

Application de la logique de Hoare aux réseaux de régulation génétique avec multiplexes

Maxime FOLSCHETTE

Cursus ingénieur : Ei3 Informatique
Cursus Master Recherche : ASP-SPIE

Encadrants :

Olivier ROUX
Morgan MAGNIN

Plan de la présentation

- ▶ Le modèle de Thomas

 - Graphes d'interaction

 - Réseaux de régulation génétique

- ▶ Les logiques temporelles

 - LTL, CTL, CTL*

- ▶ La logique de Hoare

 - Triplets de Hoare

 - Plus faible pré-condition

 - Application à l'inférence de paramètres

Introduction

- Sujet à la croisée de deux domaines :
 - Bio-informatique
 - Informatique théorique
- Problématique : modélisation des interactions entre gènes
- Problème de la modélisation
 - Peu de données quantitatives
 - Comportements complexes

Modèles d'interaction génétique

- Réseaux
 - Modèles discrets, basés sur des graphes

Modèles d'interaction génétique

- Réseaux
 - Modèles discrets, basés sur des graphes
- Équations différentielles
 - Continues, parfois étudiées sur des partitionnements discrets de l'espace

Modèles d'interaction génétique

- Réseaux
 - Modèles discrets, basés sur des graphes
- Équations différentielles
 - Continues, parfois étudiées sur des partitionnements discrets de l'espace
- Équations stochastiques
 - Probabilités

Modèles d'interaction génétique

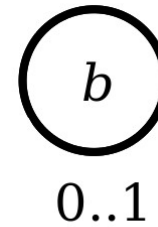
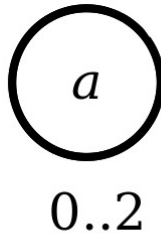
- Réseaux
 - Modèles discrets, basés sur des graphes
- Équations différentielles
 - Continues, parfois étudiées sur des partitionnements discrets de l'espace
- Équations stochastiques
 - Probabilités
- Programmation logique
 - Faits et règles

Modèles d'interaction génétique

- **Réseaux**
 - **Modèles discrets, basés sur des graphes**
- Équations différentielles
 - Continues, parfois étudiées sur des partitionnements discrets de l'espace
- Équations stochastiques
 - Probabilités
- Programmation logique
 - Faits et règles

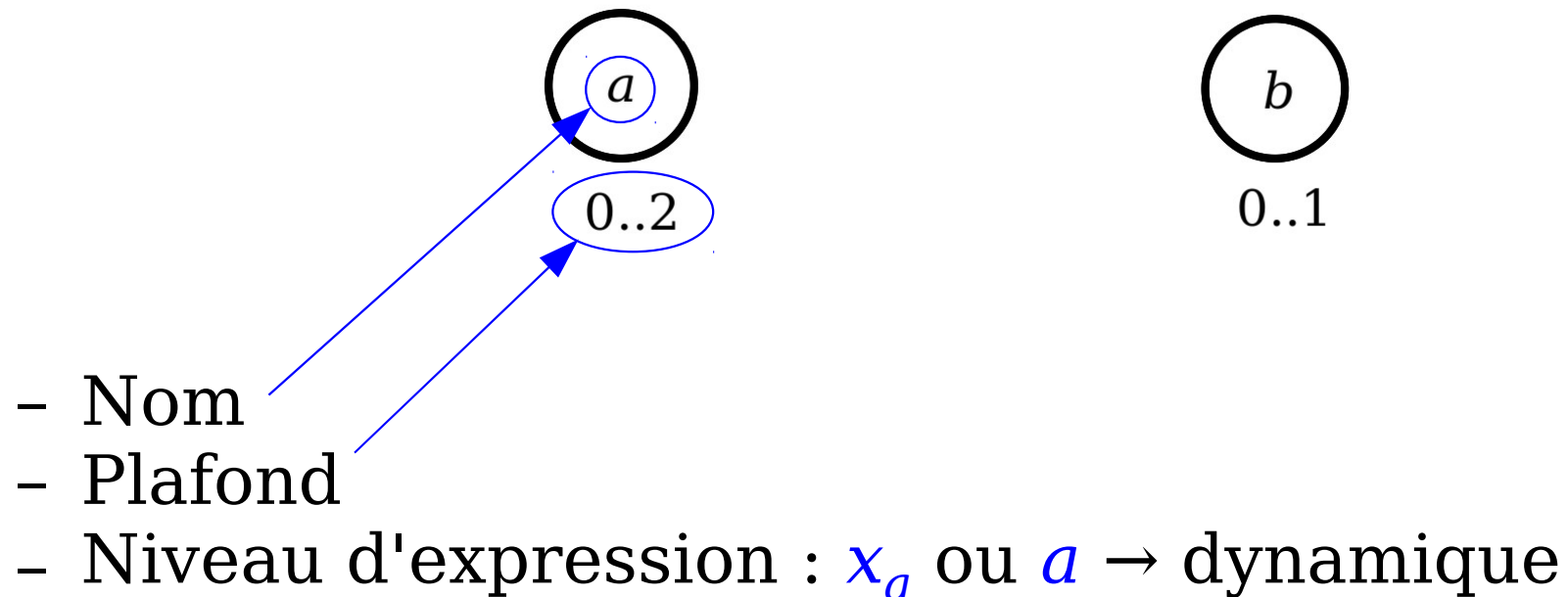
Graphe d'interaction

Nœuds = gènes



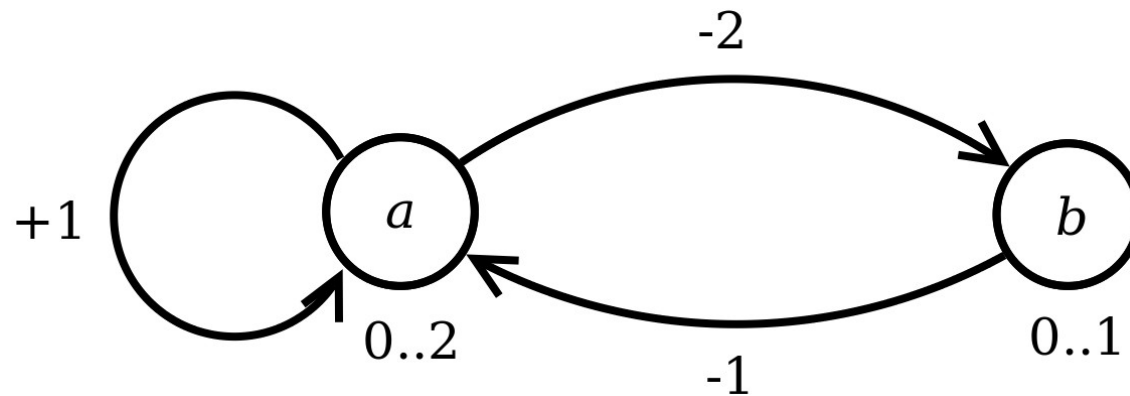
Graphe d'interaction

Nœuds = gènes



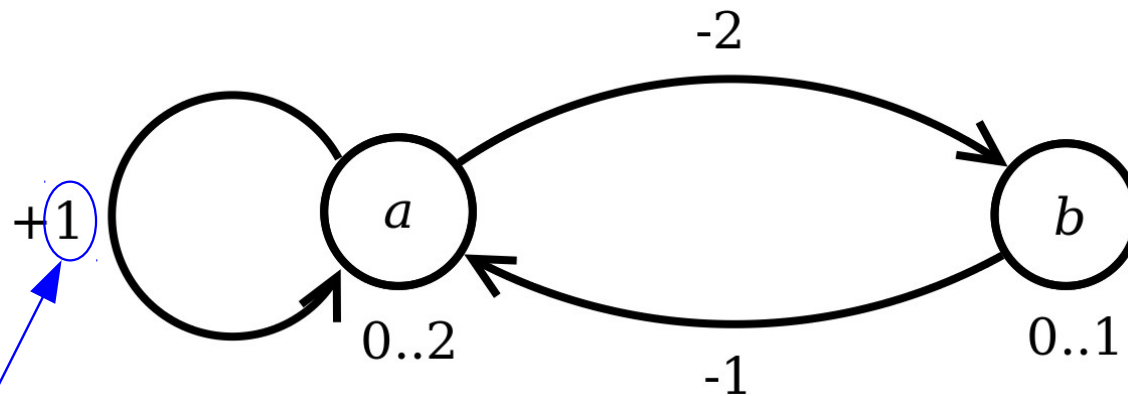
Graphe d'interaction

Arcs orientés = interactions



Graphe d'interaction

Arcs orientés = interactions



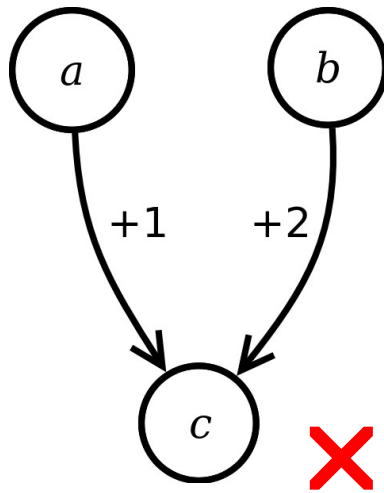
- Seuil
- Type (activation / inhibition)

+

-

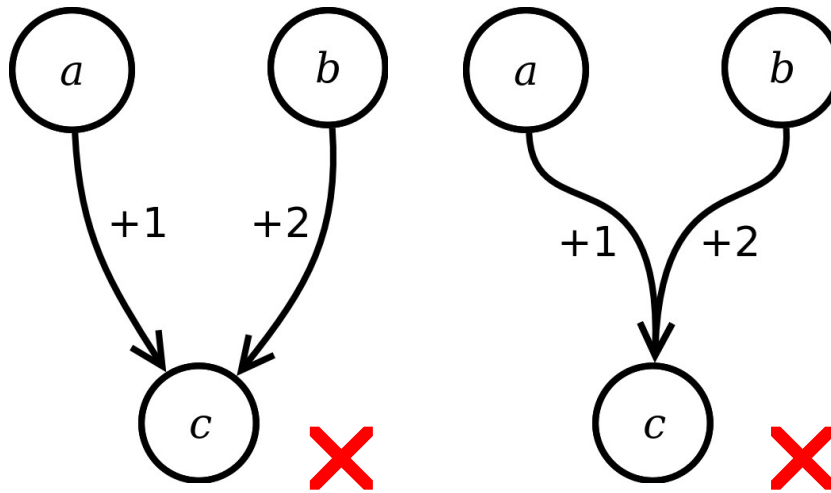
Introduction des multiplexes

Problème : insuffisant pour certains cas



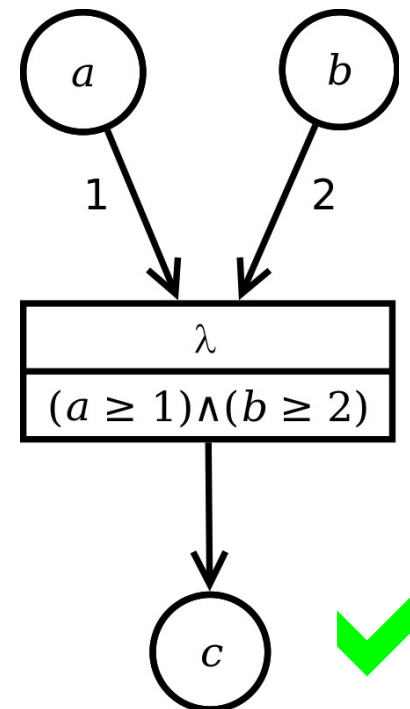
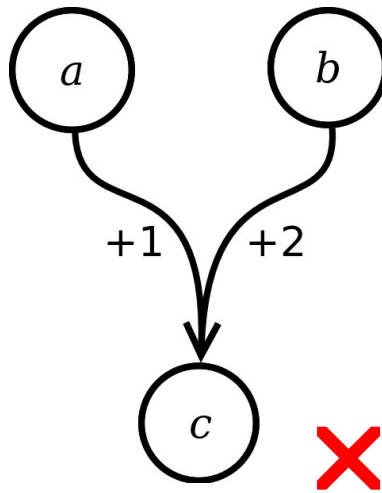
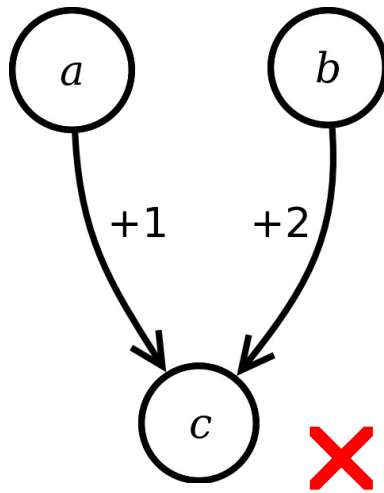
Introduction des multiplexes

Problème : insuffisant pour certains cas



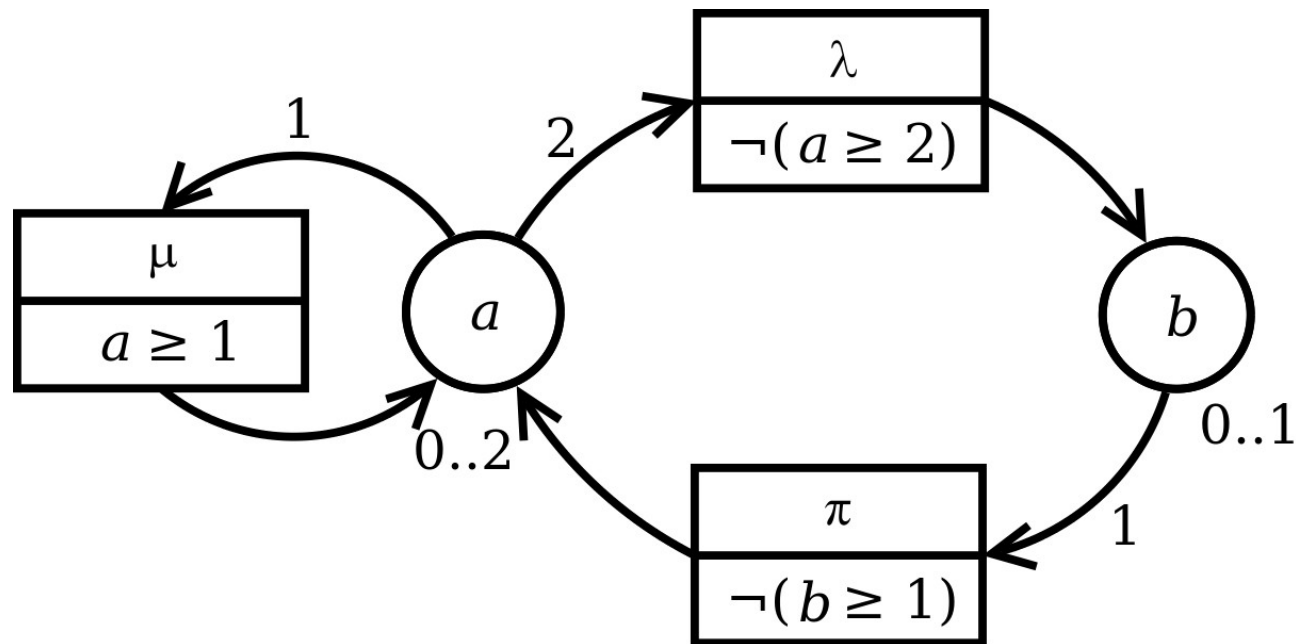
Introduction des multiplexes

Problème : insuffisant pour certains cas



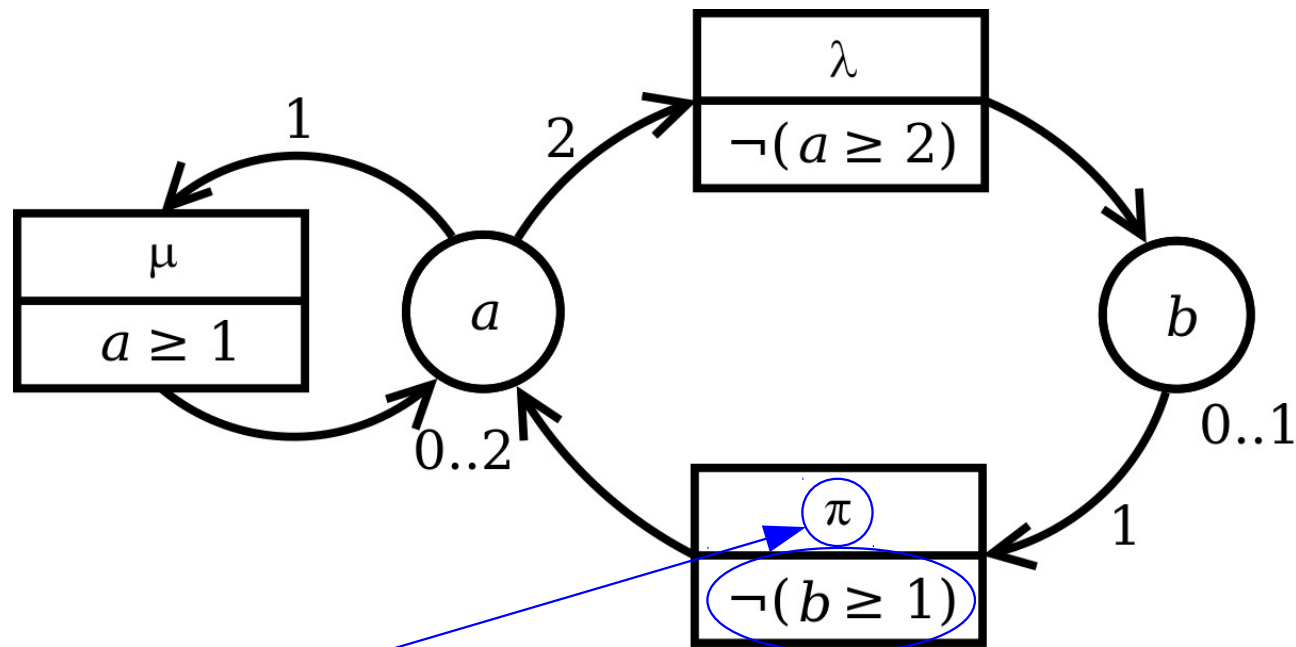
Introduction des multiplexes

Multiplexes = interactions



Introduction des multiplexes

Multiplexes = interactions



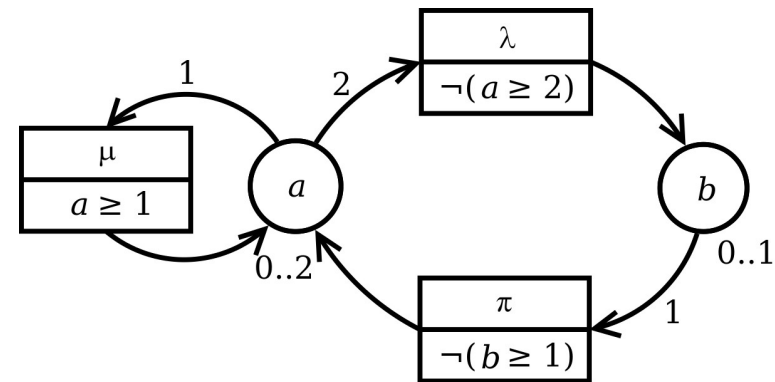
- Nom
- Condition d'influence

Paramétrisation

Quantifier la « force » des interaction
 → Création d'une carte des influences

ω	$k_{a,\omega}$
\emptyset	2
$\{\mu\}$	2
$\{\pi\}$	0
$\{\mu, \pi\}$	0

ω	$k_{b,\omega}$
\emptyset	1
$\{\lambda\}$	0

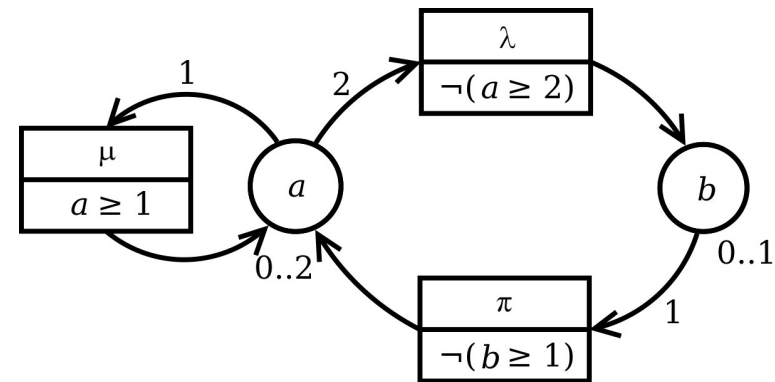


Paramétrisation

Quantifier la « force » des interaction
 → Création d'une carte des influences

ω	$k_{a,\omega}$	$k_{a,\omega}$
\emptyset	2	1
$\{\mu\}$	2	2
$\{\pi\}$	0	0
$\{\mu, \pi\}$	0	1

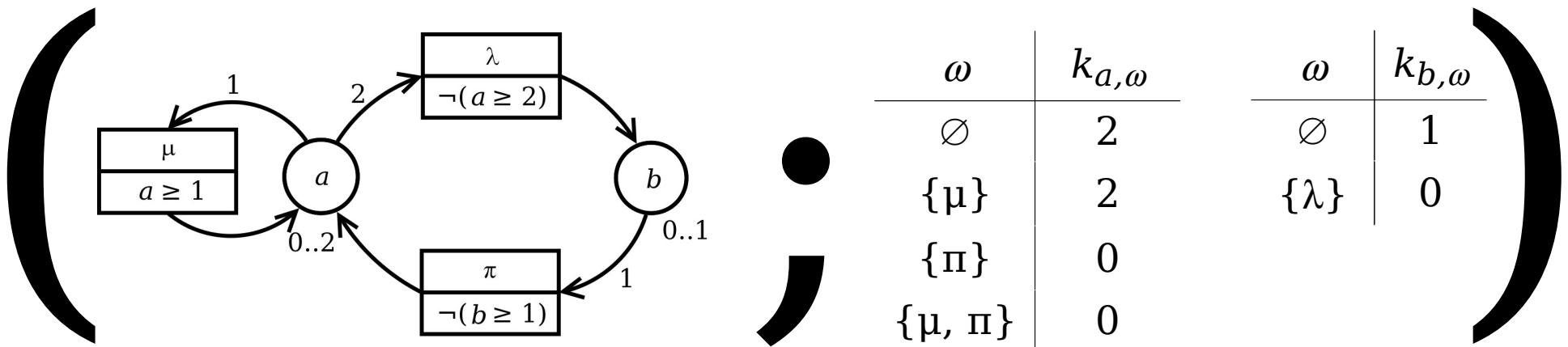
ω	$k_{b,\omega}$	$k_{b,\omega}$
\emptyset	1	1
$\{\lambda\}$	0	0



Réseau de régulation génétique

=

Graphe d'interaction + paramétrisation

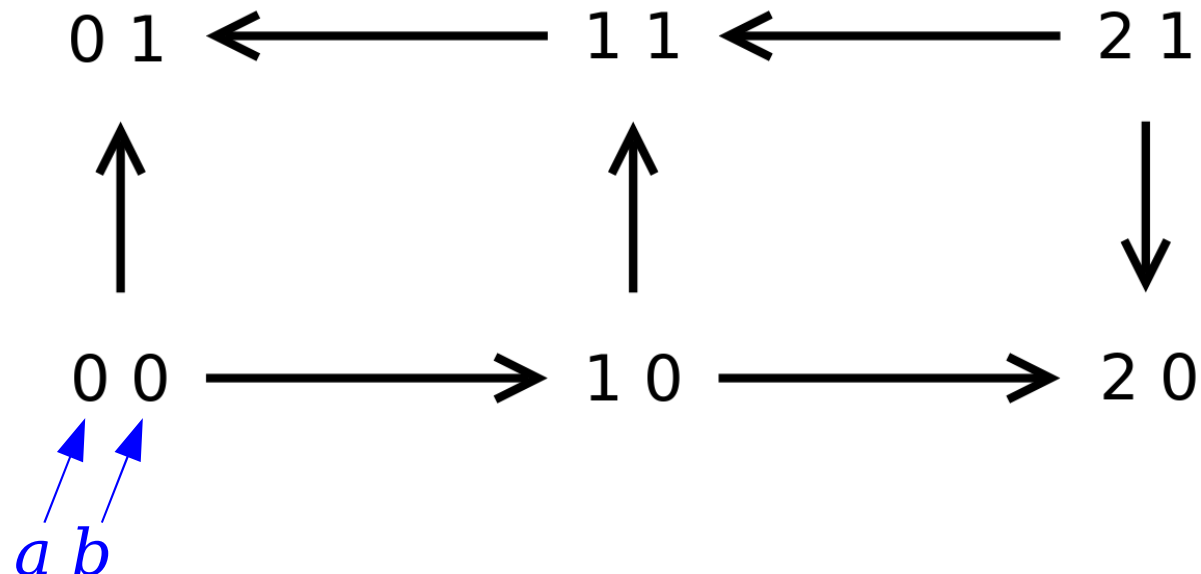


→ Décrit complètement le système

Graphe d'états asynchrone

Se déduit du réseau de régulation

- Système observé en une succession d'instants
- Un gène peut évoluer d'**une seule unité**
- **Asynchrone** : Un seul gène évolue à la fois



Discussion

Avantages du modèle

- Simplification cohérente de la réalité
- Permet de s'affranchir de la connaissance de données quantitatives

Limitations

- Nombre de paramétrisations possibles
- Taille du graphe d'états (explosion combinatoire)

→ Outils pour étudier ce modèle

Les logiques temporelles

But à l'origine :

→ Étude des chemins d'un graphe

Domaine d'utilisation :

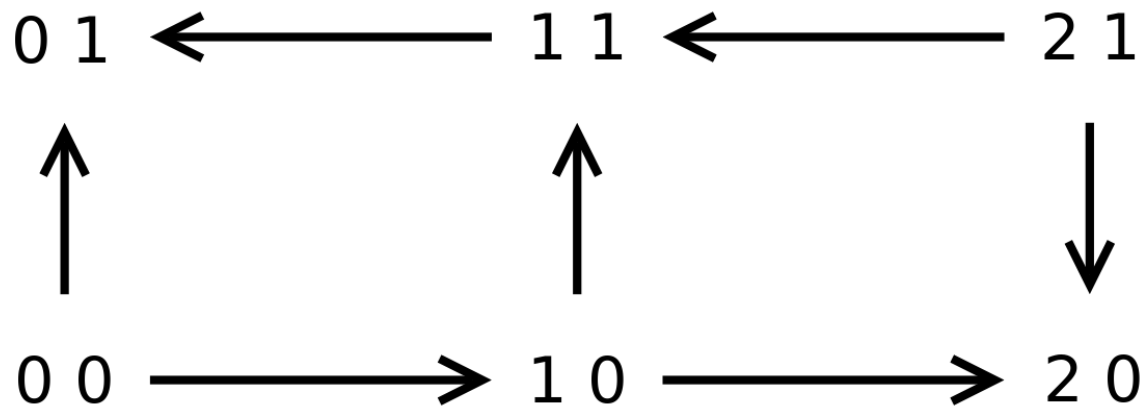
→ Vérification formelle

Application :

→ Recherche des propriétés du graphe d'états

Les logiques temporelles

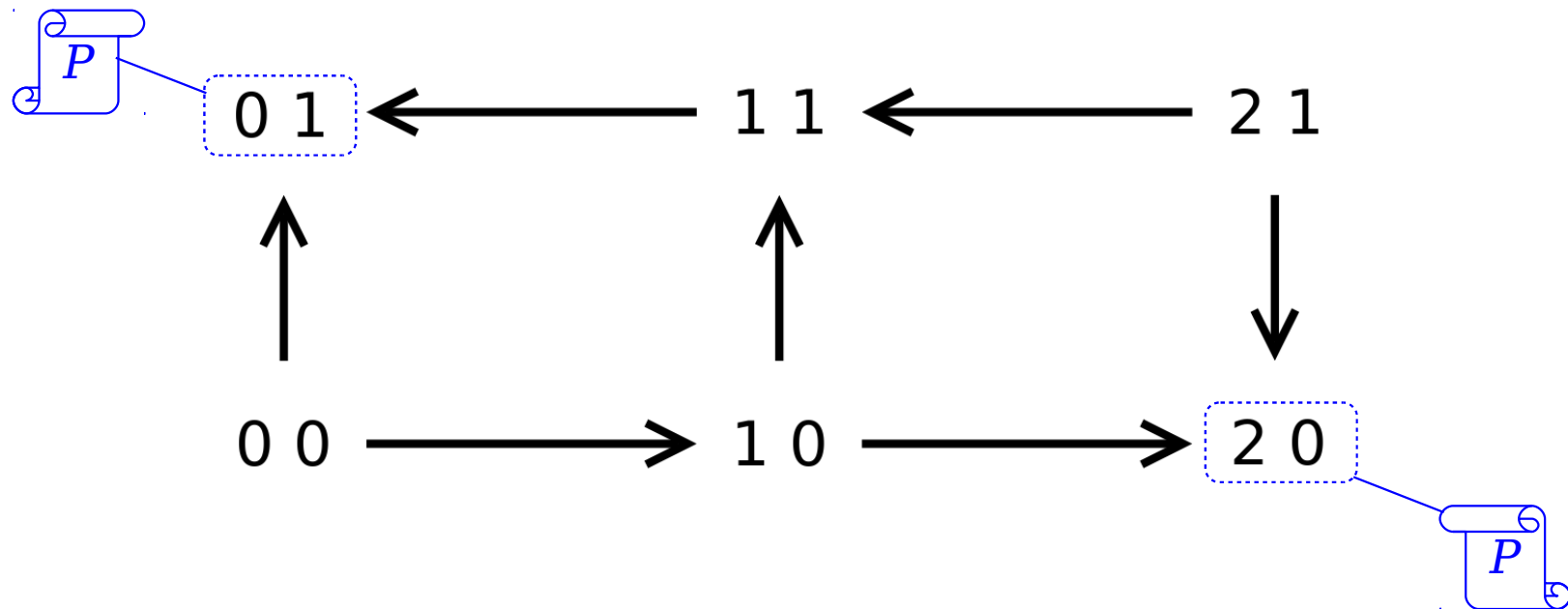
S'appliquent aux **structures de Kripke**
= graphe orienté dont les états sont étiquetés



Les logiques temporelles

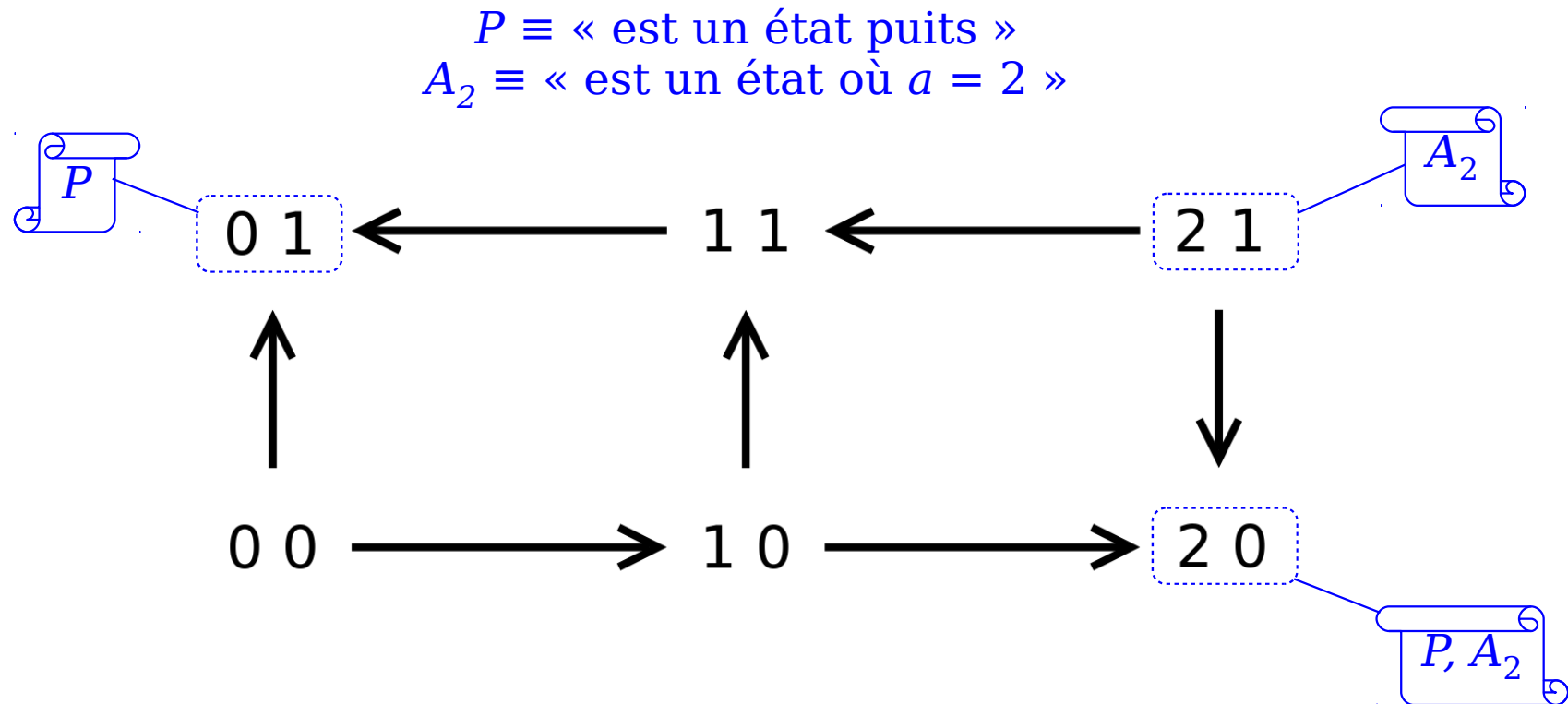
S'appliquent aux **structures de Kripke**
= graphe orienté dont les états sont étiquetés

$P \equiv$ « est un état puits »



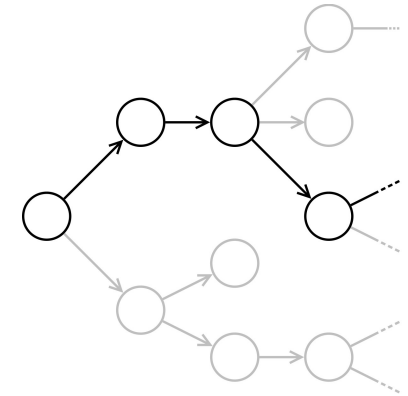
Les logiques temporelles

S'appliquent aux **structures de Kripke**
 = graphe orienté dont les états sont étiquetés



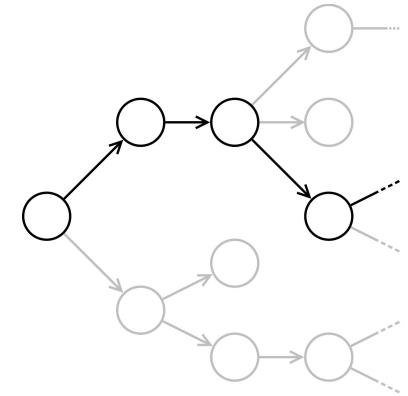
Les différentes logiques

- LTL \rightarrow sur un chemin donné
 $X\phi \quad \phi U \psi$



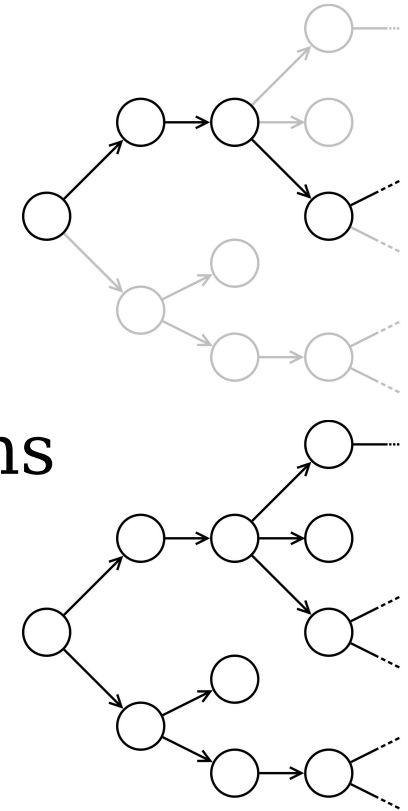
Les différentes logiques

- LTL \rightarrow sur un chemin donné
 $X\phi \quad \phi U \psi \quad + \quad G\phi \quad F\phi$



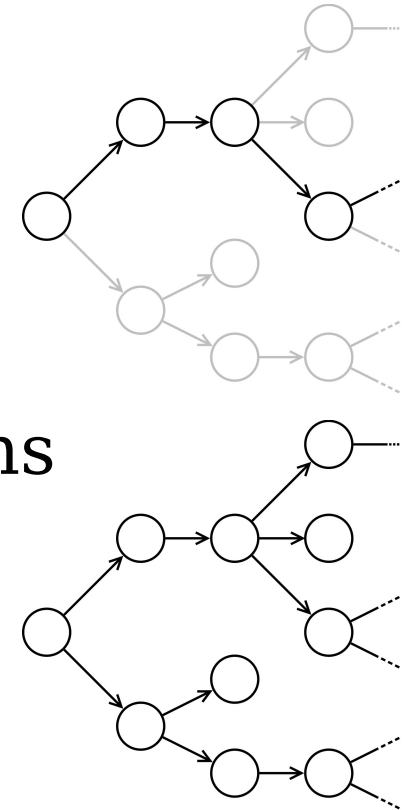
Les différentes logiques

- LTL \rightarrow sur un chemin donné
 $X\phi \quad \phi U \psi \quad + \quad G\phi \quad F\phi$
- CTL \rightarrow sur l'arborescence des chemins
à partir d'un état donné
 $EX\phi \quad E \phi U \psi$
 $AX\phi \quad A \phi U \psi$



Les différentes logiques

- LTL \rightarrow sur un chemin donné
 $X\phi \quad \phi U \psi \quad + \quad G\phi \quad F\phi$
- CTL \rightarrow sur l'arborescence des chemins
à partir d'un état donné
 $EX\phi \quad E \phi U \psi \quad + \quad EG\phi \quad EF\phi$
 $AX\phi \quad A \phi U \psi \quad + \quad AG\phi \quad AF\phi$



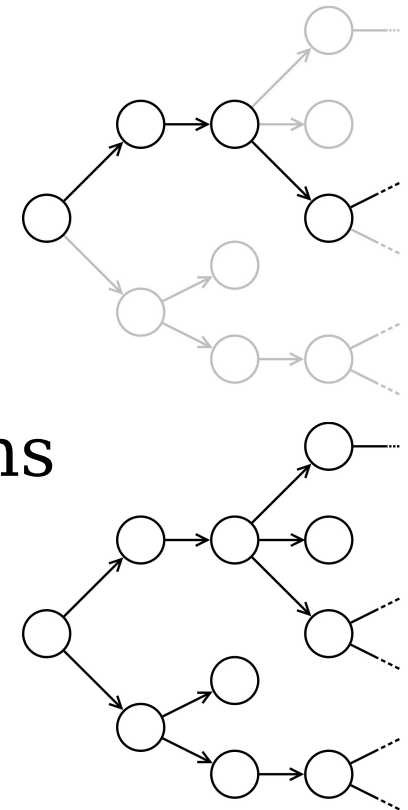
Les différentes logiques

- LTL → sur un chemin donné

$$X\phi \quad \phi U \psi \quad + \quad G\phi \quad F\phi$$
- CTL → sur l'arborescence des chemins à partir d'un état donné

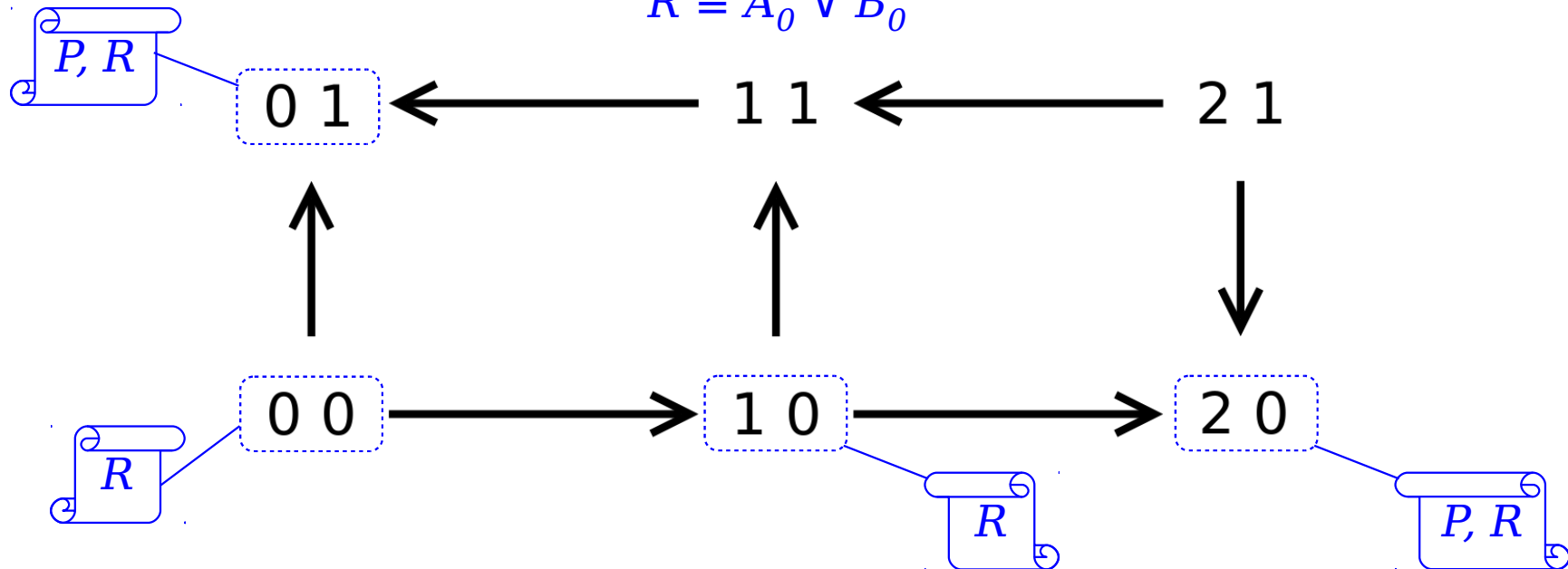
$$EX\phi \quad E \phi U \psi \quad + \quad EG\phi \quad EF\phi$$

$$AX\phi \quad A \phi U \psi \quad + \quad AG\phi \quad AF\phi$$
- CTL* → “Réunion” de LTL et CTL
 (Formules avec et sans quantificateur)



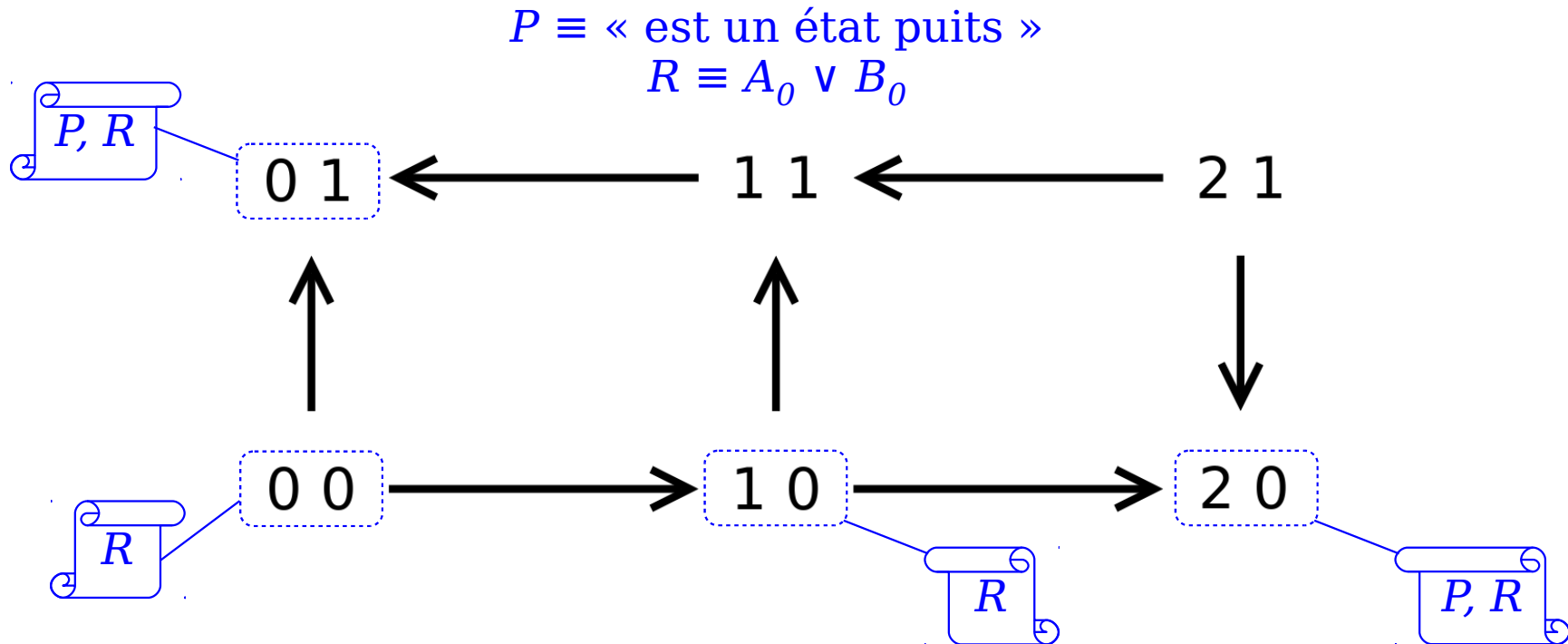
Application : CTL

$P \equiv \ll \text{est un état puits} \gg$
 $R \equiv A_0 \vee B_0$



- $\forall q :$ $q \models AF P$
- $\forall q \models A_1 :$ $q \models EX R$
- $\forall q :$ $q \models (A_1 \Rightarrow EX R)$

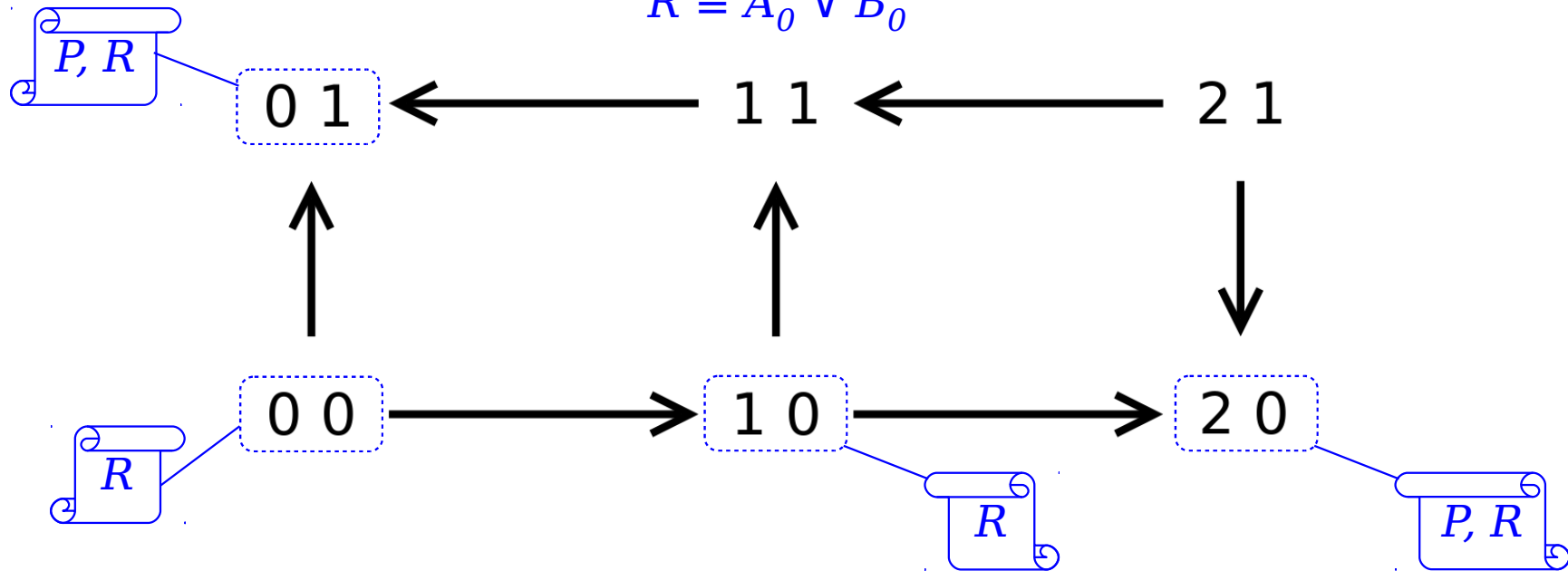
Application : CTL



- $\forall q :$ $q \models AF P$
- $\forall q \models A_1 :$ $q \models EX R$
- $\forall q :$ $q \models (A_1 \Rightarrow EX R)$

Application : CTL

$P \equiv$ « est un état puits »
 $R \equiv A_0 \vee B_0$



- $\forall q :$ $q \models AF P$
- $\forall q \models A_1 :$ $q \models EX R$
- $\forall q :$ $q \models (A_1 \Rightarrow EX R)$

Discussion

Avantage :

- Expression adaptée à l'analyse de graphes

Limite dans l'application au modèle de Thomas :

- Nécessite la construction du graphe d'états
→ Explosion combinatoire

La logique de Hoare

- Outil de l'informatique théorique
- Conçue pour la preuve de programmes

Rattachement au modèle de Thomas :

- Utilisable directement sur le réseau de régulation
- Donne des propriétés sur la paramétrisation

La logique de Hoare

Triplets de Hoare :

$$\{ P \} Q \{ R \}$$

P : Pré-condition

Q : Programme informatique

R : Post-condition

« Si P est vraie avant exécution de Q ,
alors R sera vraie après exécution de Q . »

Exemple : $\{ y = 4 \} y := y + 1 \{ y = 5 \}$

Axiomes et règles

- **Axiomes** pour fonder la logique
 - Programme vide : $\{ P \} \text{ skip } \{ P \}$
 - Affectation : $\{ P[expr/var] \} var := expr \{ P \}$
- **Règles** pour construire la logique
 - Conséquence
 - Composition de deux programmes
 - Structures Si-Alors-Sinon et Tant_que

Axiomes et règles

- **Axiomes** pour fonder la logique
 - Programme vide : $\{ P \} \text{ skip } \{ P \}$
 - Affectation : $\{ P[expr/var] \} var := expr \{ P \}$
- **Règles** pour construire la logique
 - Conséquence
 - Composition de deux programmes
 - Structures Si-Alors-Sinon et Tant_que

Exemple d'affectation :

$$\{ y + 1 \geq 5 \} y := y + 1 \{ y \geq 5 \}$$

Axiomes et règles

- **Axiomes** pour fonder la logique
 - Programme vide : $\{ P \} \text{ skip } \{ P \}$
 - Affectation : $\{ P[expr/var] \} var := expr \{ P \}$
- **Règles** pour construire la logique
 - Conséquence
 - Composition de deux programmes
 - Structures Si-Alors-Sinon et Tant_que

Exemple d'affectation :

$$\{ y \geq 4 \} y := y + 1 \{ y \geq 5 \}$$

Démarche des preuves

```
n := t;  
r := 1;  
Tant_que (n ≠ 0)  
    r := r × n;  
    n := n − 1;  
Fin_tant_que;
```

Démarche des preuves

```

{t ≥ 0}      n := t;      {n ≥ 0 ∧ t = n}
{n ≥ 0 ∧ t = n}  r := 1;      {n ≥ 0 ∧ t = n ∧ r = 1}
              Tant_que (n ≠ 0)
{r = t! / n! ∧ t ≥ n ∧ n > 0}  r := r × n;  {r = t! / (n-1)! ∧ t ≥ n ∧ n > 0}
{r = t! / (n-1)! ∧ t ≥ n ∧ n > 0} n := n - 1;  {r = t! / n! ∧ t ≥ n ∧ n ≥ 0}
              Fin_tant_que;

```

Démarche des preuves

```
{t ≥ 0}      n := t;  
             r := 1;      {n ≥ 0 ∧ t = n ∧ r = 1}  
             Tant_que (n ≠ 0)  
{r = t! / n! ∧ t ≥ n ∧ n > 0}  r := r × n;  
                                 n := n - 1;  {r = t! / n! ∧ t ≥ n ∧ n ≥ 0}  
                                 Fin_tant_que;
```

Démarche des preuves

```
{t ≥ 0}      n := t;  
             r := 1;      {n ≥ 0 ∧ t = n ∧ r = 1}
```

```
{r = t! / n! ∧ t ≥ n ∧ n ≥ 0} Tant_que (n ≠ 0)  
    r := r × n;  
    n := n - 1;  
Fin_tant_que; {r = t! / n! ∧ t ≥ n ∧ n ≥ 0 ∧ n = 0}
```

Démarche des preuves

```
{t ≥ 0}
    n := t;
    r := 1;
    Tant_que (n ≠ 0)
        r := r × n;
        n := n - 1;
    Fin_tant_que; {r = t! / n! ∧ t ≥ n ∧ n ≥ 0 ∧ n = 0}
```

Démarche des preuves

```
{t ≥ 0}      n := t;  
             r := 1;  
             Tant_que (n ≠ 0)  
               r := r × n;  
               n := n - 1;  
             Fin_tant_que;           {r=t!}
```

Plus faible pré-condition

Les triplets de Hoare doivent être bien choisis pour que les preuves soient menées à bien

$$\begin{array}{l} \{ y \geq 4 \} y:=y+1 \{ y \geq 5 \} \\ \{ y = 4 \} y:=y+1 \{ y \geq 5 \} \\ \{ y \geq 4 \} y:=y+1 \{ y \geq 0 \} \end{array}$$

→ Trouver **la plus faible pré-condition**
ou la plus forte post-condition

Plus faible pré-condition

$$\{ P \} Q \{ R \}$$

Opérateur : **WP(Q, R)**

Nouvelles règles pour cet opérateur :

- Composition
- Affectation

$$\mathbf{WP(var:=expr, R) \equiv R[expr/var]}$$

- Structures Si-Alors-Sinon et Tant_que

→ On connaît exactement la pré-condition d'une affectation

Application au modèle de Thomas

$\{ P \} Q \{ R \}$



Application au modèle de Thomas

Assertion sur le niveau
d'expression des gènes

ex : $a = 2$

$\{ P \} Q \{ R \}$



Application au modèle de Thomas

Assertion sur le niveau
d'expression des gènes

ex : $a = 2$



The diagram shows a sequence of gene expression levels: $\{ P \} Q \{ R \}$. The letters P , Q , and R are enclosed in curly braces. The Q and R are also enclosed in circles. A light blue rectangular background highlights the entire sequence. Two blue arrows point from the text above to the circled Q and R .

Application au modèle de Thomas

Évolution du système :

incrémentations ou
décrémentations

ex : $a+$ / $a-$

→ **Affectations**

Assertion sur le niveau
d'expression des gènes

ex : $a = 2$

{ P } Q { R }

Application au modèle de Thomas

Évolution du système :

incrémentations ou
décrémentations

ex : $a+$ / $a-$

→ **Affectations**

Assertion sur le niveau
d'expression des gènes

ex : $a = 2$



Application au modèle de Thomas

Évolution du système :

incrémentations ou
décrémentations

ex : $a+$ / $a-$

→ **Affectations**

Assertion sur le niveau
d'expression des gènes

ex : $a = 2$



Calcul de la plus faible pré-condition

Application au modèle de Thomas

Évolution du système :

incrémentations ou
décrémentations

ex : $a+$ / $a-$

→ **Affectations**

Assertion sur le niveau
d'expression des gènes

ex : $a = 2$



Calcul de la plus faible pré-condition

⇒ Assertion sur :

- le niveau d'expression des gènes ex : $a = 1$
- la paramétrisation ex : $k_{a,\omega} \geq 2$

Discussion

Permet la vérification de programmes

Appliquée au modèle de Thomas :

permet l'inférence de paramètres biologiques
via l'opérateur de plus faible pré-condition

Avantages :

- Pas de recherche exhaustive
- Dispense de construire le graphe d'états

Conclusion

Modèle de Thomas : Modèle puissant mais difficile à étudier pour des graphes de grande taille

Logique de Hoare : Offre une nouvelle approche pour la déduction de paramètres biologiques, complémentaire aux **logiques temporelles**

Stage d'application :

- Implémentation informatique
- Ajout de la composante temporelle

Merci pour votre attention

Bibliographie • Modèle de Thomas

- A. Richard, J.-P. Comet, G. Bernot :
[R. Thomas' logical method.](#)
Avril 2008. Tutorials on modelling methods and tools : Modelling a genetic switch and Metabolic Networks, Spring School on Modelling Complex Biological Systems in the Context of Genomics.
- G. Bernot, J.-P. Comet, Z. Khalis :
[Gene regulatory networks with multiplexes.](#)
European Simulation and Modelling Conference Proceedings, pages 423-432, France, Octobre 2008.

Bibliographie • Logiques temporelles

- Ph. Schnoebelen, B. Bérard, M. Bidoit, F. Laroussinie, A. Petit : *Vérification de logiciels : techniques et outils du model-checking*. Vuibert, Avril 1999.

- A. Arnold : *Systèmes de transitions finis et sémantique des processus communicants*. Études et recherches en informatique. Masson, 1992.

- B. Bérard : *Model checking temporisé*. In O. H. Roux, C. Jard, editors, *Approches formelles des systèmes embarqués communicants*. Hermes/Lavoisier, Octobre 2008.

Bibliographie • Logique de Hoare

- C. A. R. Hoare :

[An axiomatic basis for computer programming.](#)

Communications of the ACM, 12, pages 576–580, Octobre 1969.

- A. Møller :

[Program verification with Hoare logic.](#)

Technical report, University of Aarhus. 2004.

<http://www.brics.dk/~amoeller/talks/hoare.pdf>