'un magasin

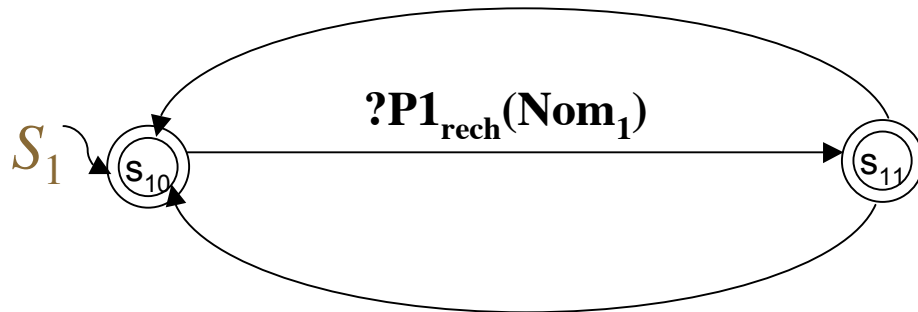
M	O ₁		O ₂	
S ₁	Att-Nom	Att-Prix	Att-Nom	Att-Prix
	{r}	{r}	{r}	{r}
S ₂	{r,w}	{r,w}	{r,w}	{r,w}

$\Pi = \{\pi_1\}$, π_1 : introduire **w** dans M(y,z,a)

Exemple: Gestion d'un magasin

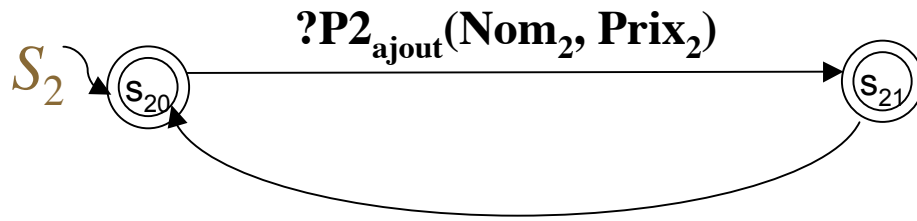
$\text{Nom} \in f(z, \text{Att-Nom})$
 $\wedge \text{Rep}(\text{cred.Issuer}) \geq 1 \wedge \text{cred.Montant} \geq 1$
/ **Rechercher**₁

Rechercher₁
 $\text{Prix}_1 := f(z, \text{Att-prix});$
 $!P1_{\text{info}}(\text{Prix}_1);$
 $\text{cred.Issuer} := S_1$
 $\text{cred.Montant} := \text{cred.Montant} - 1;$



$\forall z \text{Nom}_1 \notin f(z, \text{Att-Nom})$
 $\forall v \text{Rep}(\text{cred.Issuer}) < 1 \vee \text{cred.Montant} < 1$
/!**P1**_{conf}(échec)

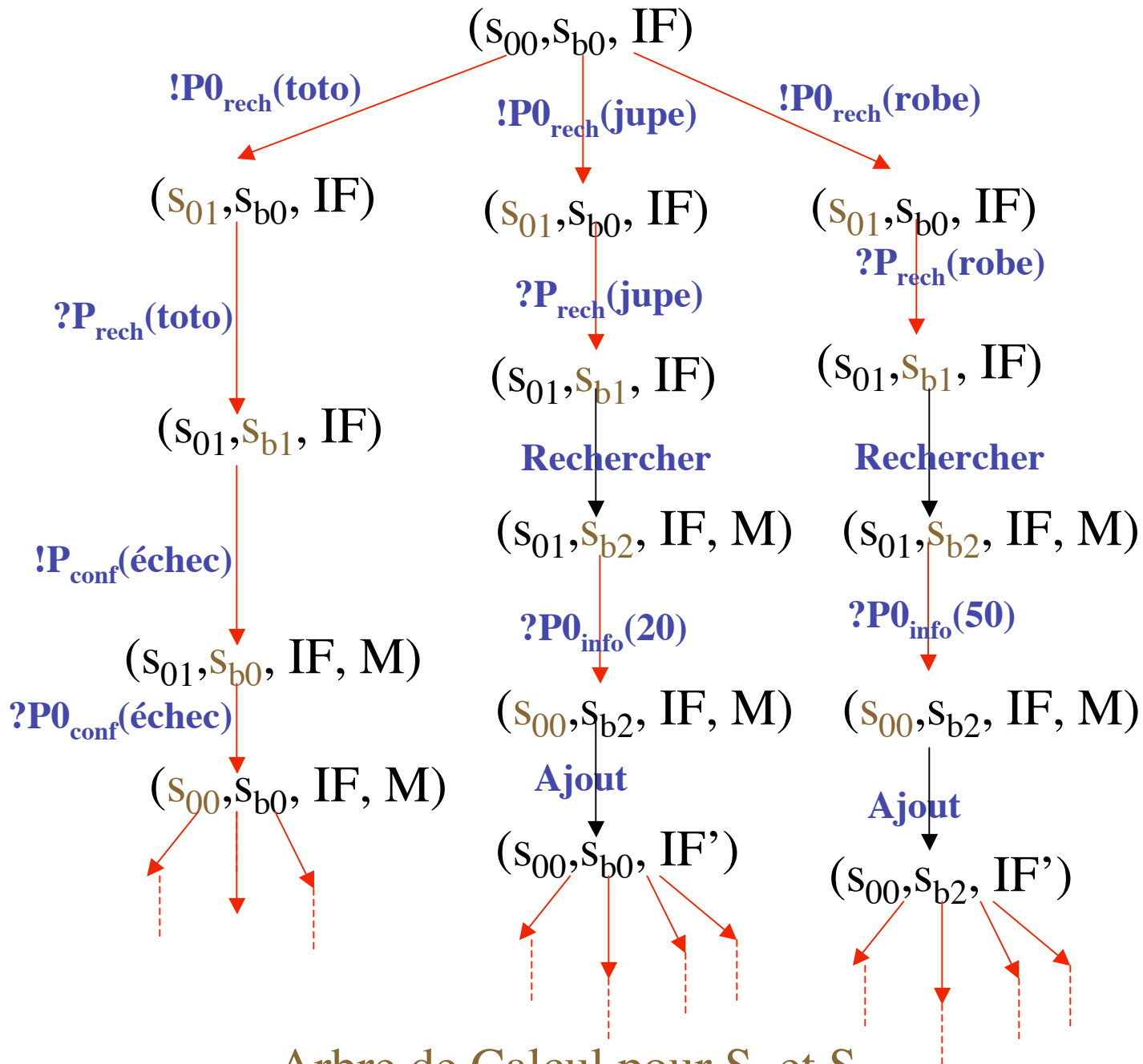
Exemple: Gestion d'un magasin



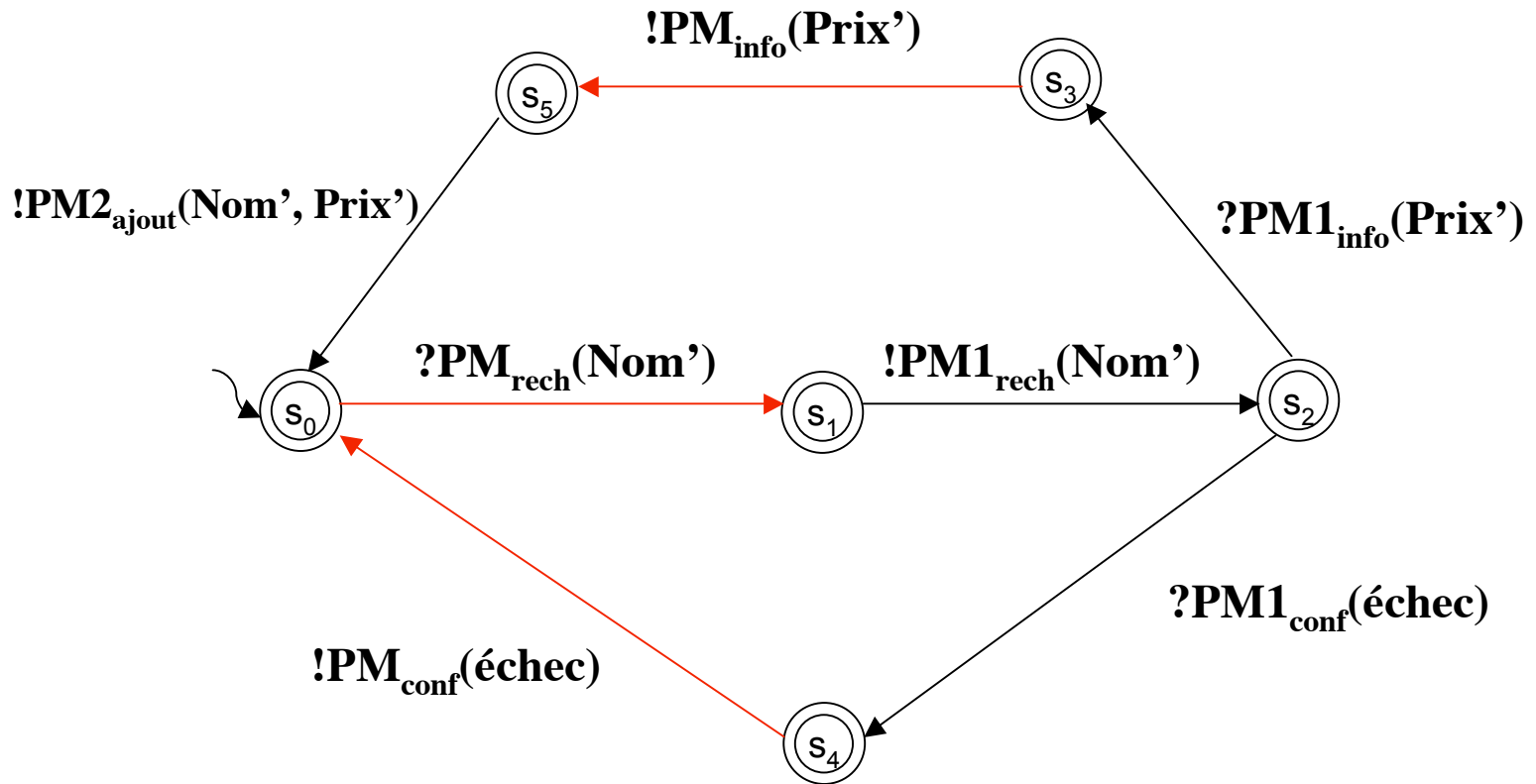
$\text{Rep}(\text{cred.Issuer}) \geq 1$
 $\wedge \text{cred.Montant} \geq 1$ / **Ajout₂**

Ajout₂
 créer l'objet z;
 π ;
 ajouter Nom₂ f(z, Att-Nom);
 ajouter Prix₂ f(z, Att-Prix);
 cred.Issuer:=S₂;
 cred.Montant:=cred.Montant-1;

Rep	S ₁	S ₂
S ₁	1	0
S ₂	1	1



Arbre de Calcul pour S_0 et S_{but}



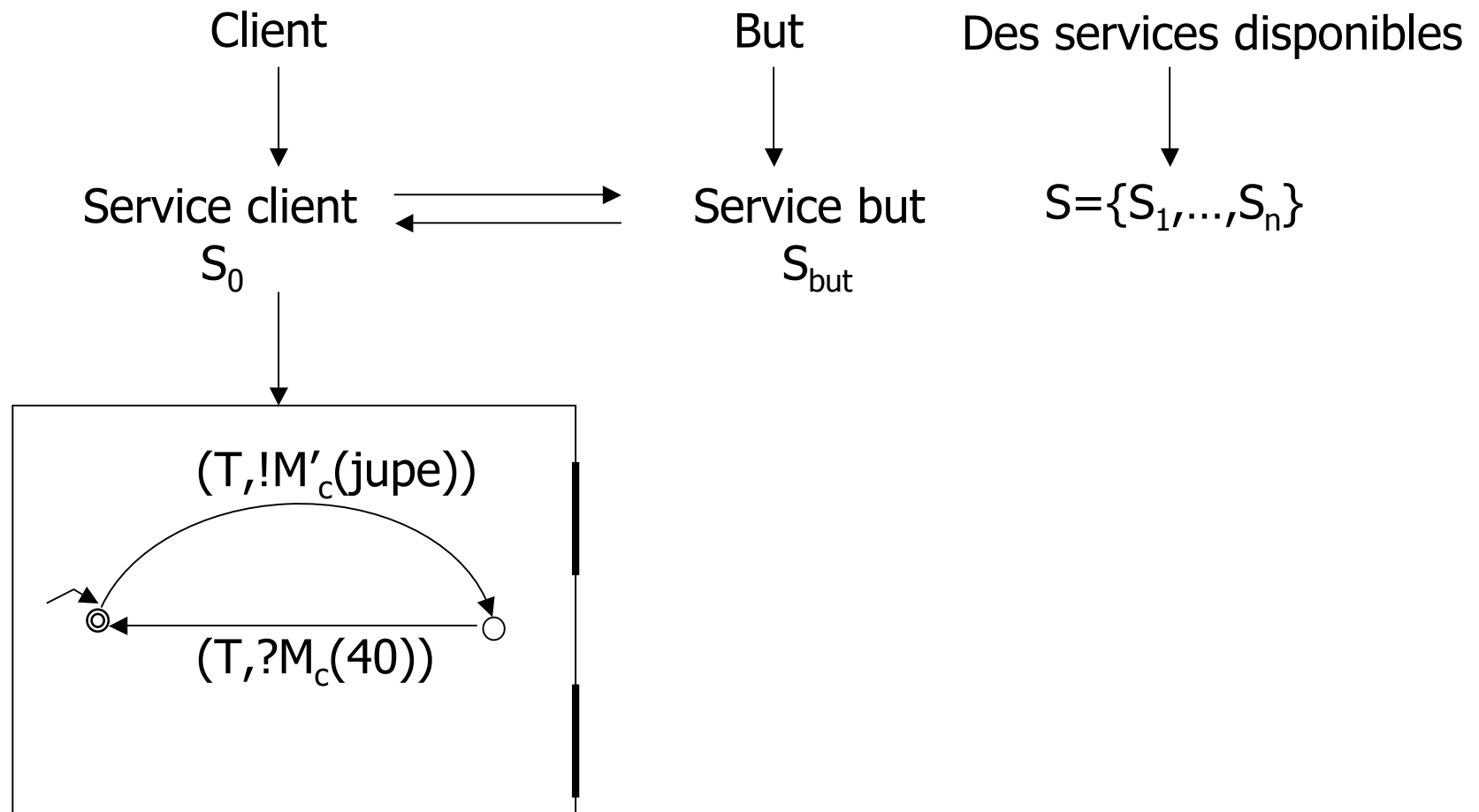
Le médiateur



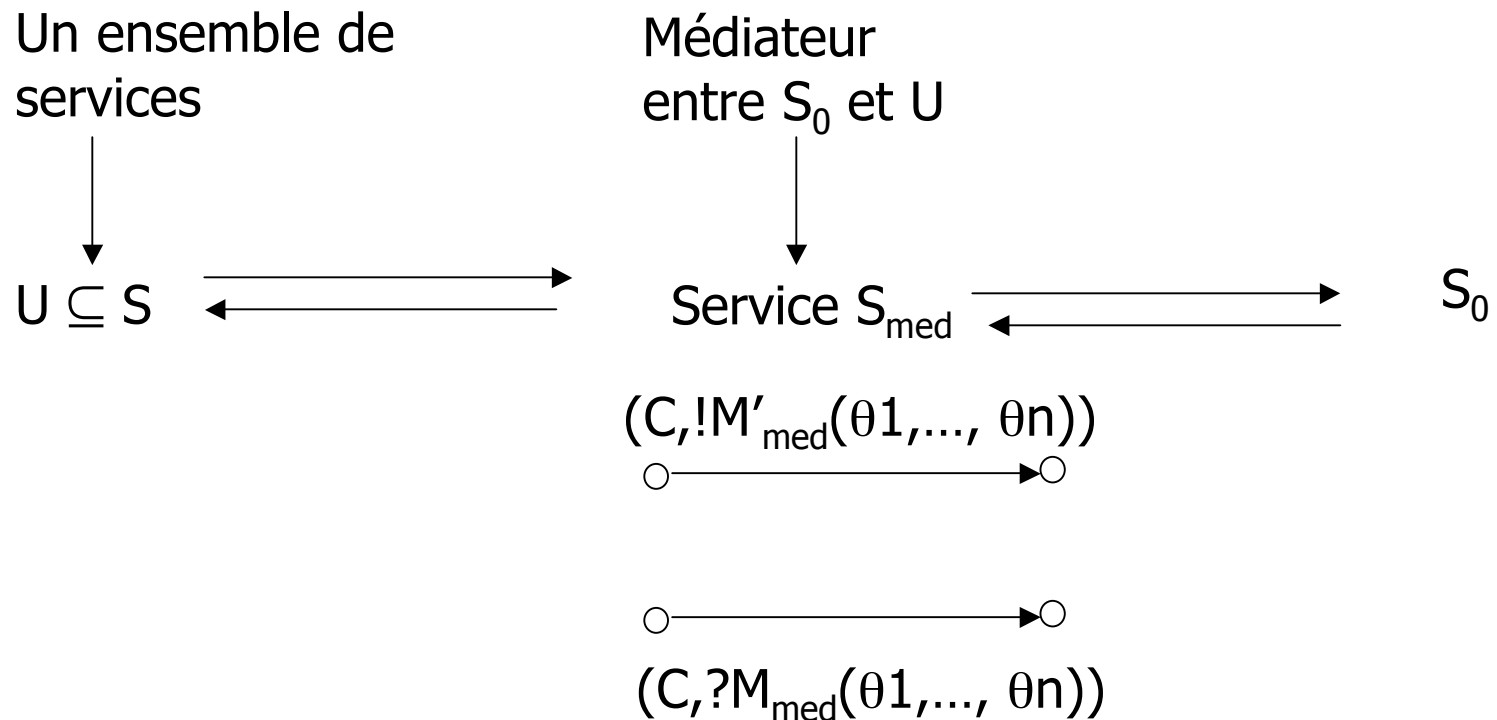
Arbre de Calcul pour S_0, S_{med}, S_1 et S_2

Problème de la composition de services

Entrée



Sortie



- Tels que le **comportement** de S_{med} combiné avec l'ensemble U doit avoir le même **comportement** que S_{but}

Complexité du problème

Décidabilité

- Théorème 1

Le problème de la composition est indécidable pour l'équivalence de traces

Preuve. Nous réduisons le problème uniforme de l'arrêt des machines de Minsky au problème de la composition

Travaux futurs

Travaux futurs

- Donner des résultats de complexité en considérons d'autres types d'équivalence
- Donner des résultats de complexité en faisant certaines restrictions sur le modèle