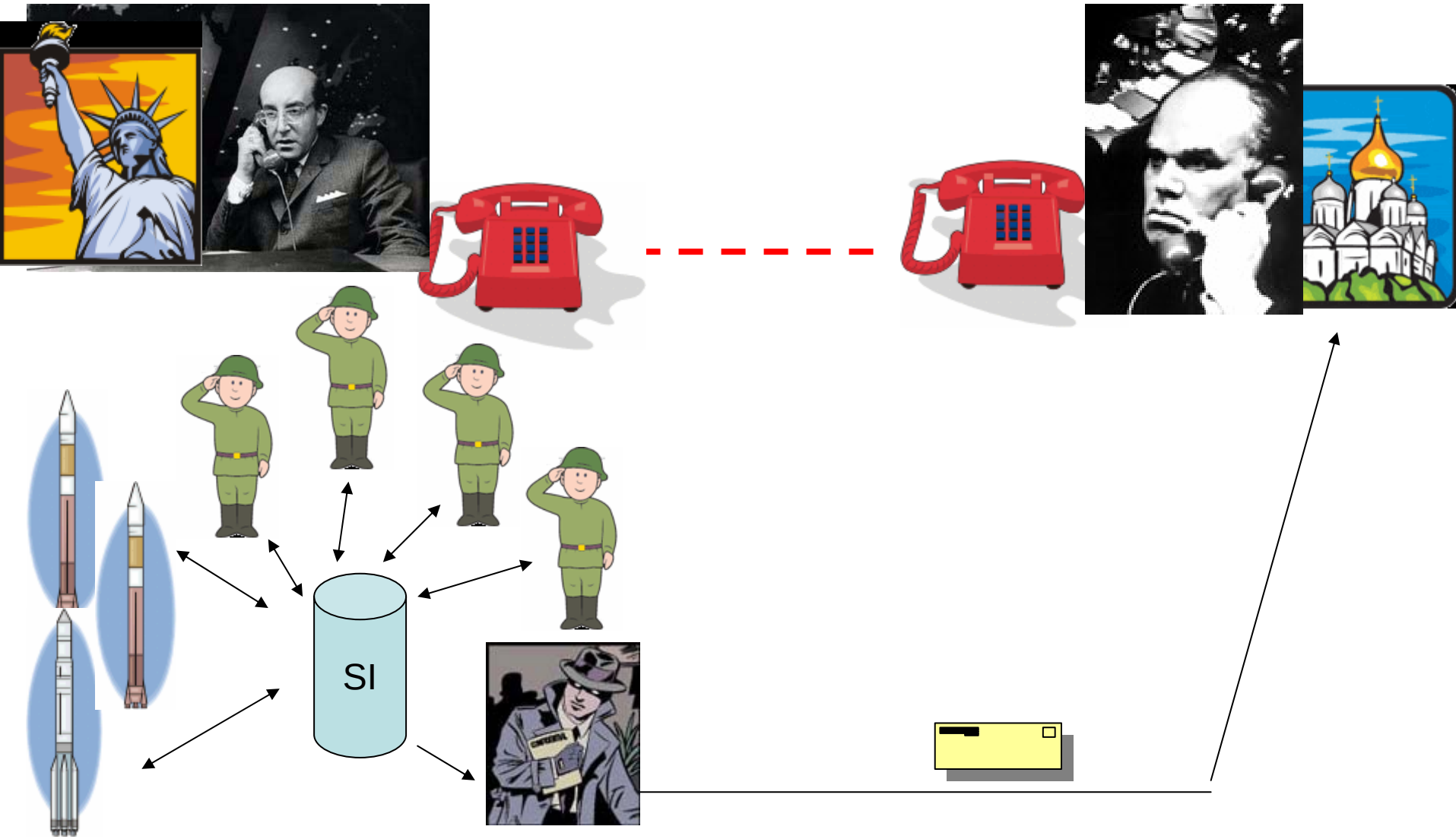
Non Interférence
&
Canaux cachés

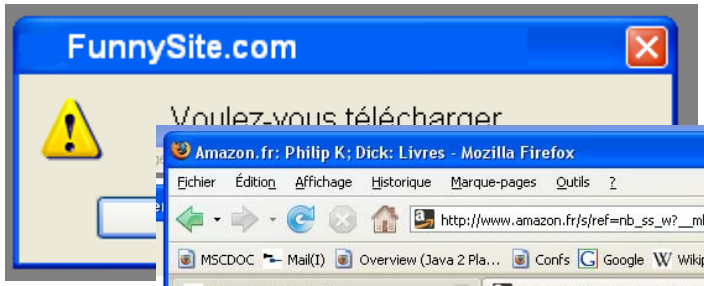
Motivations

- Protéger des informations sensibles
- Eviter des flots d’information non prévus
  - Fuites d’information
  - Utilisation frauduleuse d’un système comme medium de communication
  - …

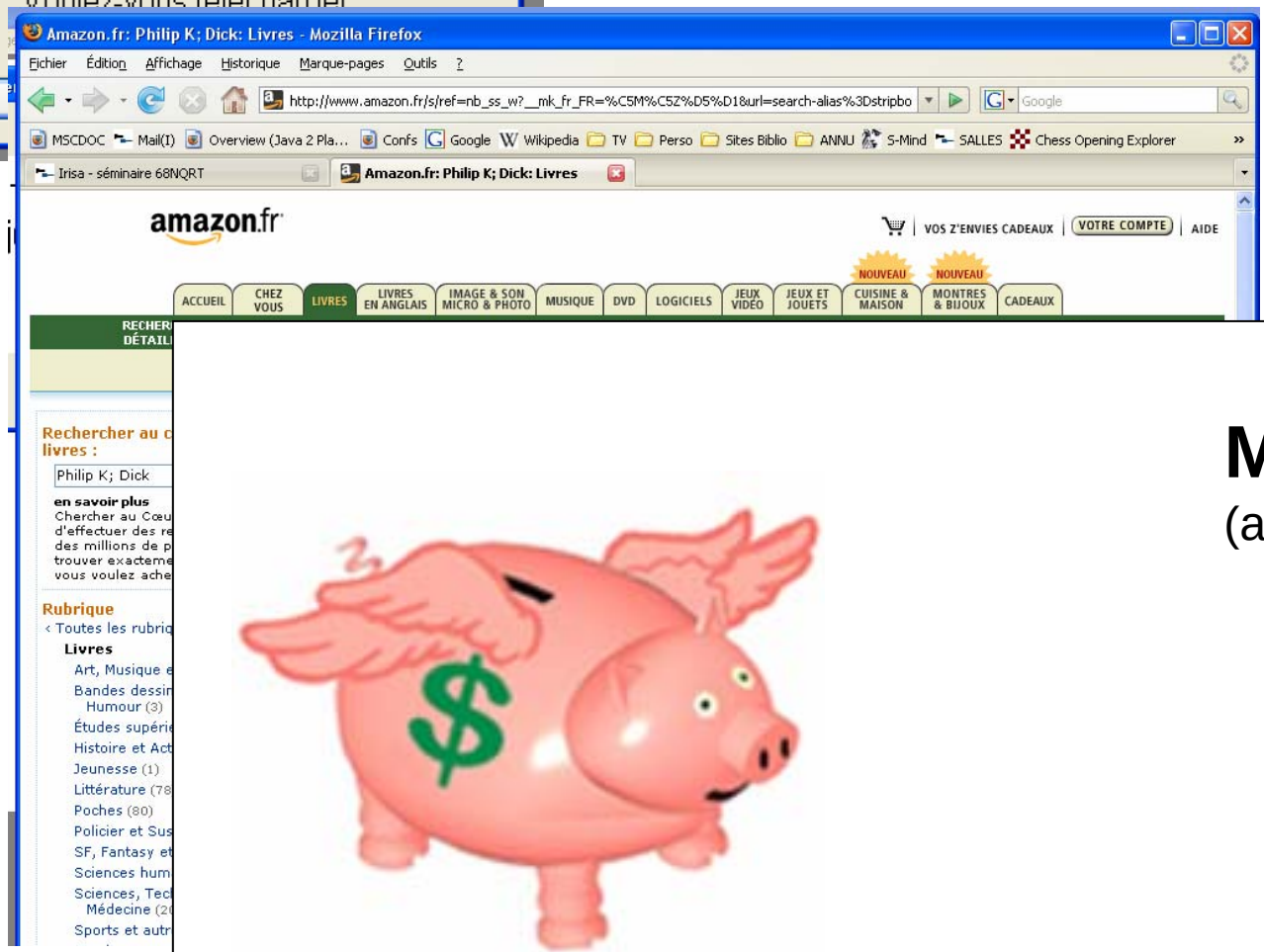
Des SI à … nos jours



Aujourd’hui…



Lundi



Mardi

Mardi
(aussi)

Les problèmes d'information…

- Exigences des systèmes d'information
  des 70's
- Toujours d'actualité, mais sous d'autres formes

Recommandations de normes :

« use reproducible & formal techniques to find illegal flows of information »

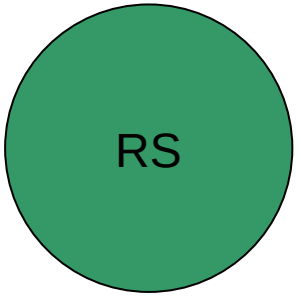
[comon criteria]

[Light pink book]

[etc]

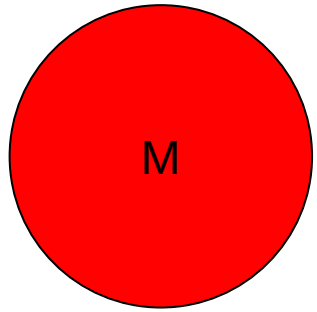
Motivations

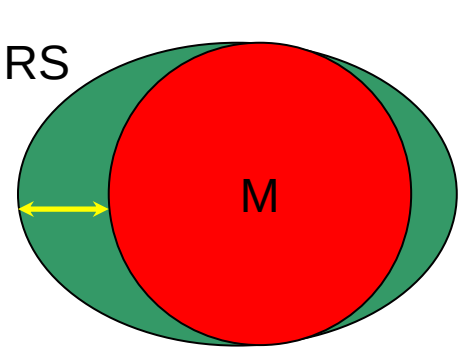
Système réel

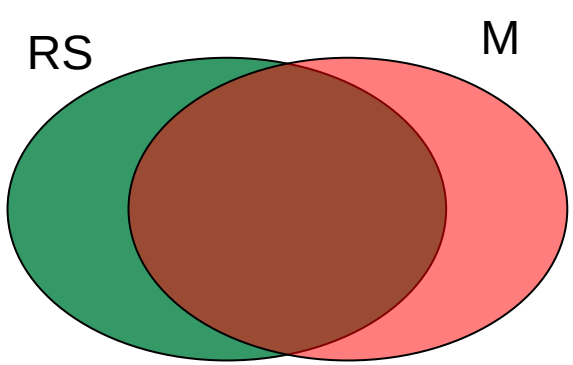


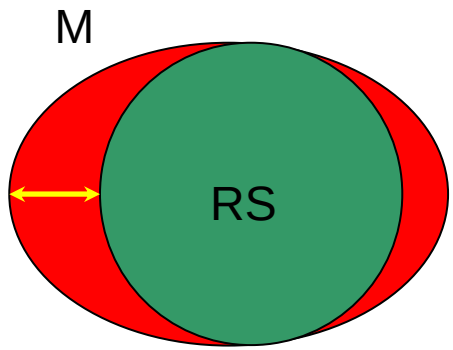
<u>But</u> : Prouver à partir M que RS est sûr…

Modèle comportemental

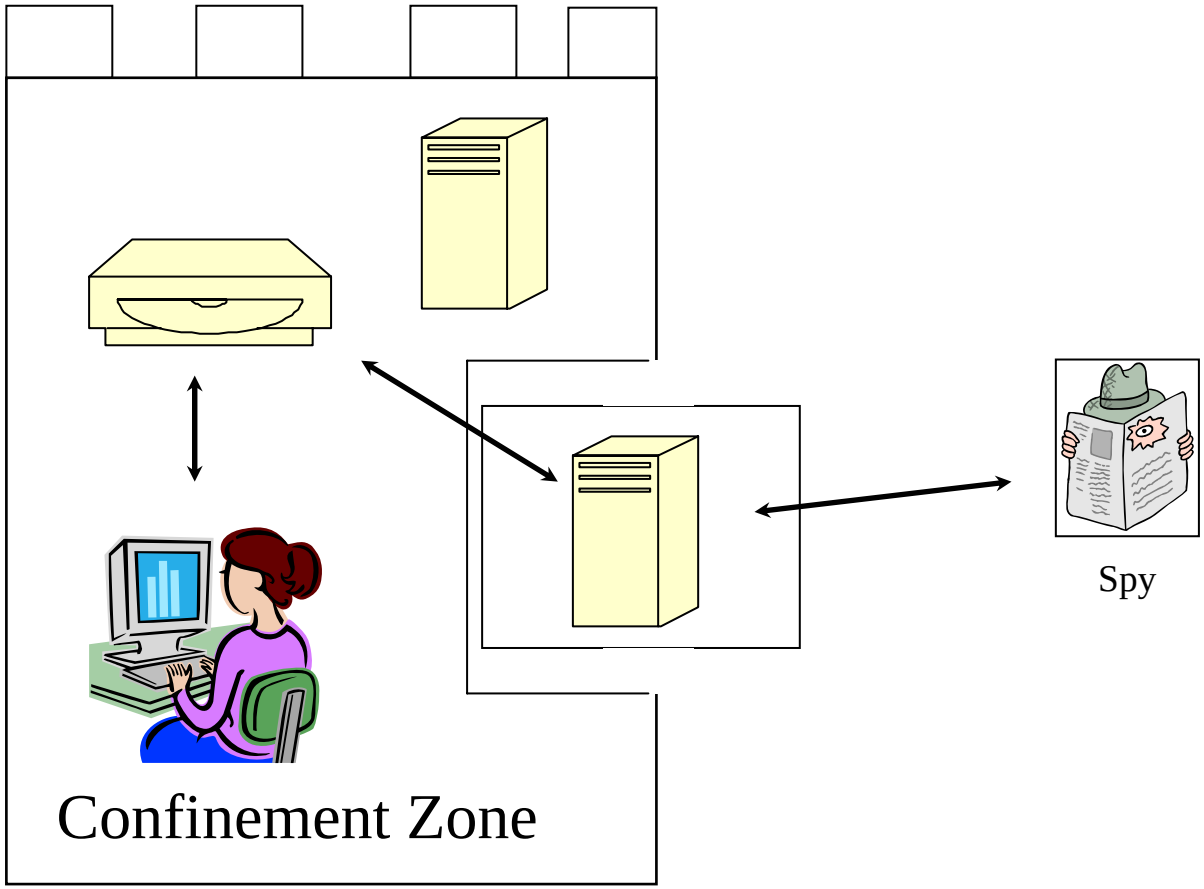






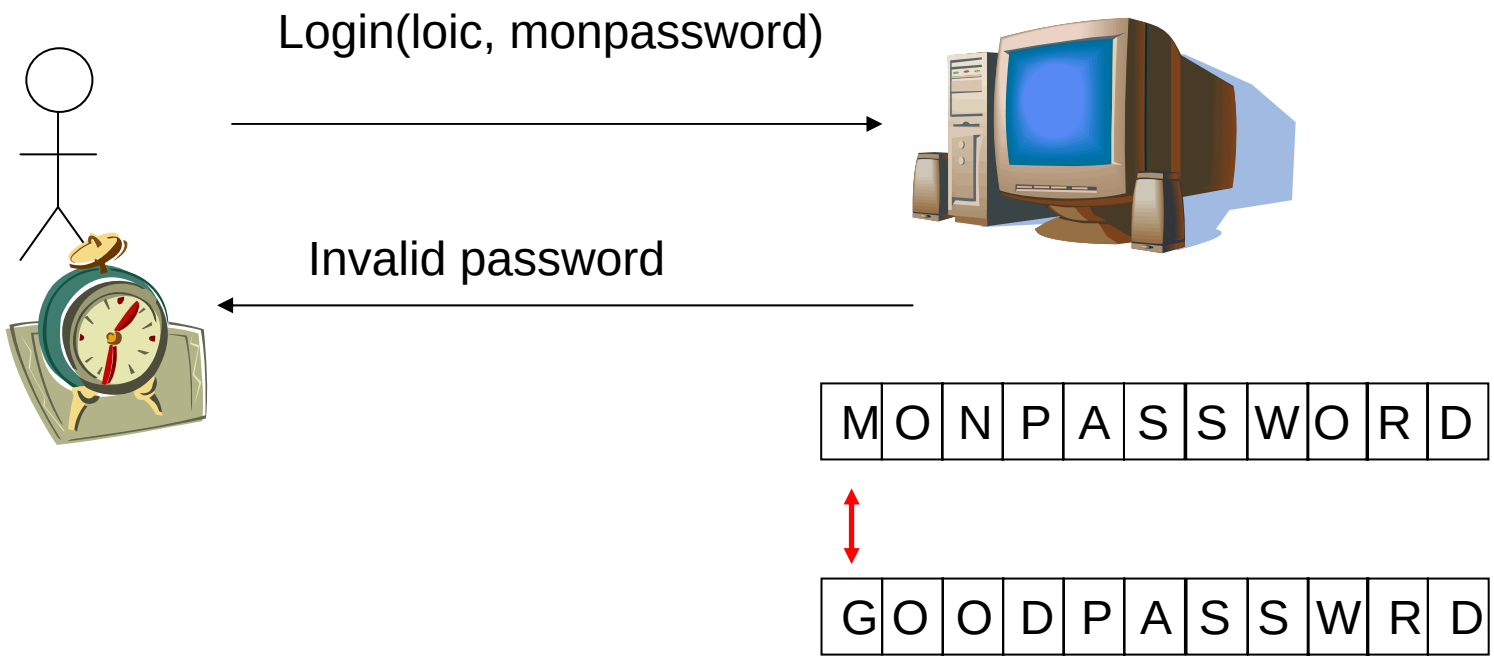


Interférence

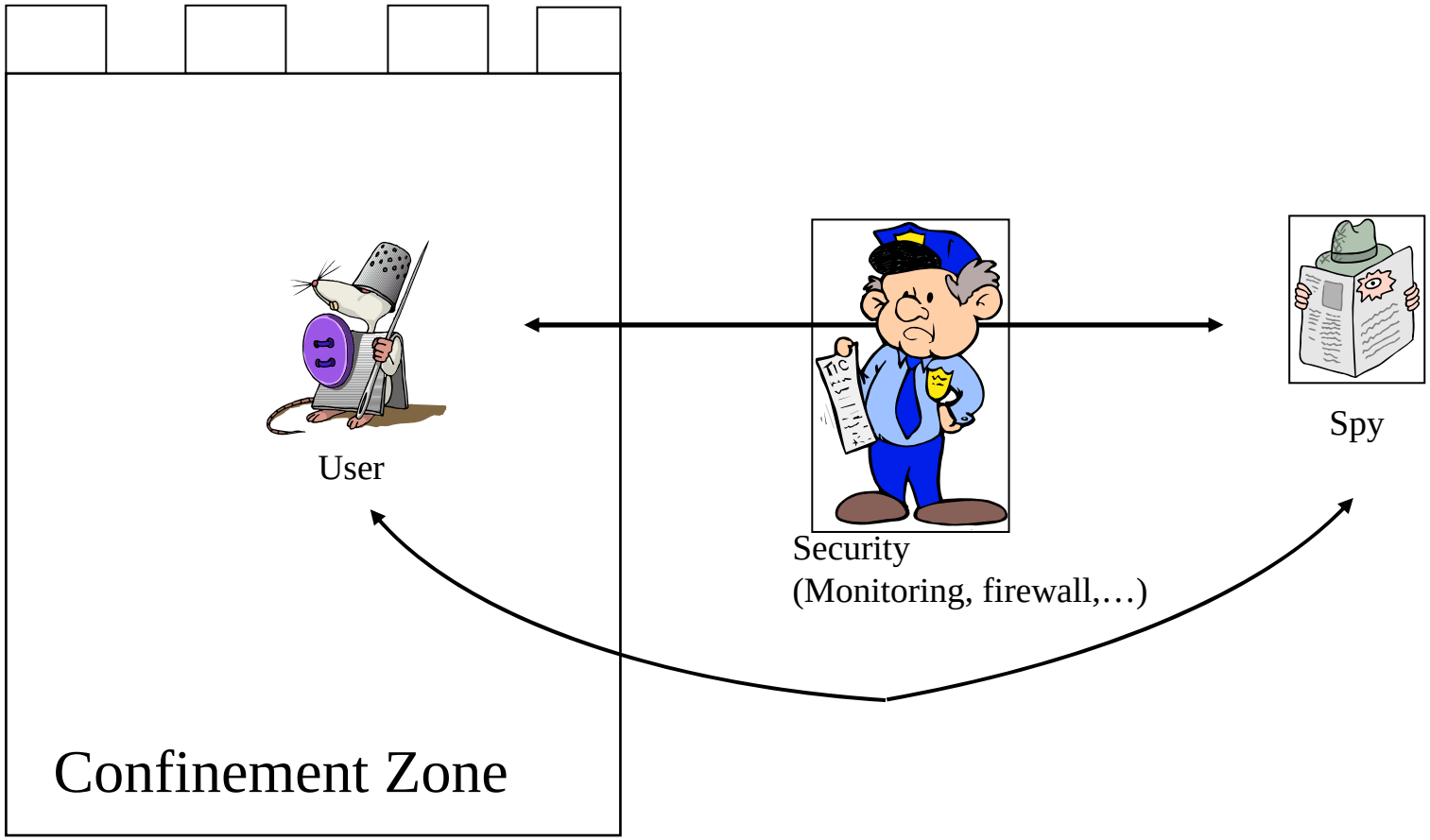


Une interférence connue

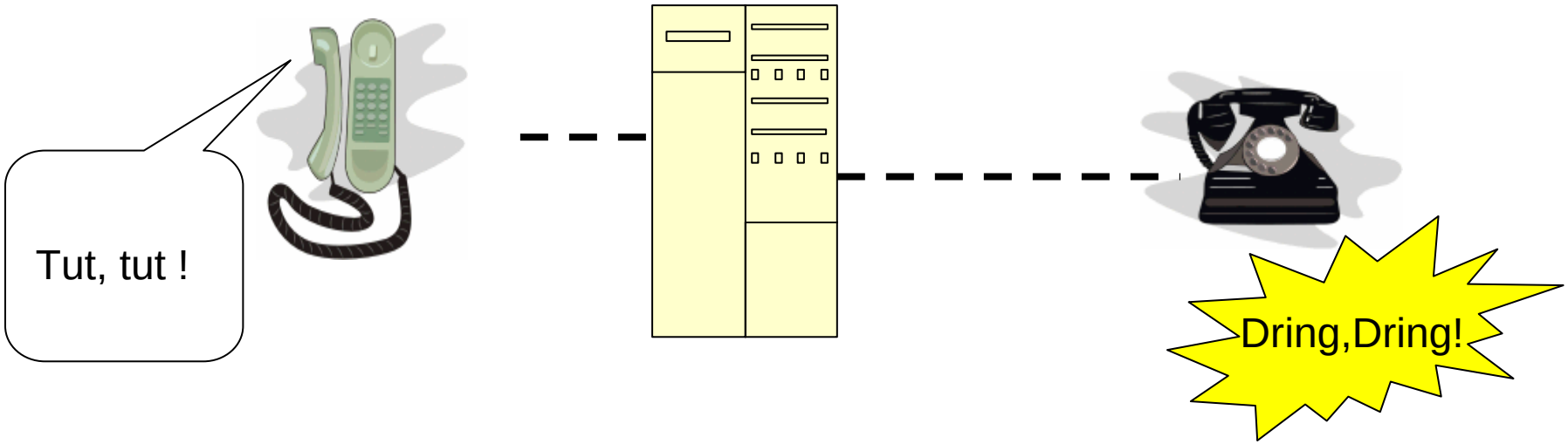
- Les mots de passe Unix



Canal caché



Votre premier canal caché…



Impensable ?

Ce canal a tout de même obligé France Télécom à modifier ses lignes …
… et à facturer les appels depuis certains pays à la troisième sonnerie

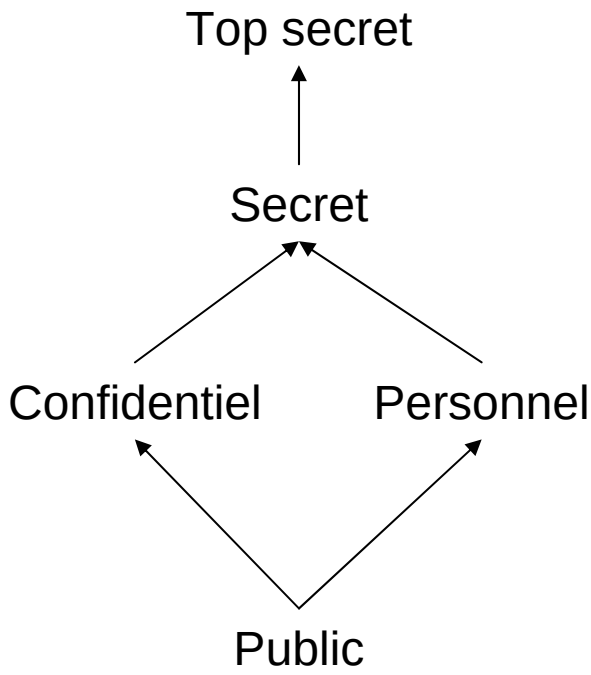
Plan

- Sécurité Multi-niveaux
- Non interférence
  - Goguen & Messeguer
  - Focardi & Gorrieri
  - Inconvénients
- Canaux cachés
  - Quantification
  - Théorie de l’information
  - Jeux

Sécurité Multi-niveaux

[Bell & La Padulla]

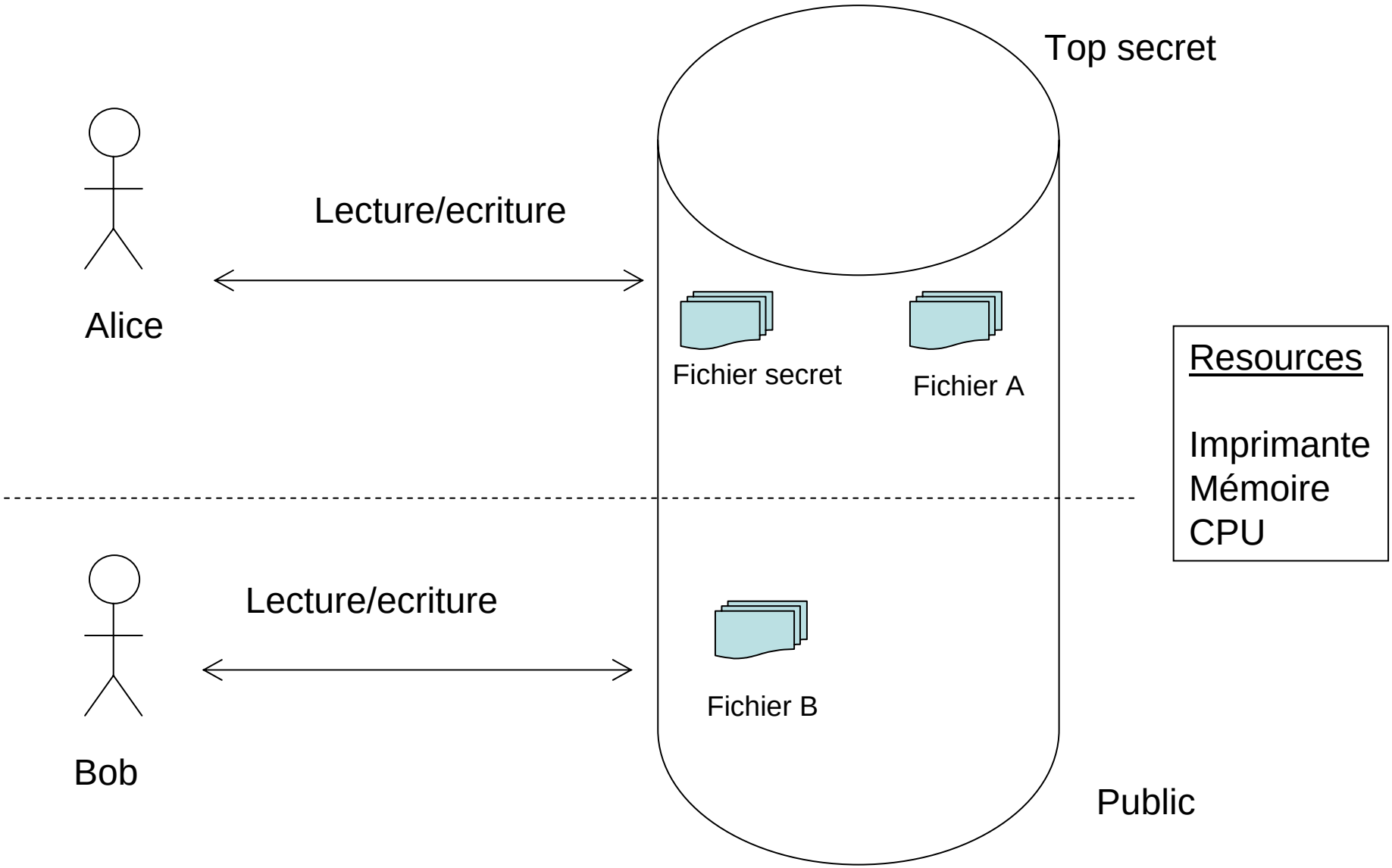
- Systèmes d'information
  - Plusieurs niveaux de sécurité
  - Un ensemble d'agents avec des accréditations
  - Un ensemble d'objets avec un niveau de sécurité



Sécurité Multi niveaux

- Règles de B&LP
  - No read up
  - No write down
  - Contrôle d’acces aux objets discrétionnaire codé dans une matrice

Sécurité multi niveaux



Matrice partagée

[Kemerrer]

	high Fichier Secret	Fichier A	Fichier B	
high Alice	r,w	r,w	-	
high Georges	-	r,w	r,w	
Bob	-	-	r,w	
John	-	-	r,w	

Matrice partagée

[Kemerrer]

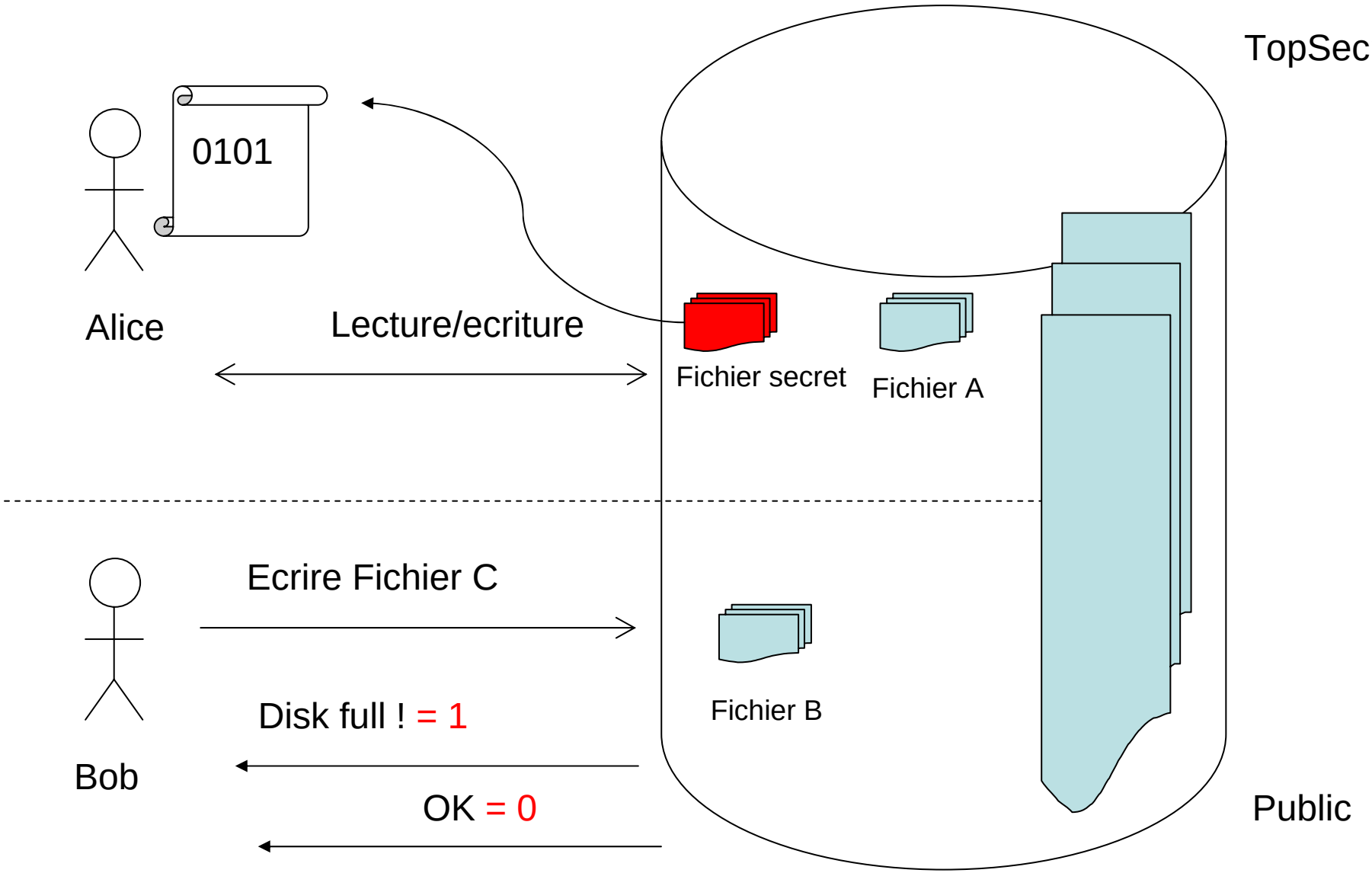
	high Fichier Secret	Fichier A	Fichier B	Alice	Georges	Bob	John
high Alice	r,w	r,w	-				
high Georges	-	r,w	r,w				
Bob	-	-	r,w				
John	-	-	r,w				
Fichier Secret	-	r,w	r,w				
Fichier A	r,w	-	r,w				
Fichier B	r,w	r,w	-				

Matrice partagée

- Calcul de la fermeture transitive des accès aux objets

- Flot illégal s'il existe un accès entre sujets /objets à des niveaux de classification différents violant les règles de Bell & La Padulla

Un exemple de canal caché



Non Interférence

- Goguen & Meseguer 1982

  – Un systéme S
  – Des utilisateurs u, v.

  « u interferes with v in system S iff
  what u does can affect what v can observe or do »

Non Interférence

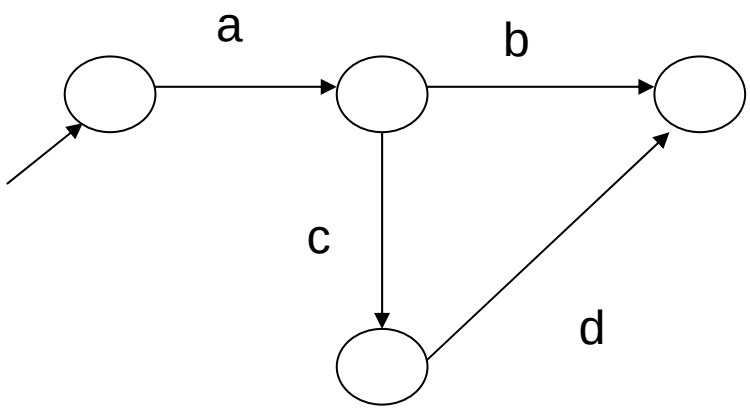
- Goguen & Meseguer 1982

  – Un systéme S
  – Des utilisateurs u, v.

  u does not interfere with *v* in system S iff

  for every behavior of u, what v can observe or do is equivalent.

Un modèle simple : les automates

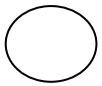


a: lock(f)

b: delete(f)

c: append(f)

d: close(f)

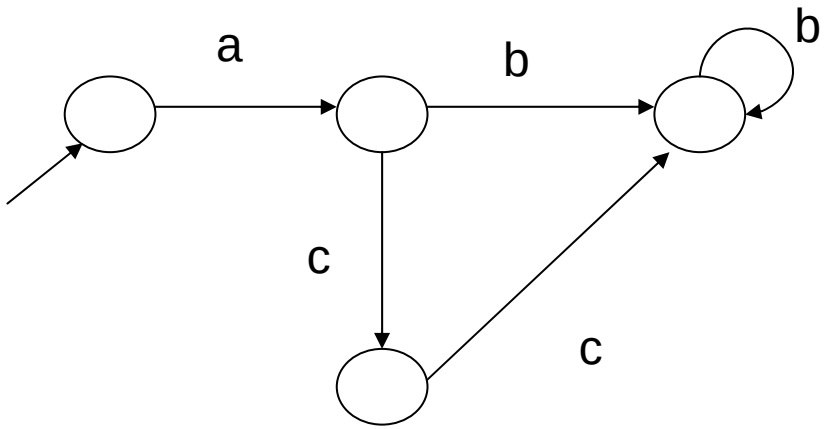


: abstraction de l'état global du système : mémoires, registres, canaux de communication….

x:=x+1

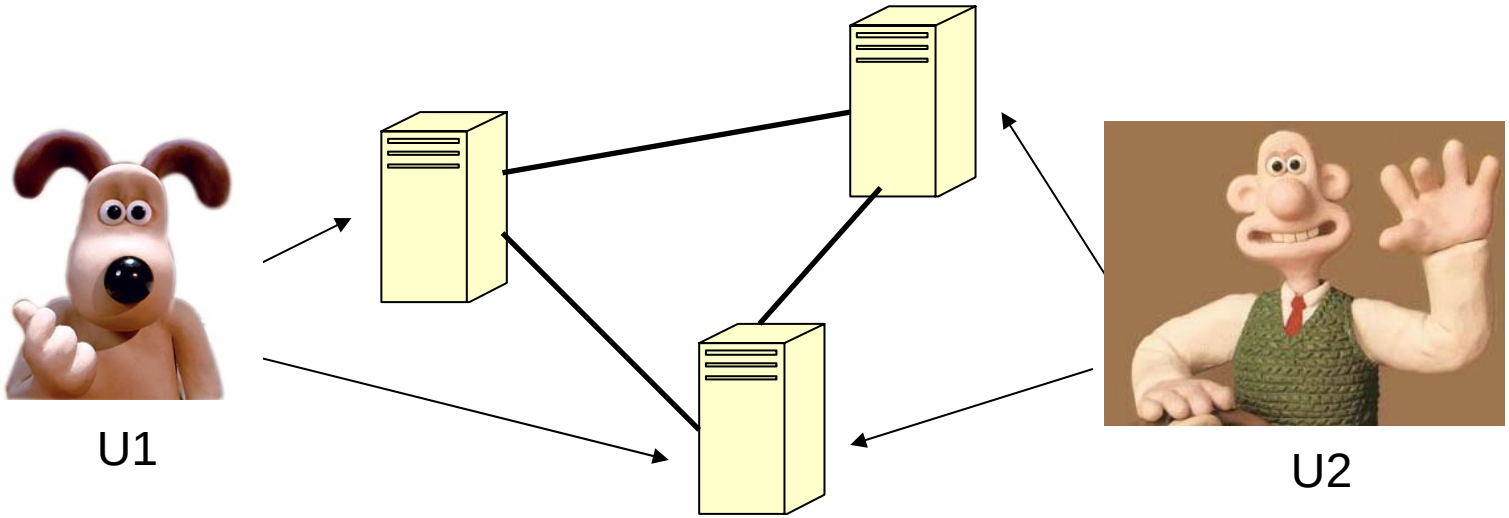
: transition – action effectuée + passage dans un autre état

Systèmes distribués



Chaque utilisateur peut :

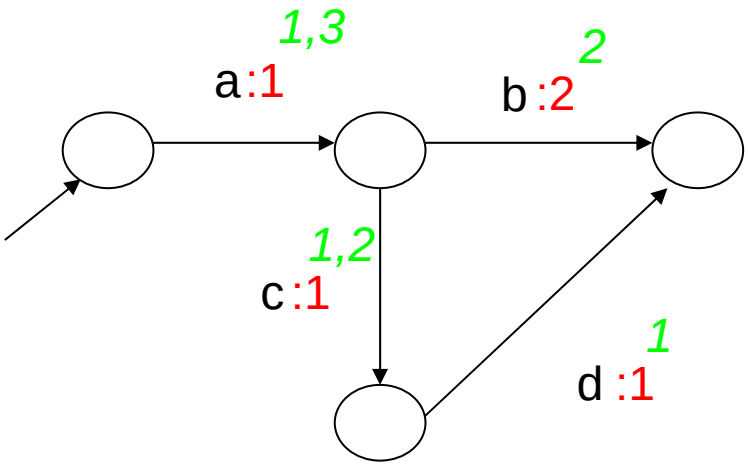
- exécuter certaines commandes
- observer certaines commandes



U1

U2

Systèmes distribués



$S=(Q, \rightarrow, \mathcal{S}, q_0)$ $P=\{U_1, \ldots, U_n\} \cup \{R_{n+1}, \ldots, R_k\}$

Q : etats du système, $q_0 \in Q$ état initial

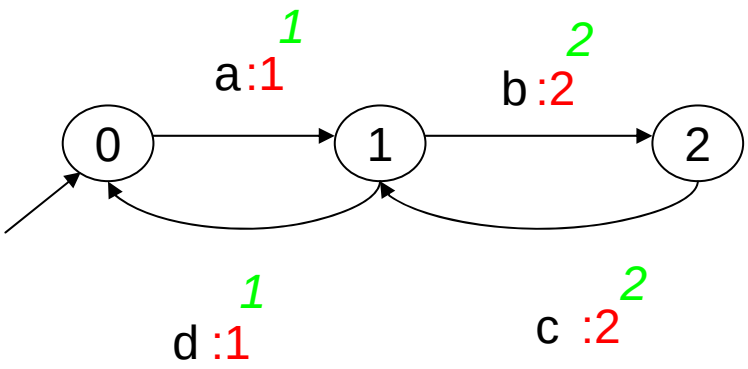
$\rightarrow \subseteq QxSxQ$ transitions du système

$\mathcal{S}$: ensemble d'actions

$Obs : S \rightarrow 2^P$ (observable par…)

$Ex : S \rightarrow P$ (Executé par…)

Exemple



a: create_account(Loic, pwd=« toto »)

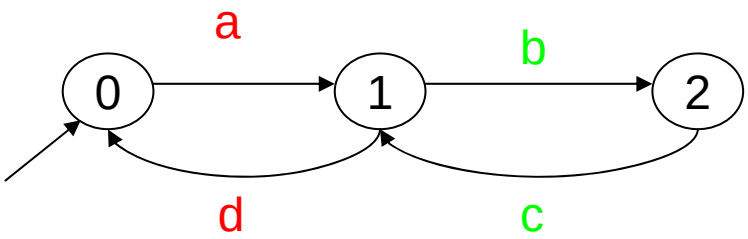
b: login(Loic, « toto »)
c: logout
d: delete_account(Loic)

$S=(Q, \rightarrow, S, q_0)$ $P=\{U_1,U_2\}$

Q ={0,1,2} , q_0= 0
S={a,b,c,d}
Obs : {a,d} $\rightarrow U_1$
{b,c} $\rightarrow U_2$
Ex : {a,d} $\rightarrow U_1$
{b,c} $\rightarrow U_2$

U_1= administrateur
U_2 = Loïc

Exemple



a: create_account(Loic, pwd=« toto »)

b: login(Loic, « toto »)
c: logout
d: delete_account(Loic)

$S=(Q, \rightarrow, \mathcal{S}, q_0)$ $P=\{U_1,U_2\}$

Q ={0,1,2} , q_0= 0
S={a,b,c,d}
Obs : {a,d} $\rightarrow U_1$
{b,c} $\rightarrow U_2$
Ex : {a,d} $\rightarrow U_1$
{b,c} $\rightarrow U_2$

U_1= administrateur
U_2 = Loïc

Langage

$S=(Q, \rightarrow, \mathcal{S}, q_0)$

$$L(S) = \{ w=a_1.a_2 \ldots a_k \in \mathcal{S}^* \mid q_0 \xrightarrow{a_1} q_{i1} \xrightarrow{a_2} q_{i2} \xrightarrow{a_3} \ldots \xrightarrow{a_k} q_{ik} \}$$

(clos par préfixe)

Une notion d'équivalence

Equivalence de langages

$S_1 = (Q_1, \rightarrow_1, S_1, q1_0)$

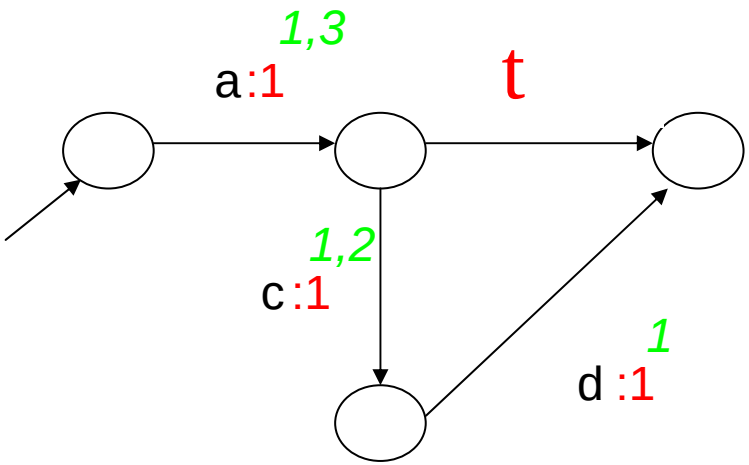
$S_2 = (Q_2, \rightarrow_2, S_2, q2_0)$

$S_1 \approx_T S_2$ si et seulement si

$$L(S_1) = L(S_2)$$

Masquage

$P_{\{a,c,d\}}(S)$



a.t = t.a = a

$L(S) = \{ a, a.b, a.c, a.c.d\}$

$L(P_{\{a,b,c\}}(S)) = \{a, a.c, a.c.d\}$

$p \overset{a}{\Longrightarrow} p'$: il existe $p_1, \ldots p_n$ tels que

$$p \xrightarrow{\tau} p_1 \xrightarrow{\tau} p_2 \xrightarrow{\tau} \ldots \xrightarrow{a} p_{n-1} \xrightarrow{\tau} p_n \xrightarrow{\tau} p'$$

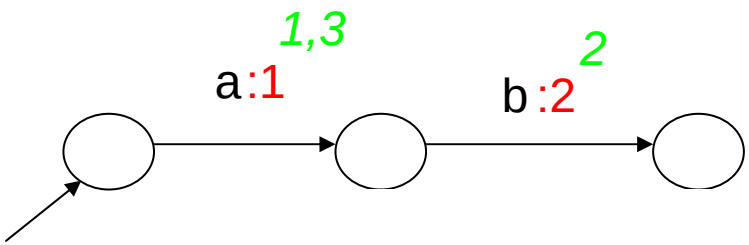
Utilité du masquage

$\Pi_{Ex^{-1}(u)}(S)$: Tout ce que *u* peut faire dans *S*

$\Pi_{Obs^{-1}(u)}(S)$: Tout ce que u peut voir dans S

Restriction

S/{c}



$L(S) = \{ a, a.b, a.c, a.c.d\}$

$L(S/\{c\}) = \{a, a.b \}$

Non Interférence

[Goguen 82]

u :| v

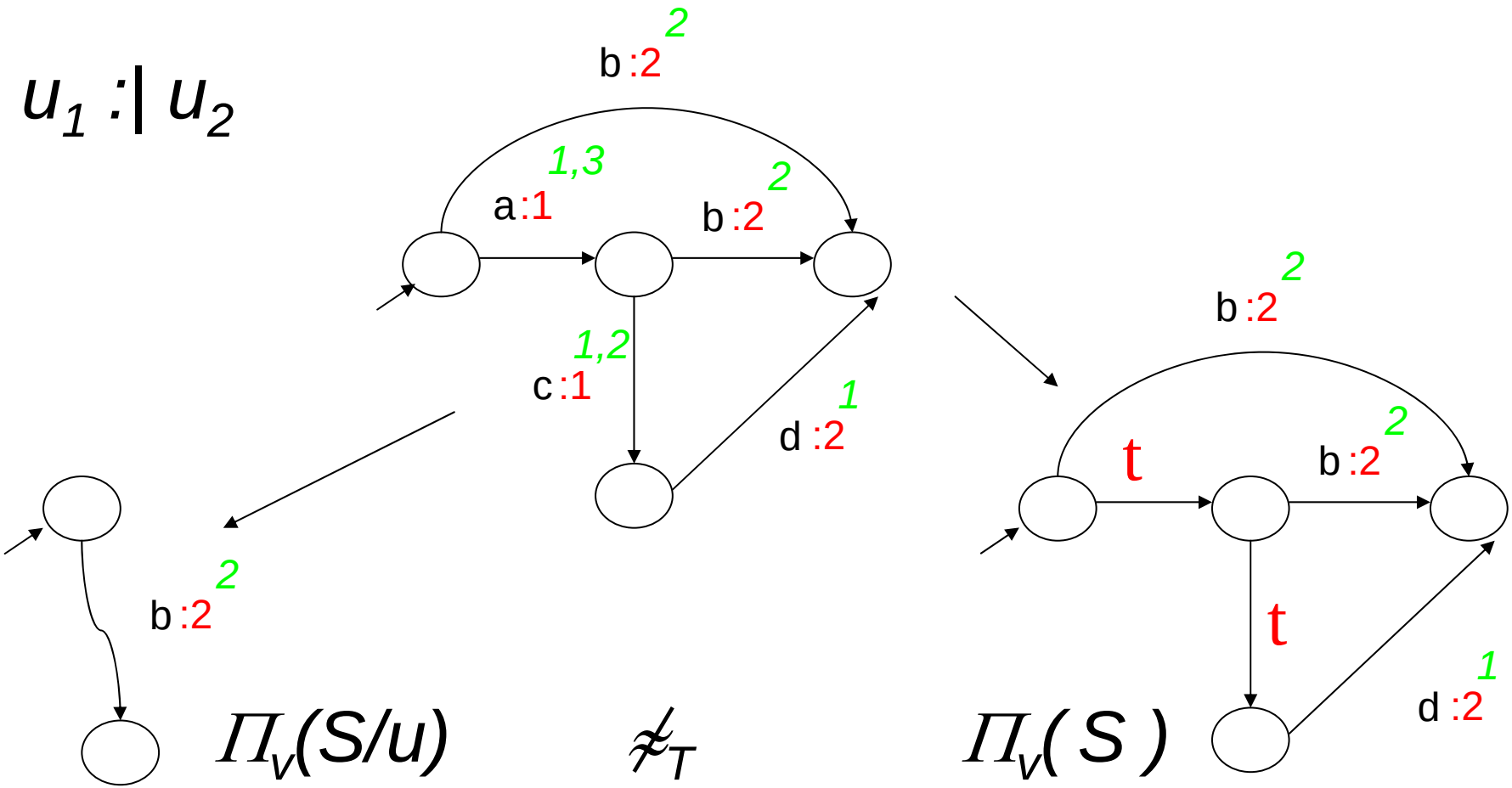
u n’interfère pas avec v dans S ssi

$$\Pi_v(S/u) \approx_T \Pi_v(S)$$

Il n’y a pas de différence dans ce que v observe,que u agisse ou non

Non Interférence

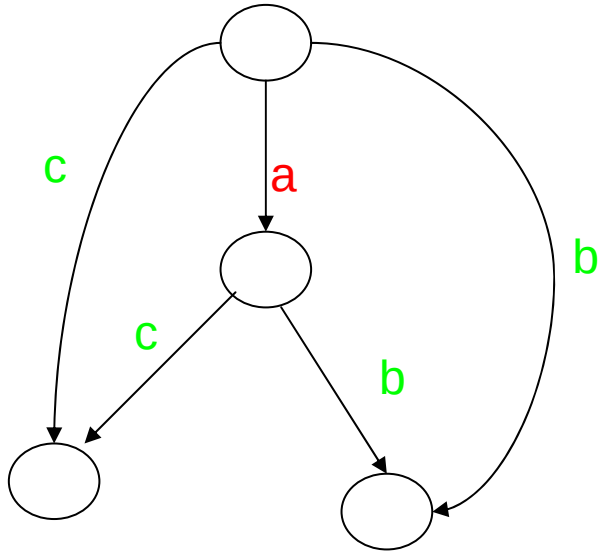
$u_1 :| u_2$



$\Pi_V(S/u)$ $\not\approx_T$ $\Pi_V(S)$

Equivalence…s

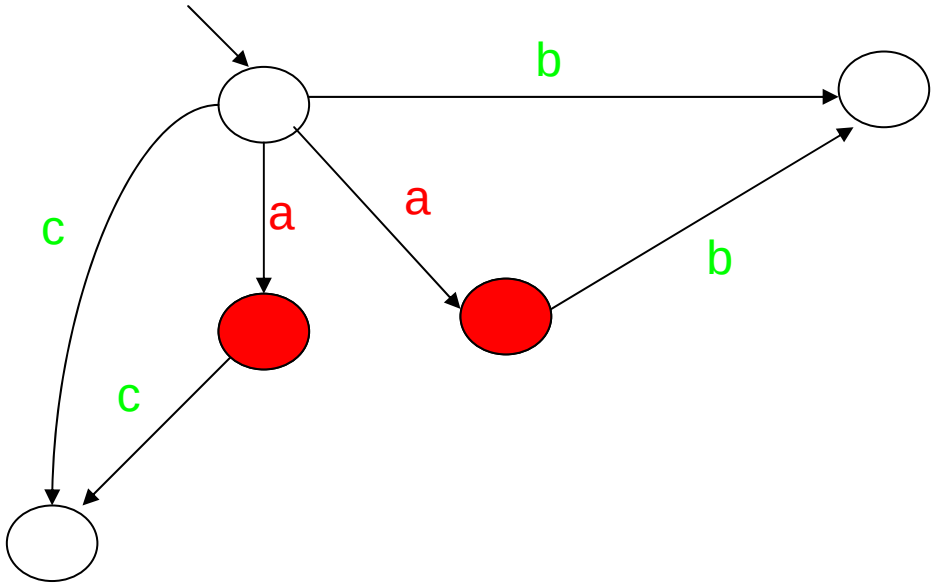
$u_1 :| u_2$



$L(S)=\{a,b,c,ab,ac\}$ $L(\Pi_{u2}(S)) = \{b,c\} = L(\Pi_{u2}(S)/ u_1)$

Equivalence…s

$u_1 :| u_2$



$L(S)=\{a,b,c,ab,ac\}$ $L(\Pi_{u2}(S)) = \{b,c\} = L(\Pi_{u2}(S)/u_1)$

Bisimulation

$R \subseteq Q1 \times Q2$ est une bisimulation faible si
$\forall (p,q)$ dans R, et tout a dans Σ,

- pour tout p' dans Q_1,

$p \overset{a}{\Longrightarrow} p'$ implique qu'il existe un $q' \in Q_2$ tel que

$q \overset{a}{\Longrightarrow} q'$ et $(p',q') \in R$,

<u>et réciproquement</u>

-
- pour tout q' dans Q_2,

$q \overset{a}{\Longrightarrow} q'$ implique qu'il existe un $p' \in Q_1$ tel que

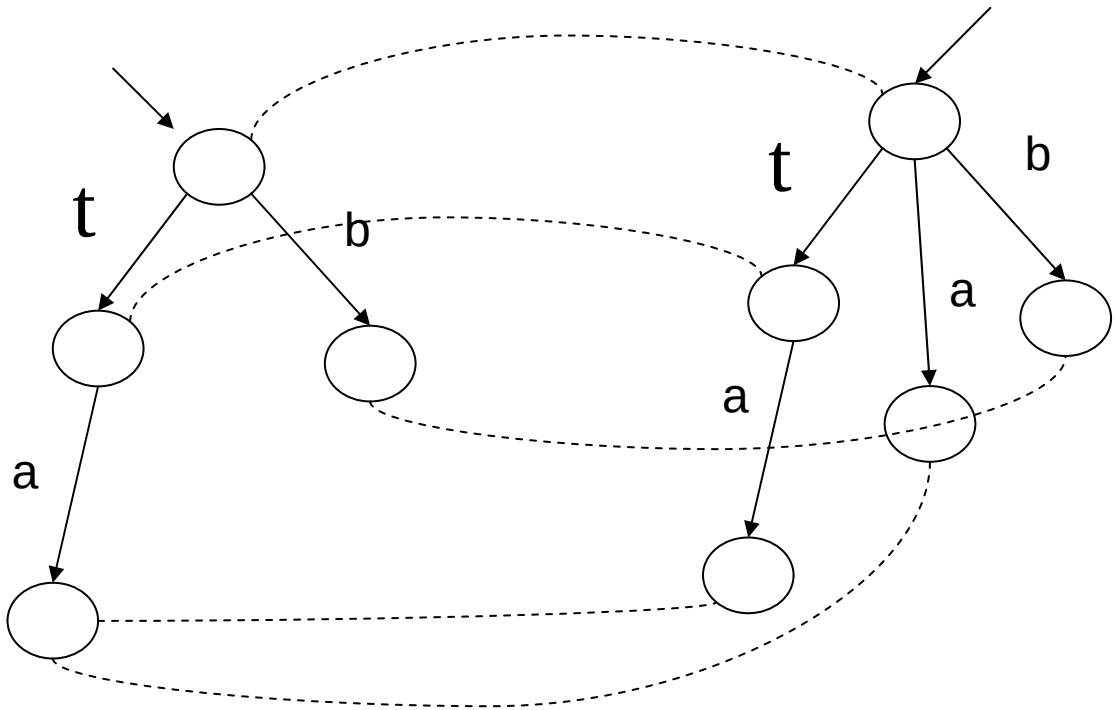
$p \overset{a}{\Longrightarrow} p'$ et $(p',q') \in R$.

Bisimulation

S_1 et S_2 sont (faiblement) bisimilaires ssi

- Il existe une relation de bisimulation $R \subseteq Q_1 \times Q_2$
- $q1_0$ R $q2_0$

$$S_1 \approx_B S_2$$



Nb : $S_1 \approx_B S_2$

$\Rightarrow S_1 \approx_T S_2$

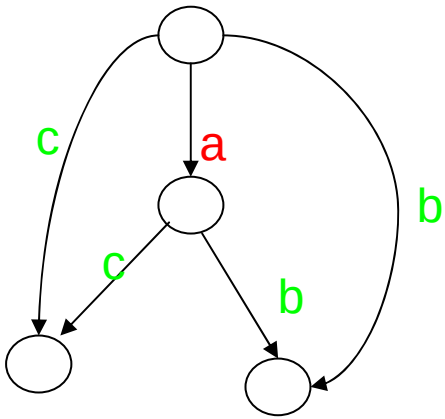
Generalisation

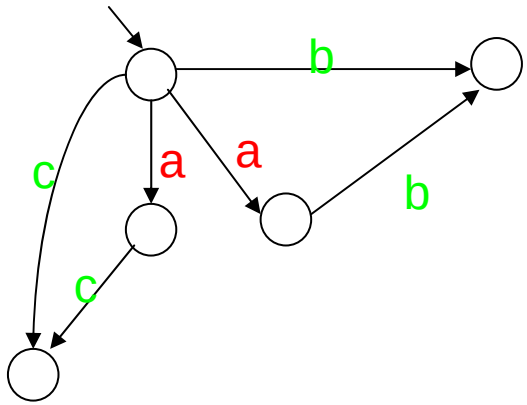
- S est NNI ssi *(non deterministic non interference)*

$$\Pi_V(S/u) \approx_T \Pi_V(S)$$

- S est BNNI ssi

$$\Pi_V(S/u) \approx_B \Pi_V(S)$$





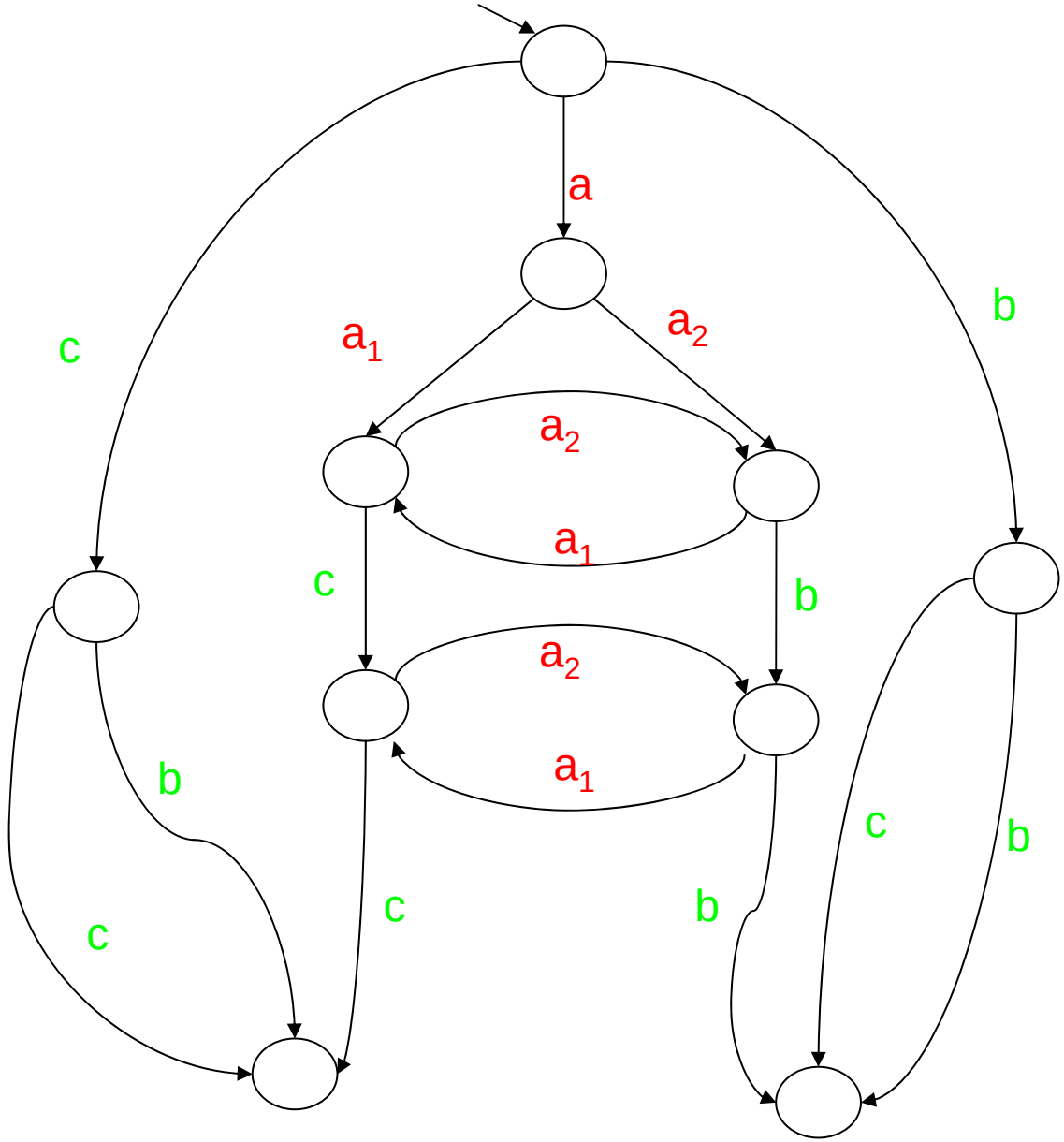
NDC

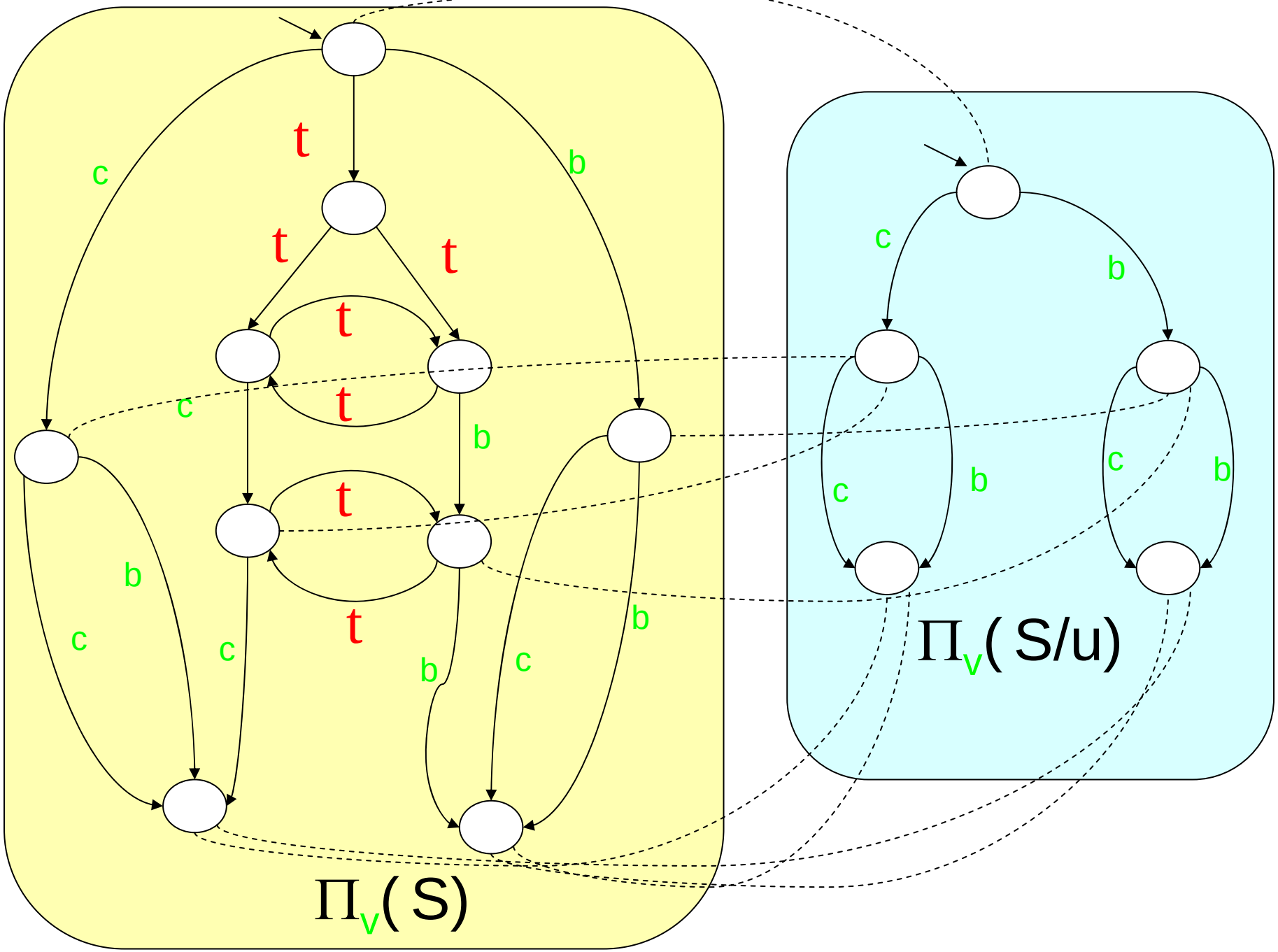
- Non disclosure on Composition

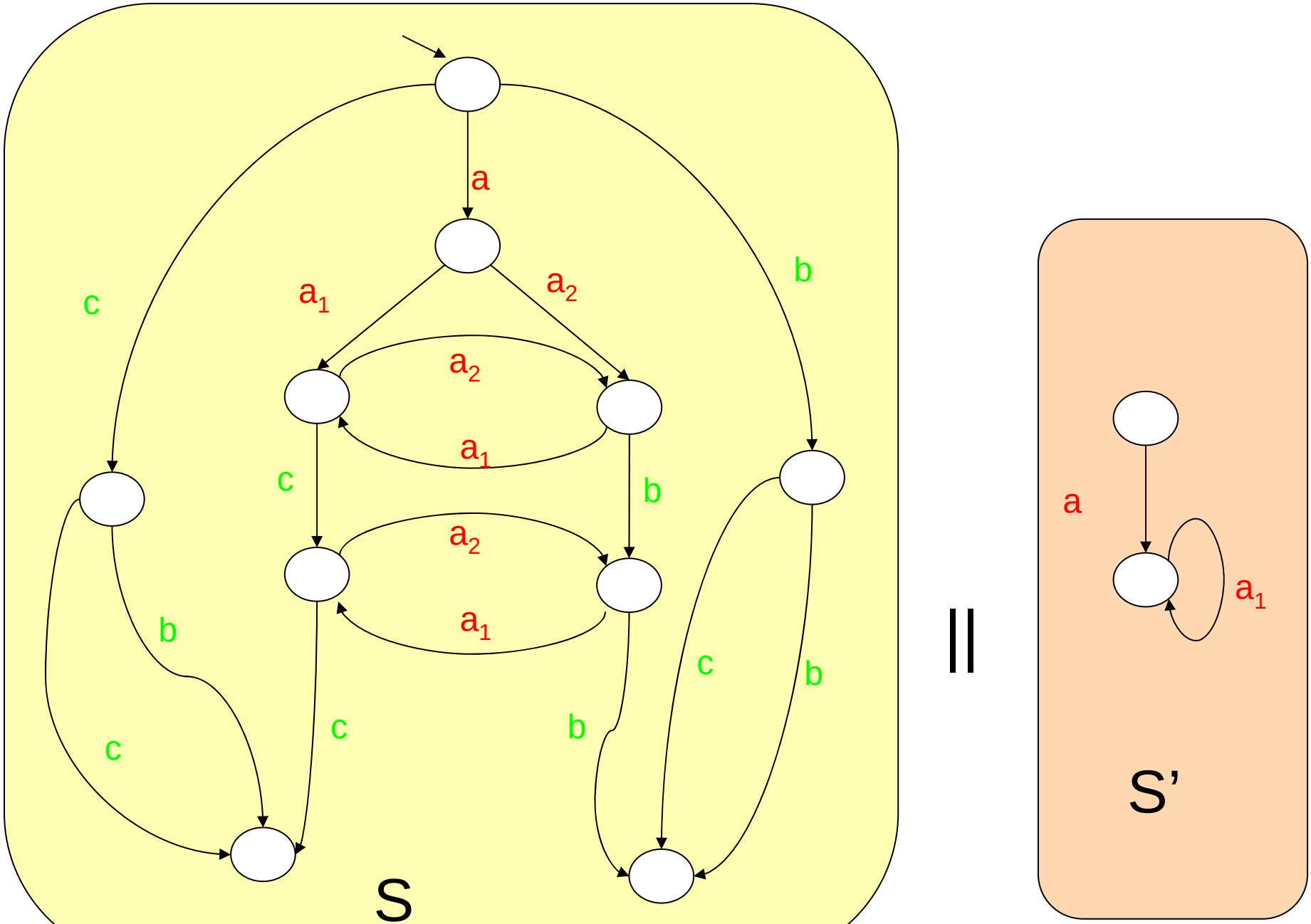
S est NDC ssi pour tout S’

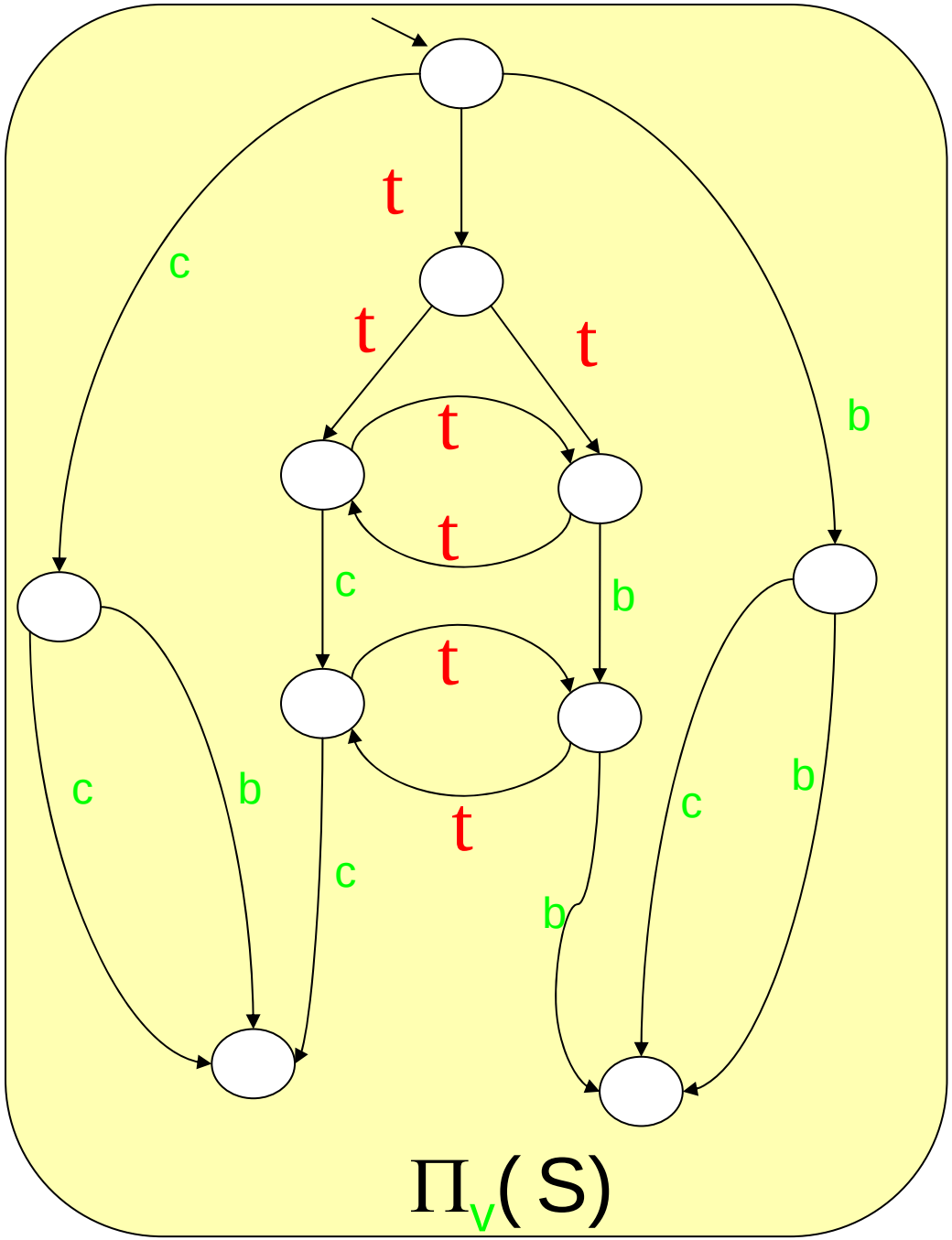
$$\Pi_v(S \,||\, S') \approx \Pi_v(S)$$

S

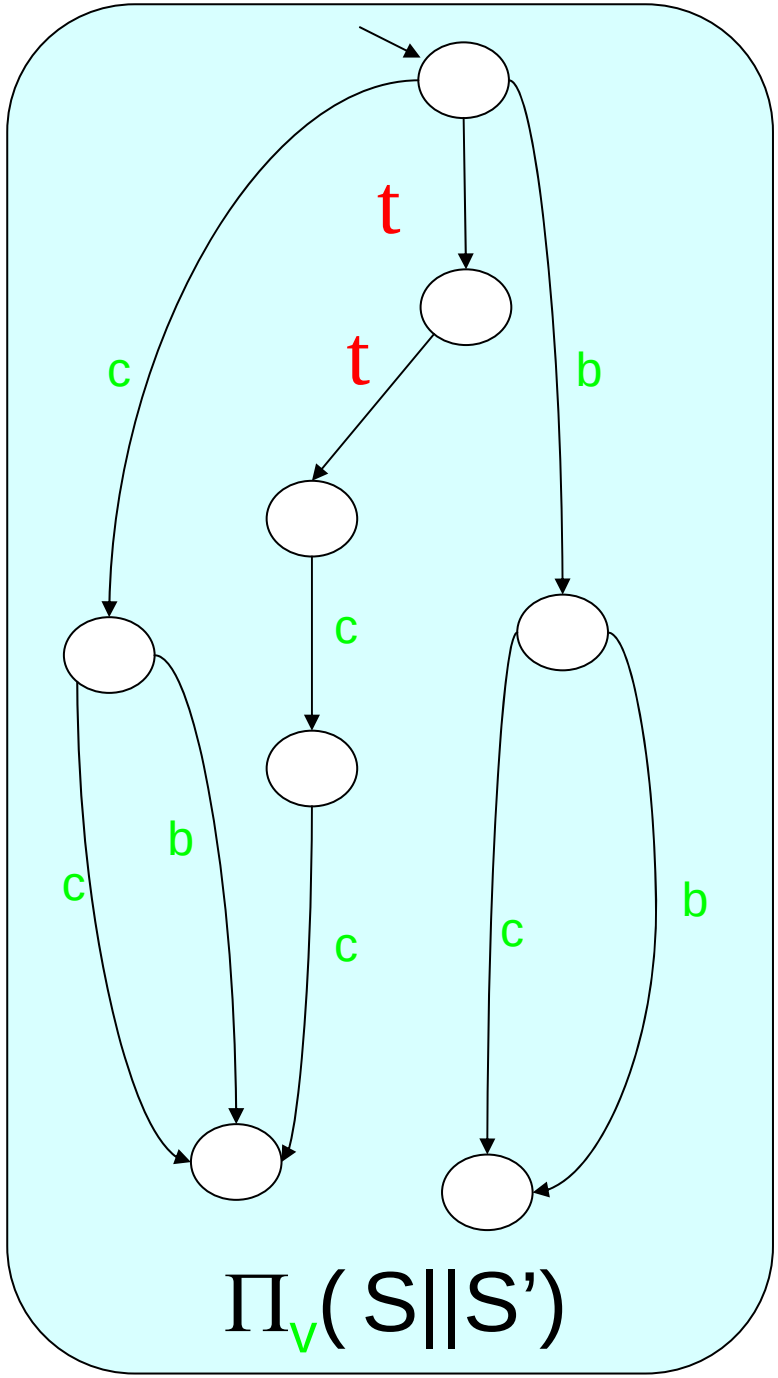



t
t
t
t
t
t
c
b
c
b
b
c
c
b
c
b
b
c
c
b
$\Pi_v(S)$
$\Pi_v(S/u)$


a
a1
a2
b
c
a2
a1
c
b
a2
a1
b
c
c
b
c
b
S
||
a
a1
S'



$\Pi_v(S)$



$\Pi_v(S||S')$

Non Interférence, variations

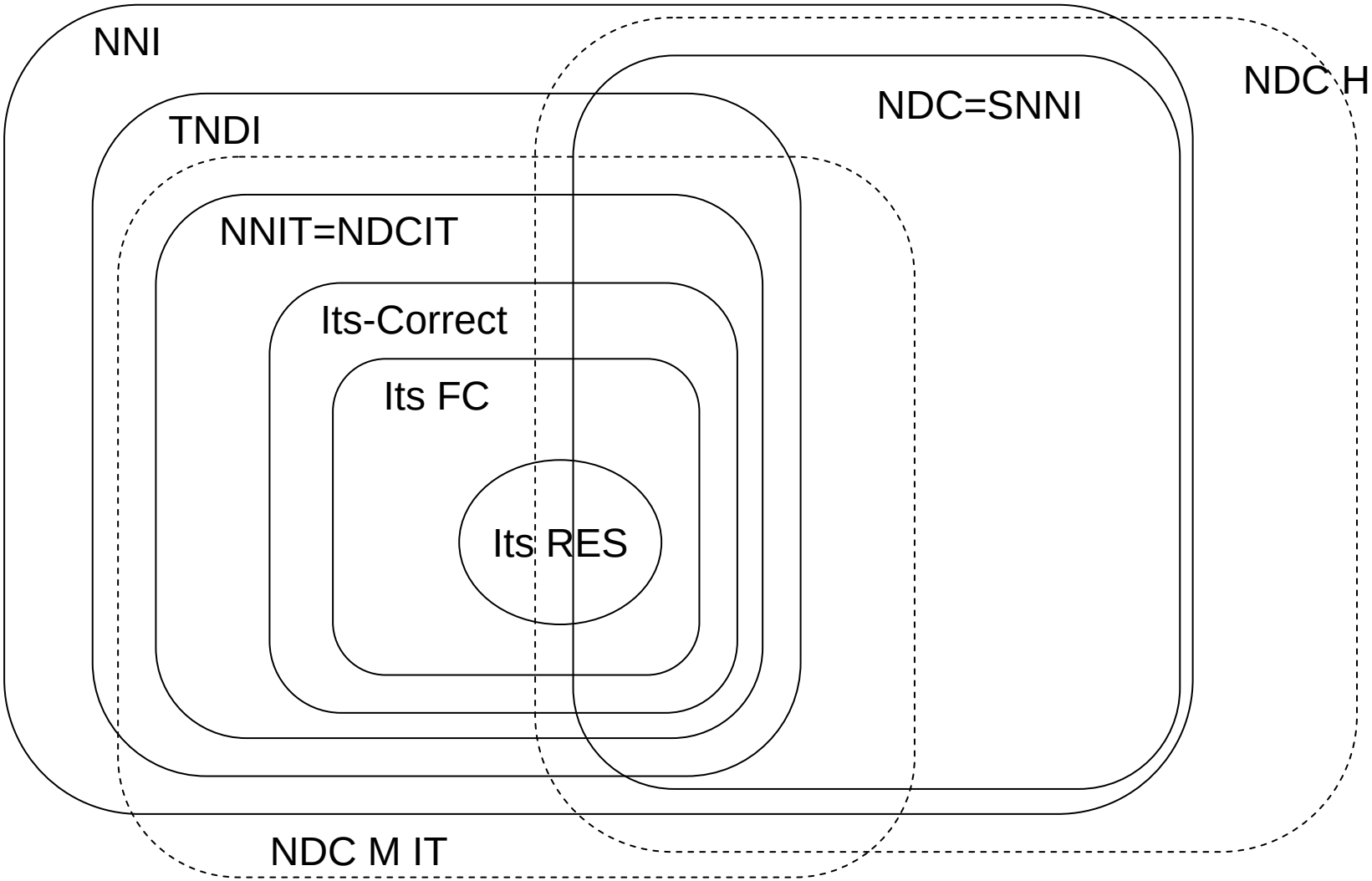
- Modèles pour le système (Automates, CSP, …)
- Semantique de ||
- Notion d'équivalence $\approx$
  - Bisimulation, trace, testing, …. + Input/output
  - Observation sur les transitions/états

  [Lowe][Ryan]

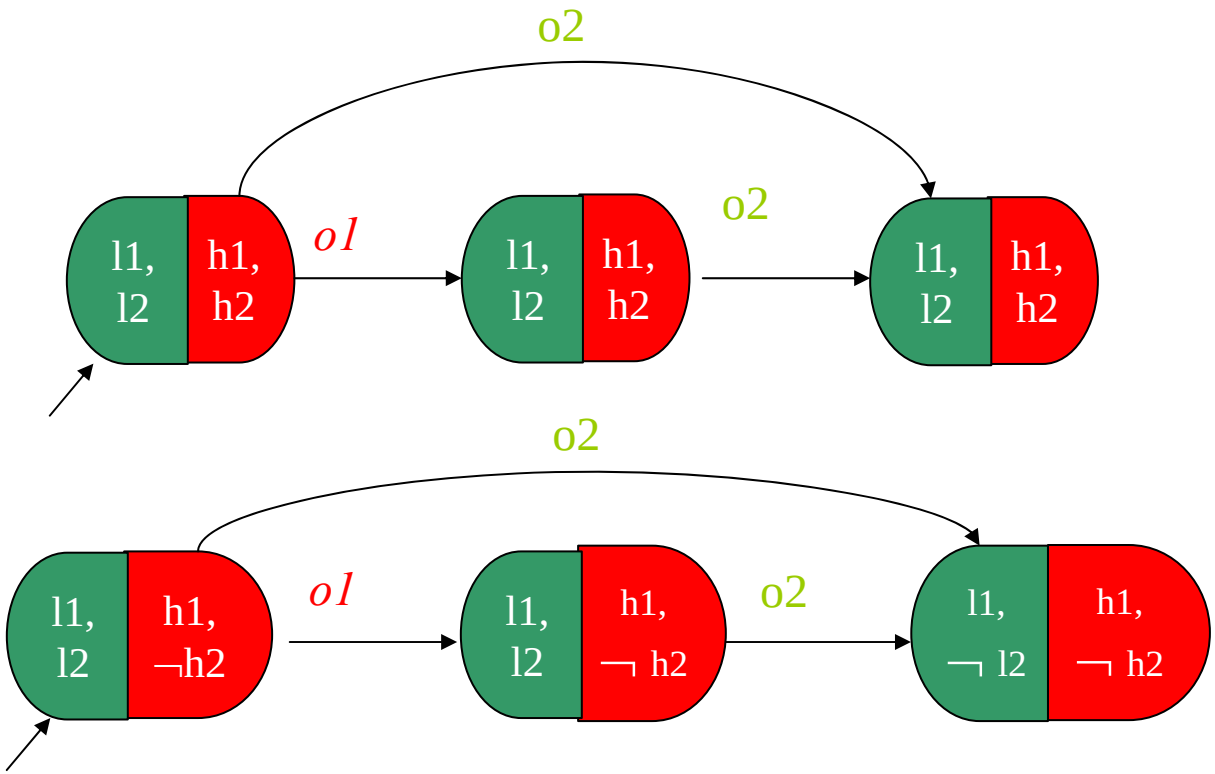
  [Focardi&Gorrieri 00]

  …

Bestiaire des interférences



Interférence entre variables



Non interference par typage

[Volpano & Smith]

Typage de programes:

```
p::= e | c
e ::= x | l | n | e+e’
        | e – e’ | e=e’ | e<e’

C::= e:= e’ | c; c’
     | if e then c else c’
     | while e do c
     | letvar x:=e in c
     | try x= e op e’ in c
```

Quelques règles

$$\frac{e : \tau,\ c : \tau\ ,\ c' : \tau}{\text{If e then c else c'}}$$

Un programme bien typé est non-interférent

Rejeter des programmes comme :

```
If h then l := true else l:=false
```

Idem pour un pseudo langage avec threads [Volpano98]
Idem avec non-interference probabiliste [Volpano99]

[Boudol &Catellani] langage concurrent

Mais: L'inteférence dépend de l'accessibilité de certaines instructions (undecidable)

Solution: L'approche pessimiste :

- utiliser une sémantique « gros grain »,
- pas de faux négatifs, mais des faux positifs…

```
While e>e' do
      e':= f(e,e')
      If e' mod y = 0 then e'':=0
Done
If e'' = 0 then l:=h else l:= l'
```

- Une politique de sécurité est un ensemble d'assertions de non-interférence *[Goguen82]*

- Contrôle de canal :

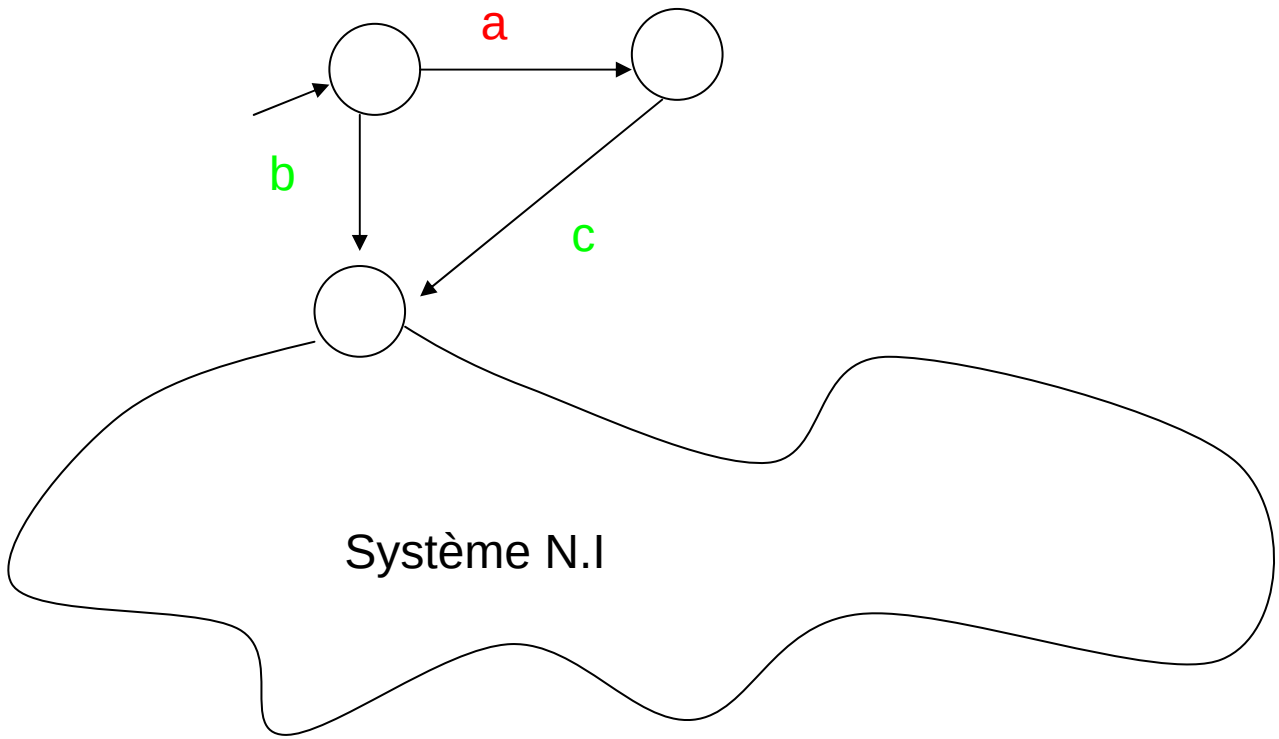
U et V ne communiquent que par le canal A
(vu comme un sous ensemble d'actions)

$$\neg A,u :| v \text{ et } \neg A,v :| u$$

$\equiv$ pas de canal caché entre u et v ?

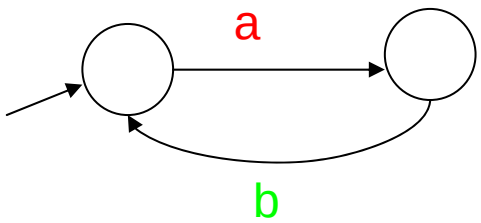
Inconvénients de la N.I.

- 1 bit suffit pour être non interférent



Inconvénients de la N.I.

- La répétition du même phénomène est aussi une interférence



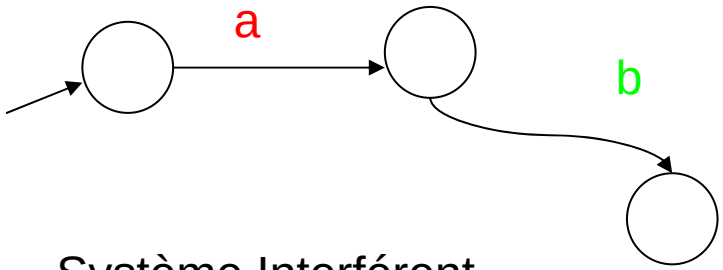
Système Interférent

(Pas toujours suffisant pour coder de l'information)

Inconvénients de la N.I.

- Interférer puis plus rien

est aussi une interférence

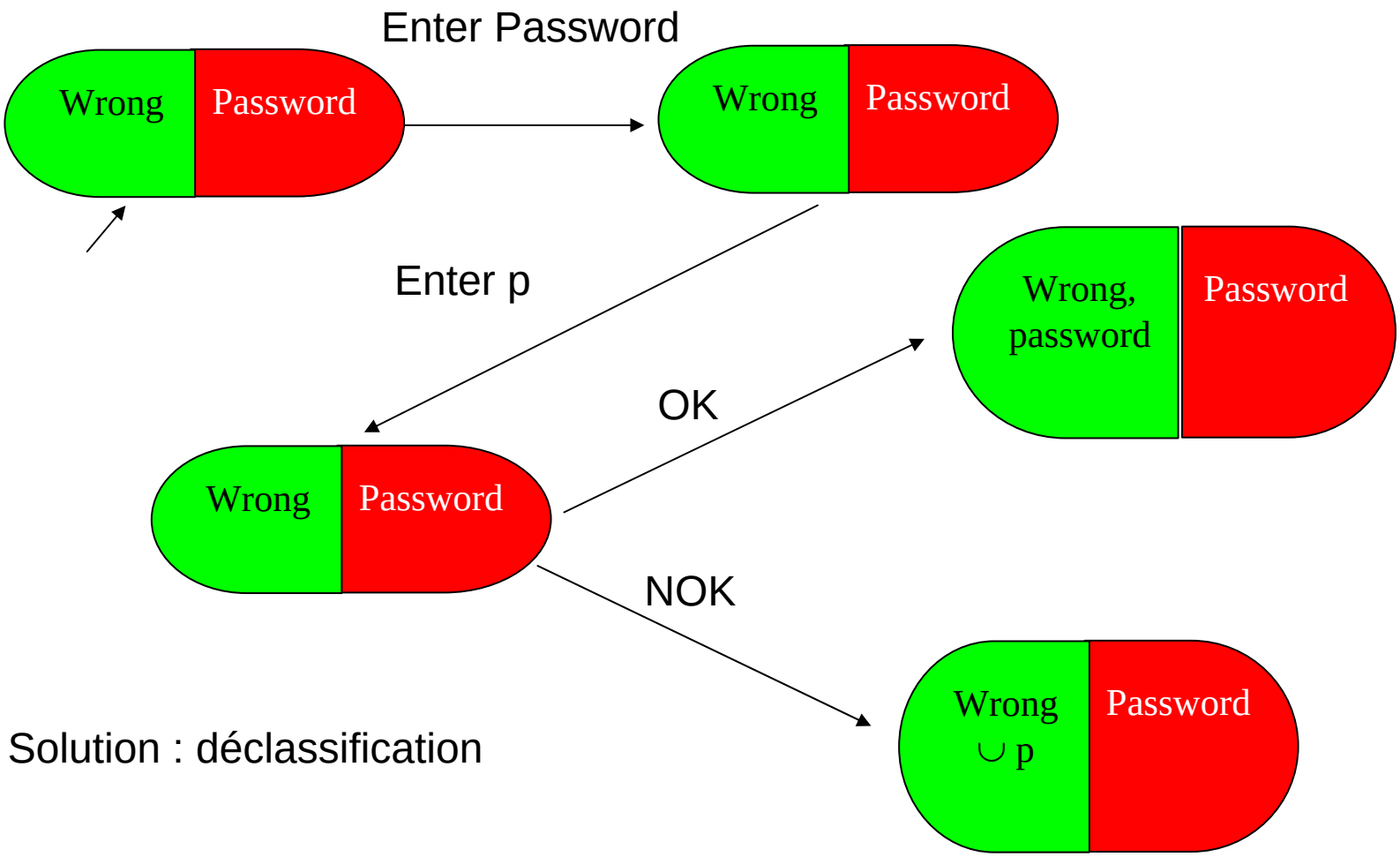


Système Interférent

Implicitement, on peut toujours reproduire l'interférence

mais est-ce toujours vrai ?

Inconvénients de la N.I.



Solution : déclassification

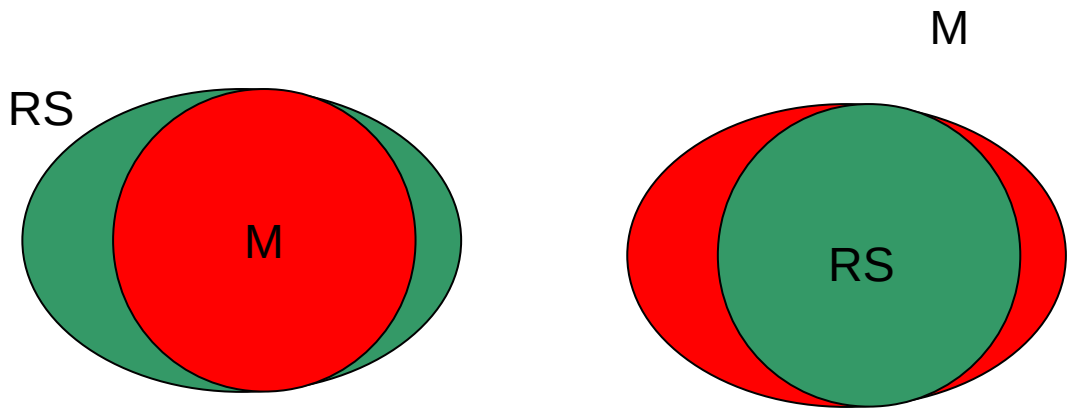
Inconvénients de la N.I.

- La relation d'équivalence choisie doit être décidable
  - Restriction sur les modèles considérés

    automates communicants,

    traces de Mazurkiewicz,

    ….



Canaux cachés

- Storage Channels
  - Utilisent une resource du système

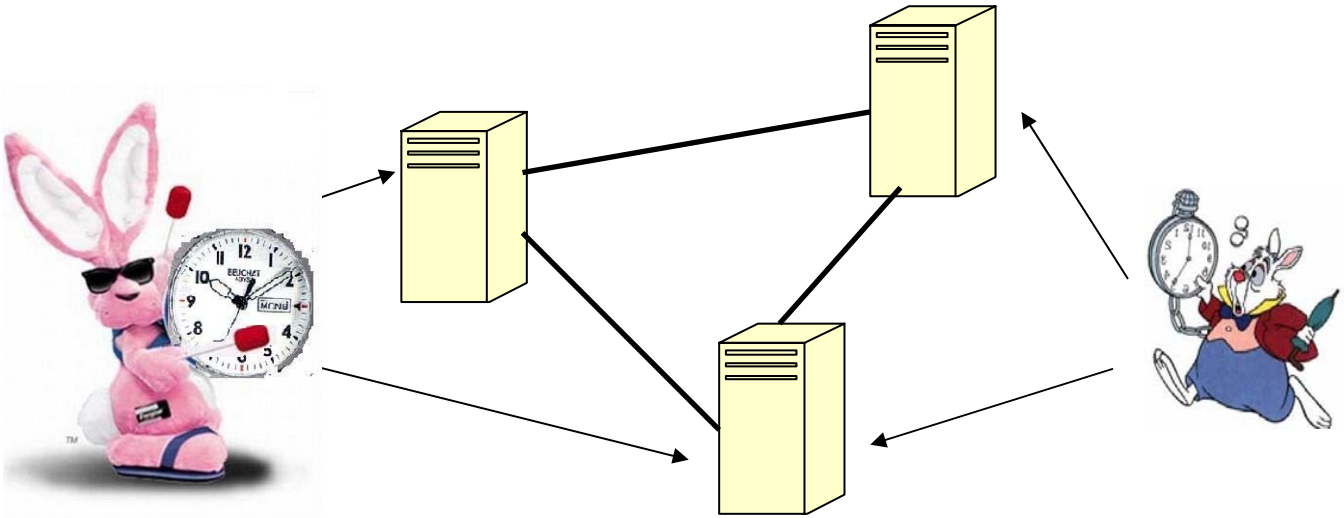
    Pour stocker des bits d'information

(exemple : Disk full)

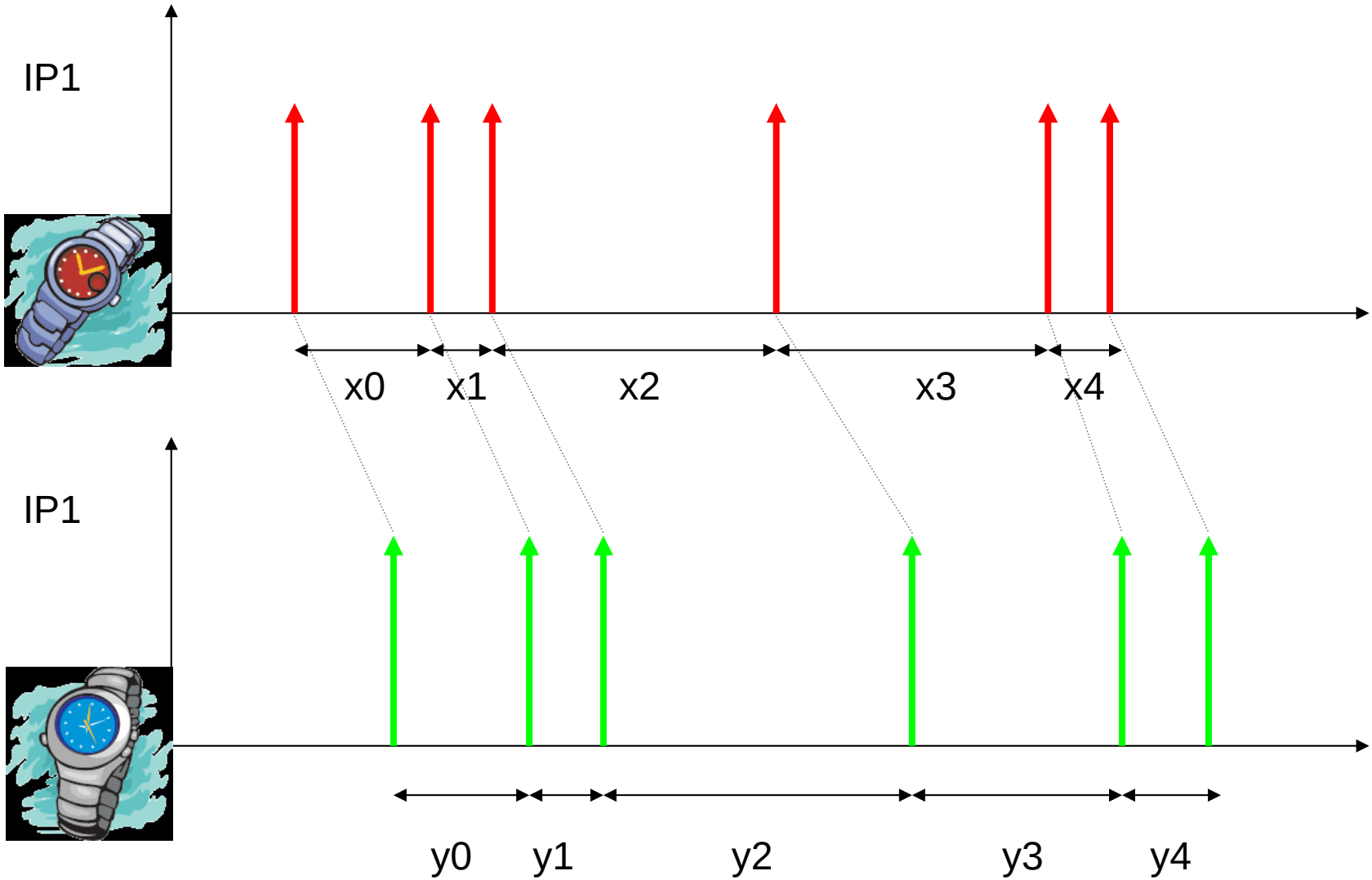
- Timing Channel
  - Modulation dans le temps de l'utilisation de

  L'utilisation du système

Timing channels



Timing Channels



CC vs Interférence

Canal caché = possibilité de transmettre un message de taille arbitraire.

CC= interférence répétée ?
Dans un CC, u et v se sont mis d'accord sur un codage.

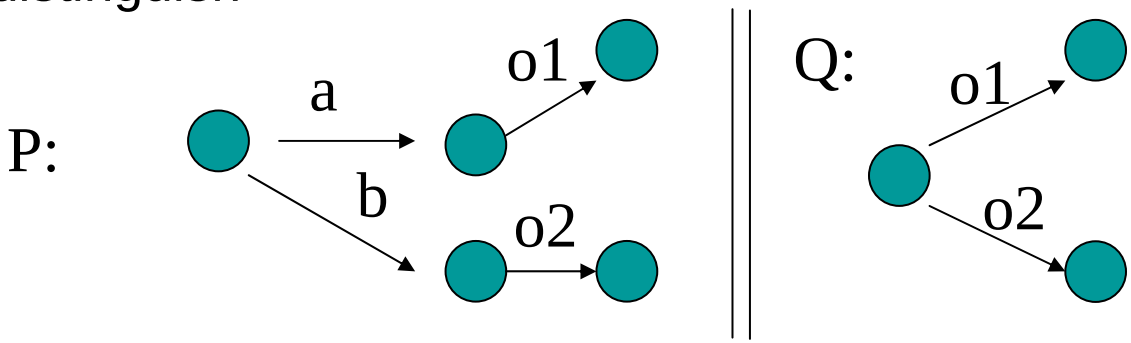
Canaux cachés

- Storage channels :
  - Trouver des flots d'information indirects
    - Shared ressources matrices
    - CSP, automates, théorie des jeux, …
- Timing channels
  - Pour un medium de communication choisi
  - Calcul de la quantité d'information subliminale au dessus du canal
    - Théorie de l'information

Quantification

[Lowe]

- IQF(S) = maximal number of runs of S that V can distinguish



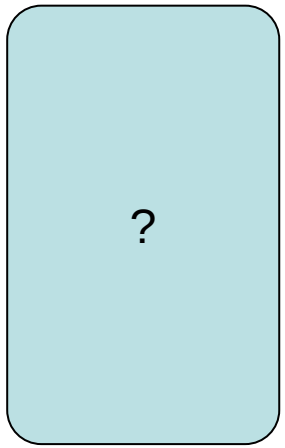
$IFQ_t(S)$ = same thing within duration t

C(S) = lim n-> inf (IFQt(S) / t)

système NDC avec capacité 1 !

Information theory

2 variables aléaroitres
Valeur = { A, Q, K }
Couleur = {R, B}



H(X) = nb questions pour connaitre X (en bits)

$$H(x) = -\sum_{x \in X} p(x) \log_2 p(x)$$

Théorie de l’information

- Quand je connais la couleur d’1 carte,
- J’ai déjà un peu d’information sur sa valeur

- Information mutuelle

- $I(X;Y) = H(X) + H(Y) – H(X,Y)$

Theorie de l'information

[Moskowitz 94]

- Relation entre entrées et sorties sur un canal choisi (variables $\mathcal{X}$, $\mathcal{Y}$)
- Canaux discrets sans mémoire (Timing Channels)

Information mutuelle
Entre une entrée et une sortie
Sur le canal

$$I_t(X;Y) = \frac{H(X) - H(X|\ Y)}{E(T)}$$

Information mutuelle
par unité de temps

Durée moyenne entre
Entrée et sortie

Le critère du « petit message »

- Interférence oui, mais…

Canal dans lequel chaque utilisation permet de transférer 1 bit
Mais le neme transfert d'information prend 2^n UT

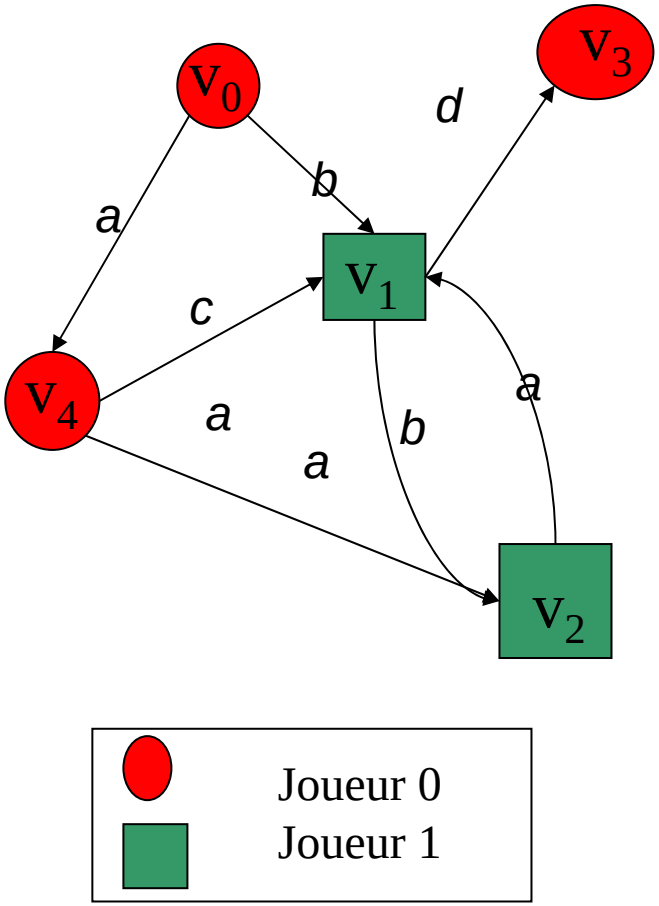
$$E(T) = \lim_{n\to\infty} \frac{1}{n} \sum_{1..n} 2^n$$

$$I_t(X;Y) = 0$$

Pourtant, essayons de transmettre 4 bits …

… tout message fini se transmet en un temps fini

Canaux cachés & Jeux



Arène :

Sommets V

Arcs E

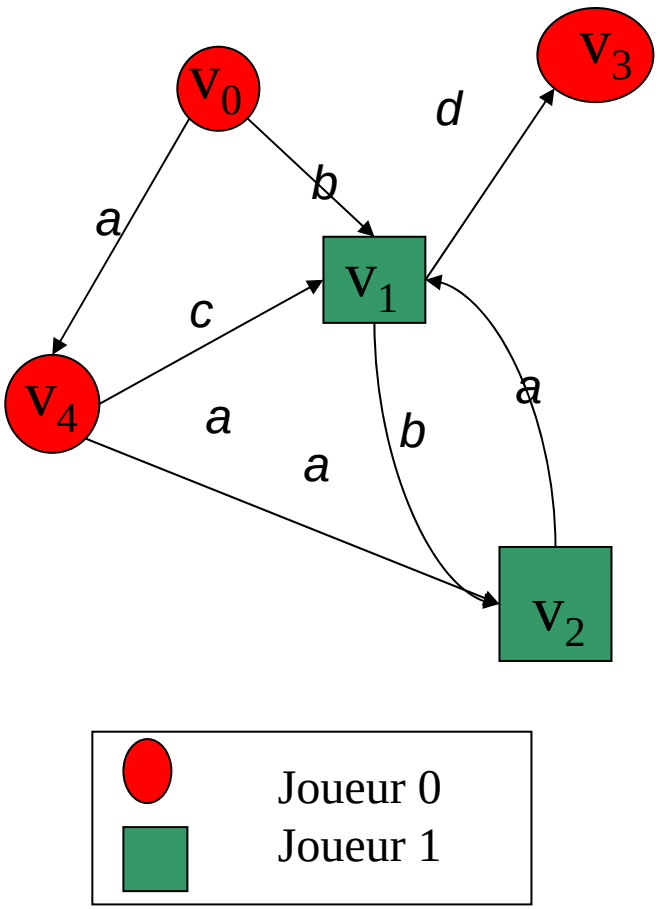
2 joueurs : $\sigma = \{ 0,1 \}$

Conditions de gain:

$Win \in \mathcal{P}(V)$ *(Buchi Game)*

$Win \subseteq \mathcal{P}(V)$ *(Muller Game)*

...



Partie :

finie :

$v = v_{i1}.v_{i2} \ldots v_{ik}$ avec v_{ik} sommet puits

infinie :

$w = v_{j1}.v_{j2} \ldots \in V^{\omega}$

$Inf(w) = \{v \mid \} \forall i, \exists j > i , v_j = v\}$

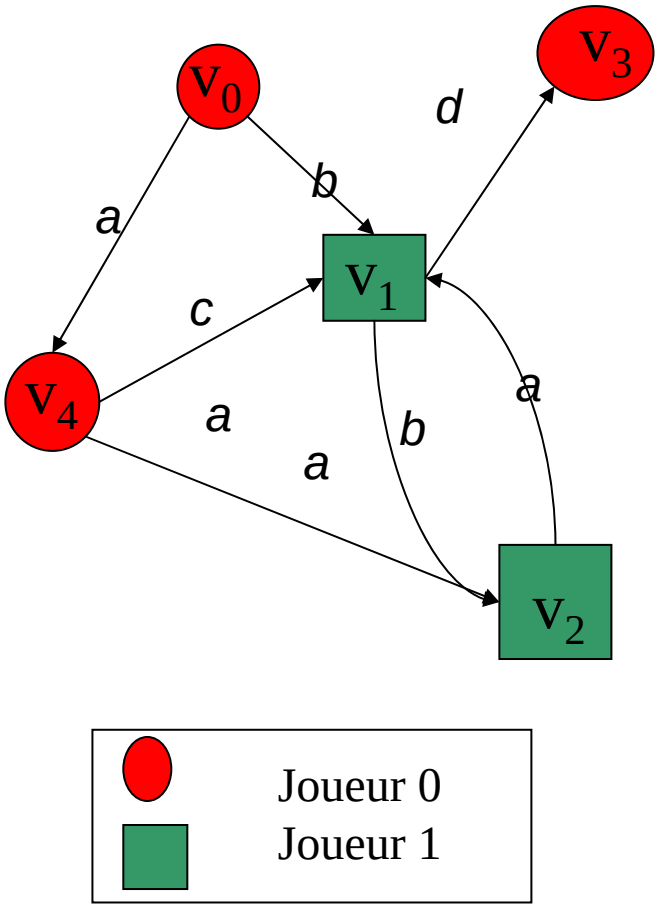
Player *0* wins a play v iff

$v = n_{i1}.n_{i1} \ldots n_{ik}$ finite and P_1's turn

or $w = n_{j1}.n_{j2} \ldots \in V^{\omega}$

and $Inf(w) \cap Win \neq \emptyset$ *(Büchi)*

$Inf(w) \in Win$ *(Muller)*



Fonction $f : V' \subseteq V \rightarrow \mathcal{P}(E)$

Win = $\{v_1, v_2\}$

Strategy pour P_1 :

$v_1 \rightarrow \{ (v_1, v_2) \}$

$v_2 \rightarrow \{ (v_2, v_1) \}$

Winning subset for P_σ :

subset for which a strategy for P_σ exists

Canaux cachés

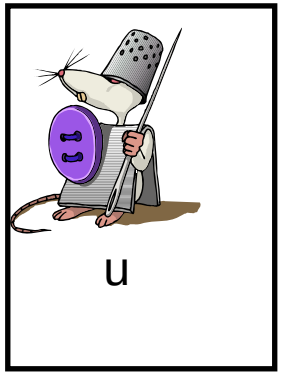
Hypothèse :

Pour transmettre un message de taille arbitraire, il faut répéter certains comportements:

Les canaux cachés vont apparaitre dans des cycles ou des composantes connexes des spécifications

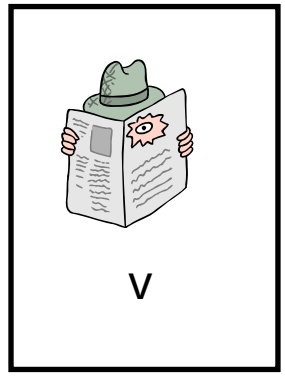
Regarder un canal caché entre u et v comme un jeu dans lequel la paire (u,v) gagne si elle Peut transmettre un message de taille arbitraire

- rester dans une composante connexe
- Tout en transmettant de l'information



u

Joueur 1



v

message

Arène

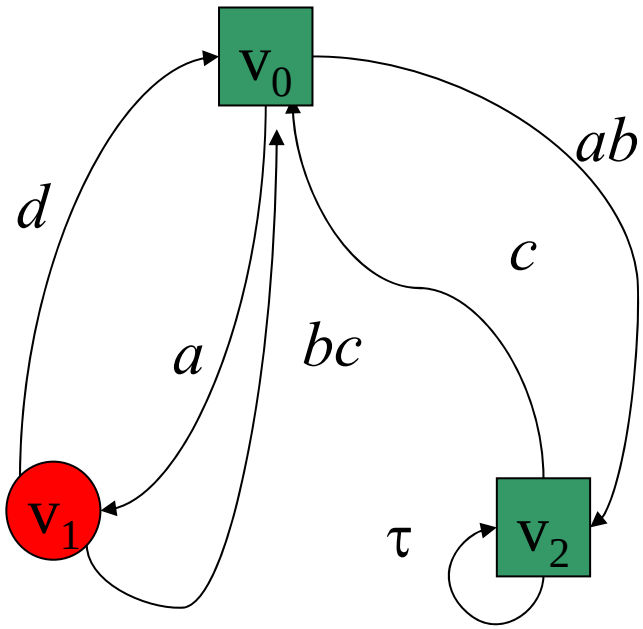
message

Strategie

Decodage

Joueur 0

Reste du Systeme

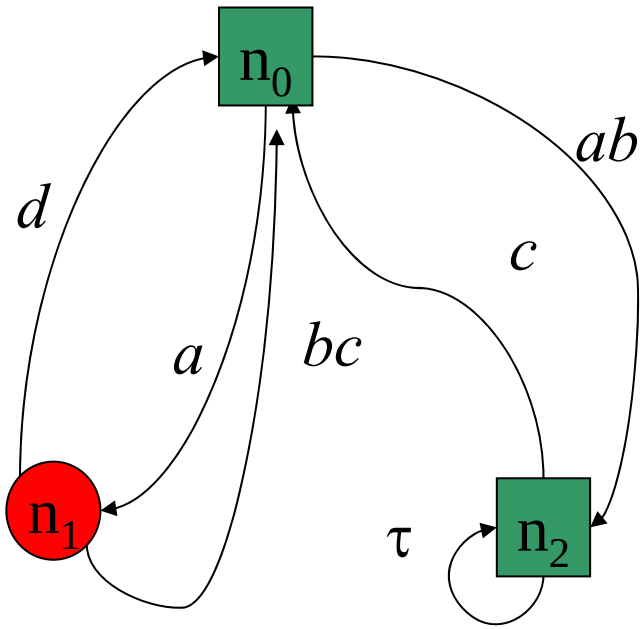


v_1 Joueur 0 : ce que P-u fait

v_0 Joueur 1 : ce que u fait

ab ce que v observe à chaque coup

Ambiguité



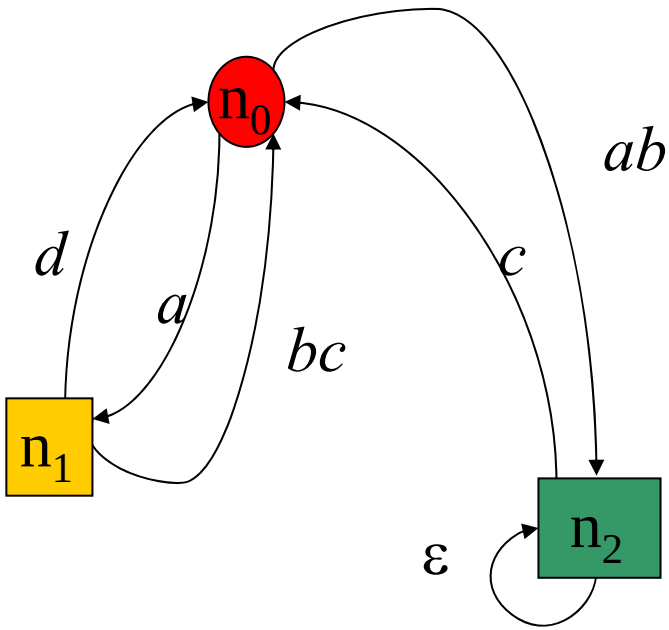
D={n_0 ; n_1 ; n_2 }
A(D,v_0)
A(D,v_1)
A(D,v_2)

D strongly connected component

A(D,n_i) ssi :
n_i sommet de P- u
ou
n_i sommet de *u* tel que
$\forall\, t_1,t_2,\ t_1=(n,b,n_1')\ t_2=(n,b,n_1')$

v ne peut pas différencier les chemins débutant par t_1 *de ceux débutant par* t_2

Codage d'information



Joueur 0 : protocol
Joueur 1 : u
Joueur 1 : u + codage

$D=\{n_0 ; n_1 ; n_2\}$

$A(D,n_0)$

(pas un choix de u)

$\lceil A(D,n_1)$

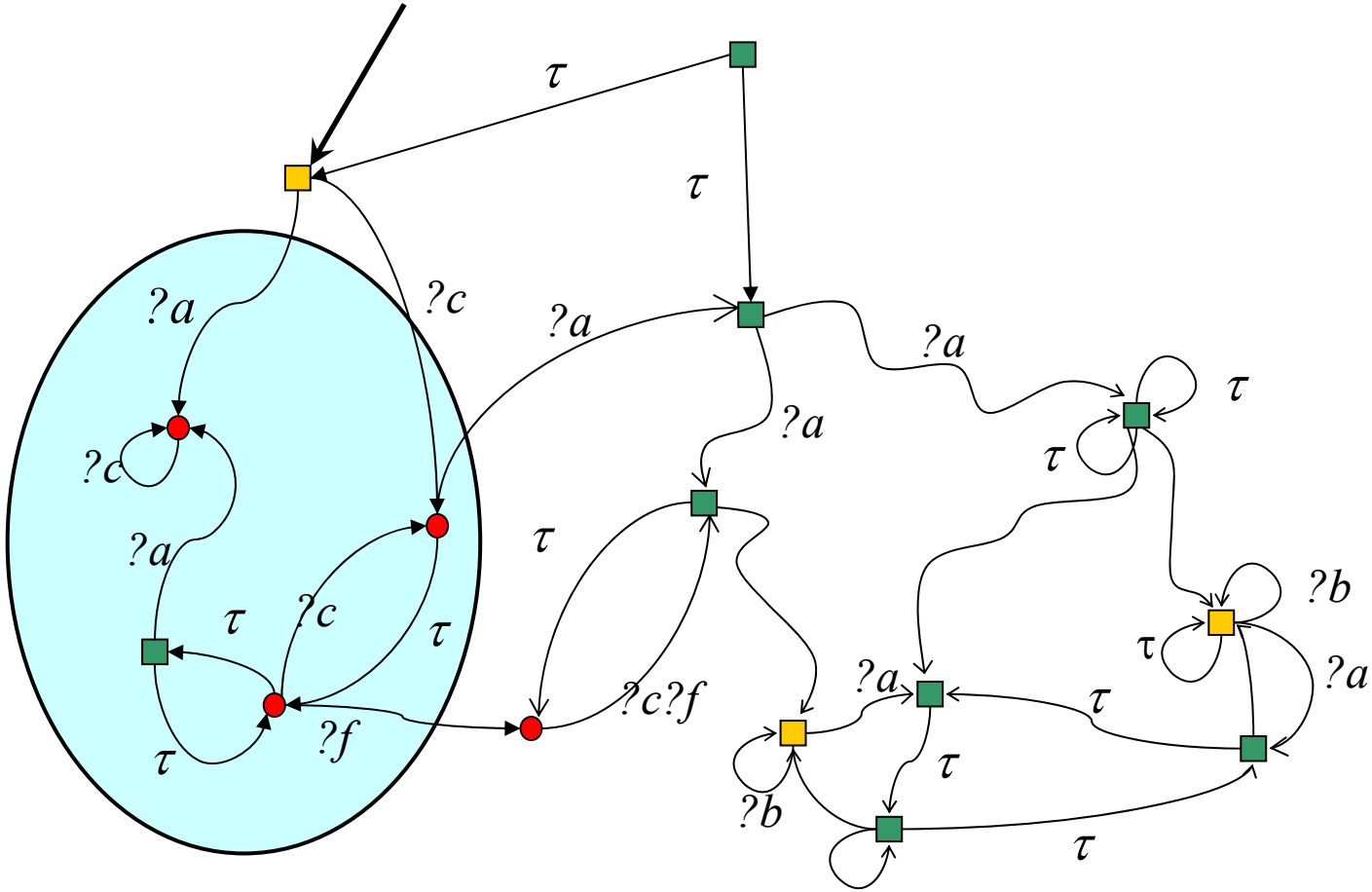
(deux choix distinguables par v)

$A(D,n_2)$

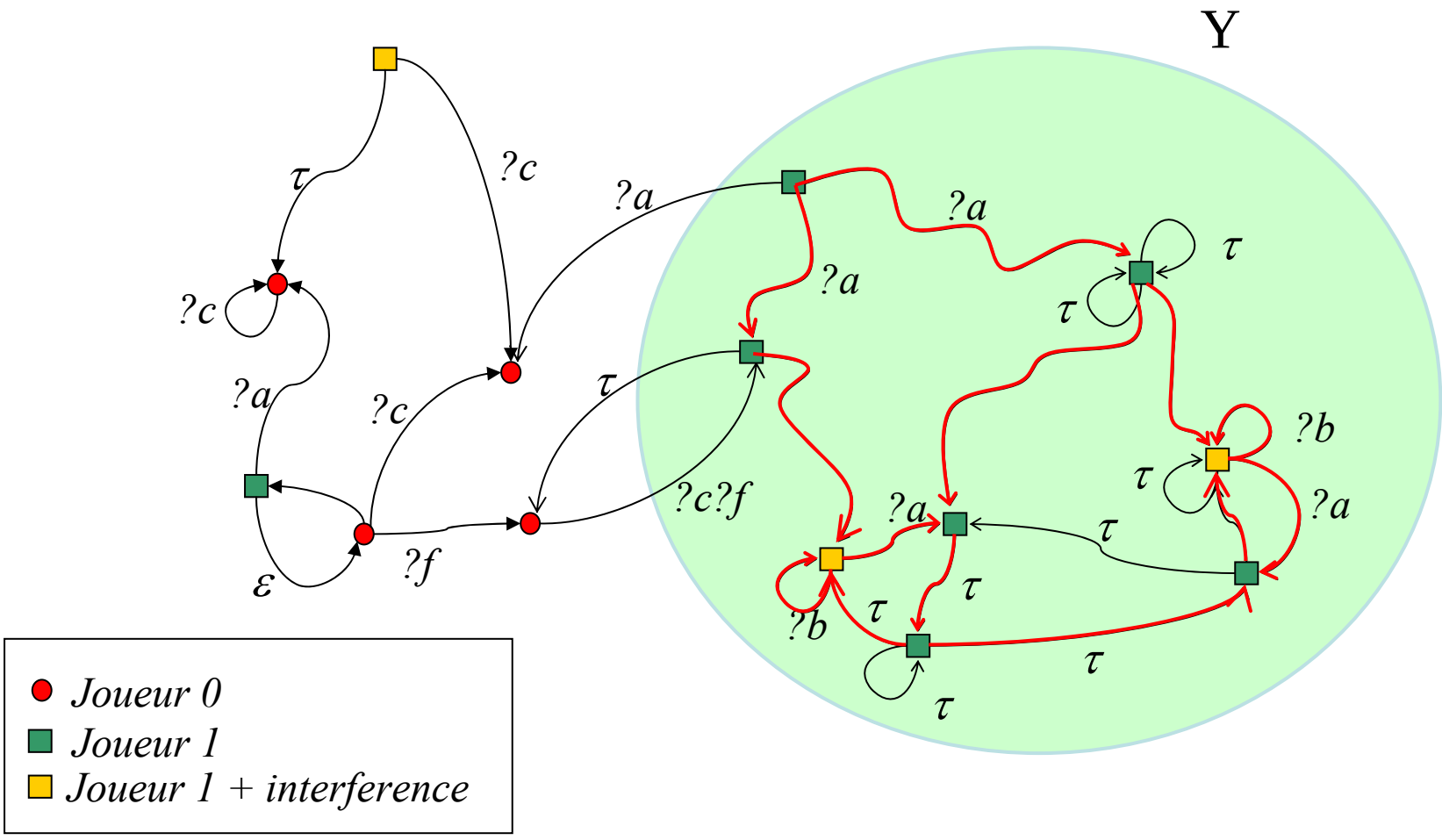
(un seul choix *c*)

Codage d'information mais unique

(Est-ce un canal caché ?)



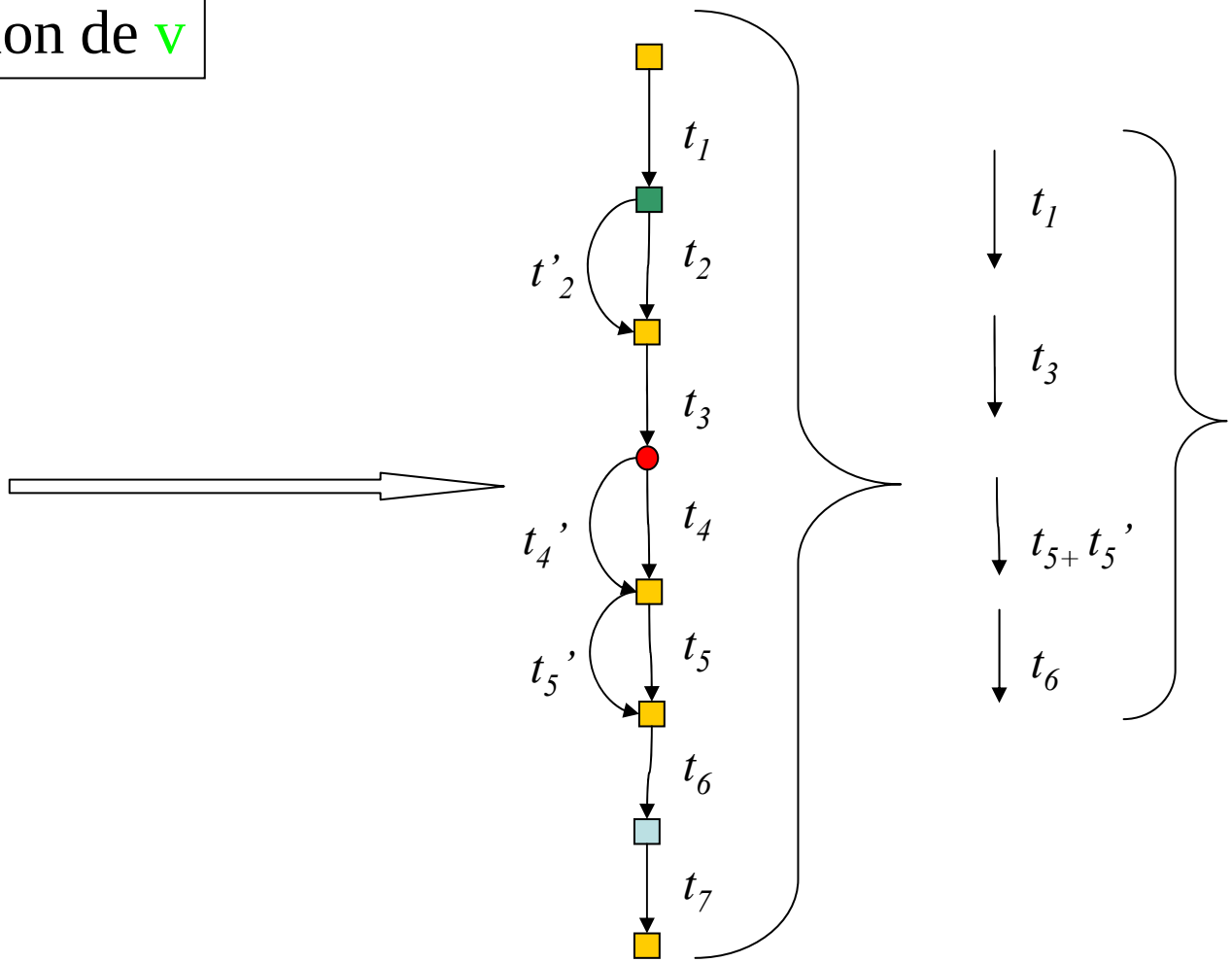
Step 2 : Search a winning subset Y in which
player 1 has a winning strategy f_Y to pass infinitely often through red Vertices while producing observable events



Quand Y et f_Y existent

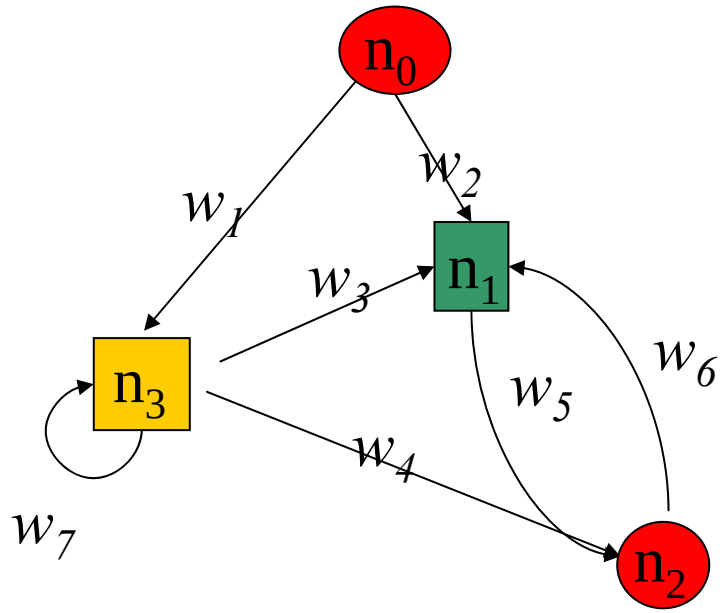
Observation de v

?a
?b
!e
?f
?b
?a
?f
?f
!x
?c
?d
?e

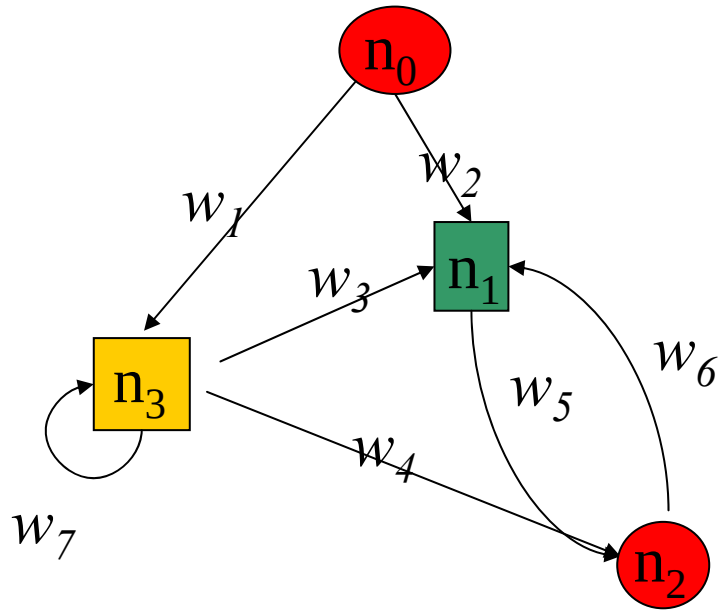


Message

0
1
0
1



Identifier les positions ou
u peut transmettre de l'information à v



Passer infiniment souvent par ■

Canal caché = strategie gagnante dans un jeu
(Muller or Buchi winning condition)

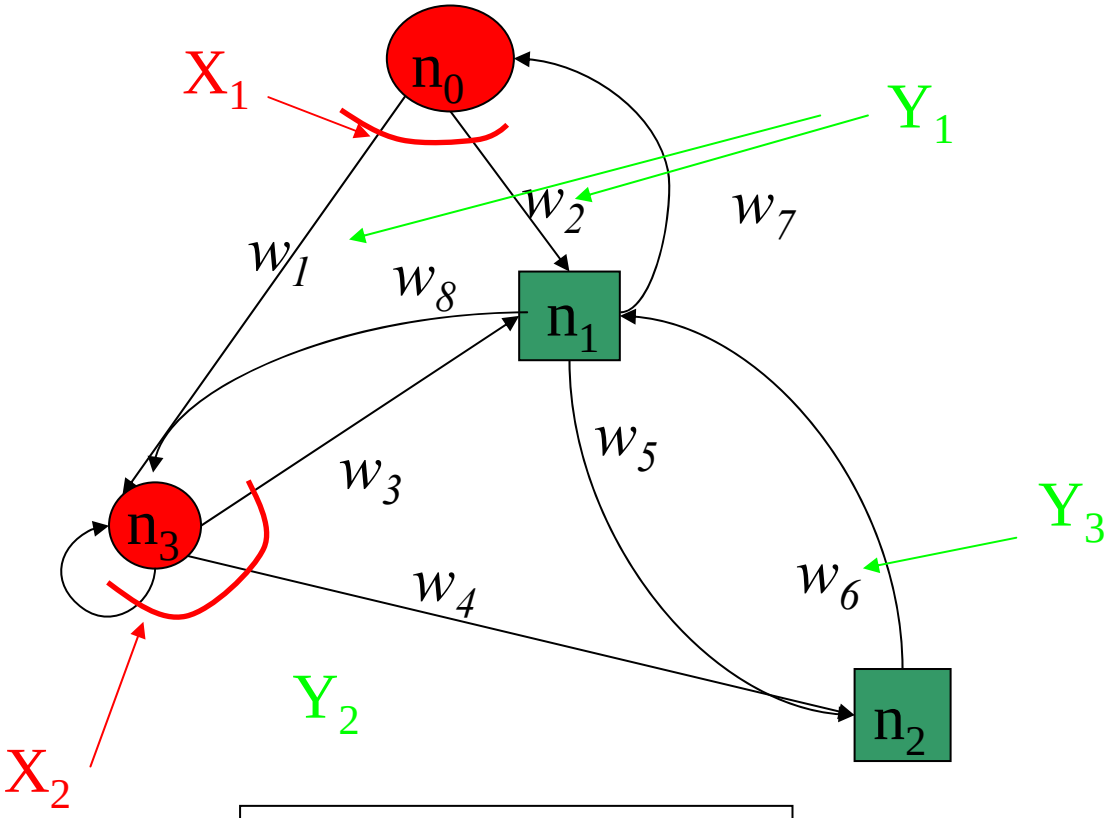
CCs & NI = jeux à somme nulle ?

Maximiser / minimiser l'information mutuelle entre les choix de u et les observations de v

Canal caché / interference ssi il existe une un gain positif pour la paire (u,v) sur les parties infinies

Problème: ce n'est pas un jeu positionnel (le gain le plus élevé ne dépend pas seulement de l'état).

Jeux + Théorie de l'info



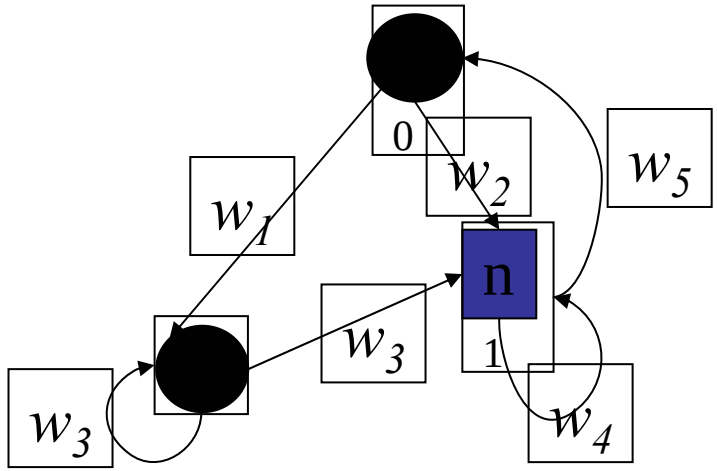
$I(X_1 \ldots X_n ; Y_1 \ldots Y_q)$

Problems

- Some information lost during concatenation:
  - *abba.a = ab.baa=ab.ba.a*
    - $Y_1=w_1$ & $Y_2=w_2$
    - *or* $Y_1=w_3$ & $Y_2=w_3$
    - *or* $Y_1=w_3$ & $Y_2=w_4$ & $Y_3=w_5$
  - Solution: $w_1, \ldots, w_k$ form a code (unique decomposition).
- No « nice form » for

$$C = \lim_{n\to\infty} \frac{I(X_1 \ldots X_n ; Y_1 \ldots Y_q)}{n}$$

- The amount of information sent at n^{th} use of the channel may depend on the *n-1* previous ones.
- Special channel model
  - With memory
  - Stuttering



Conclusion

Variations sur:

- Les modèles
- Les propriétés à satisfaire

…pour assurer la protection des données

La puissance d'expression des modèles utilisés ou la précision du diagnostic sont nécessairement restreints si on veut automatiser…

Modèle = abstraction / approximation

L'analyse formelle d'un modèle :
ne démontre pas l'absence de fuites, mais augmente la confiance
ne met pas à l'abris d'erreurs d'implémentation

La caractérisation d'une fuite est à pondérer par
sa sévérité
sa difficulté de mise en œuvre
sa présence dans l'implémentation réelle du système