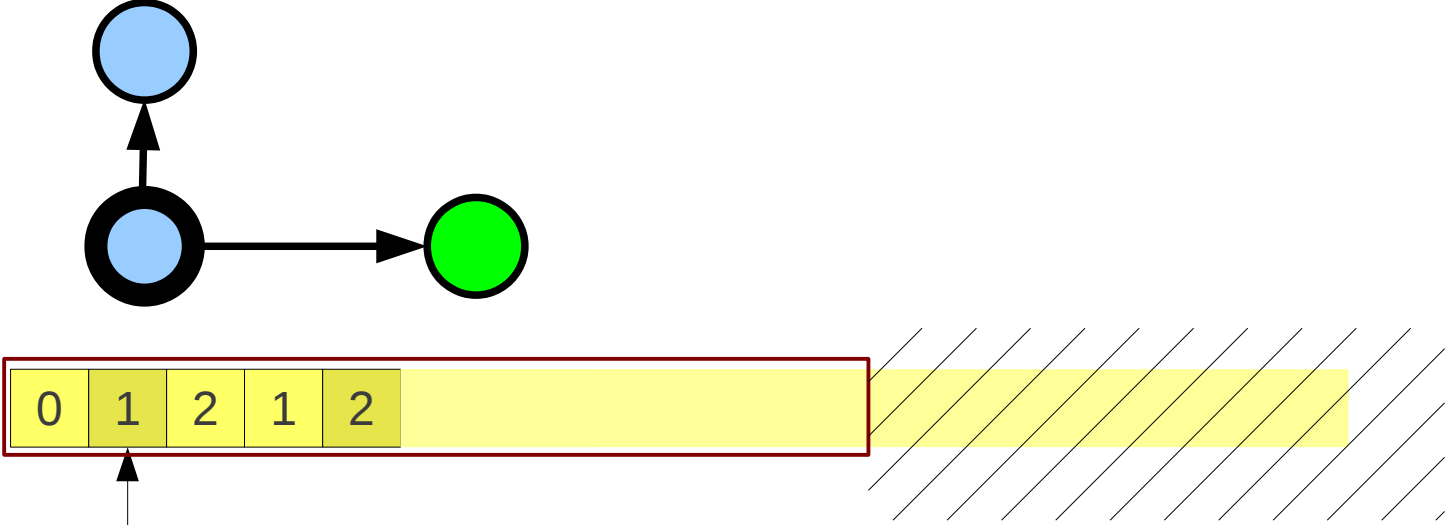


# PSPACE

François Schwarzentruber  
ENS Cachan



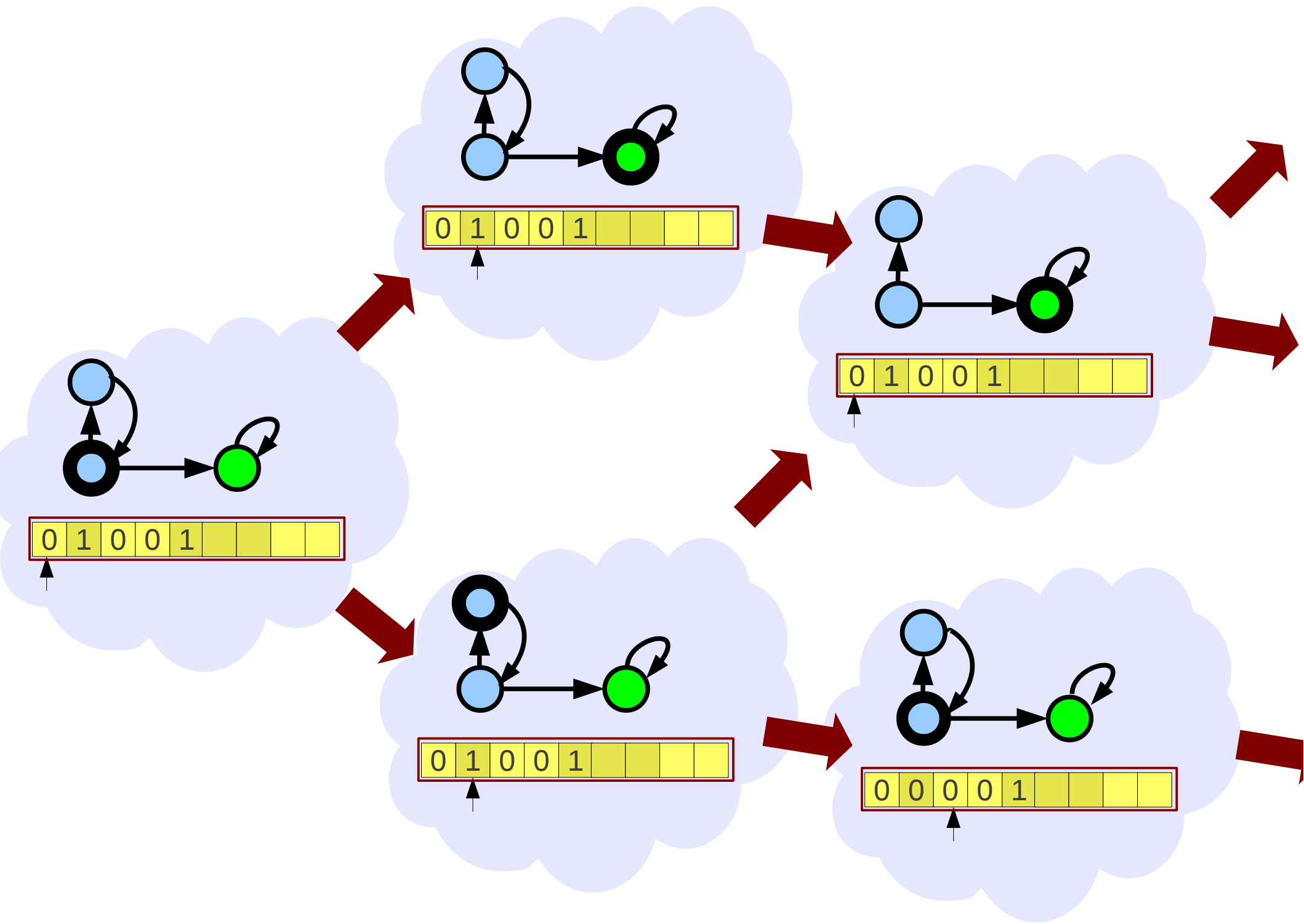
# Motivation

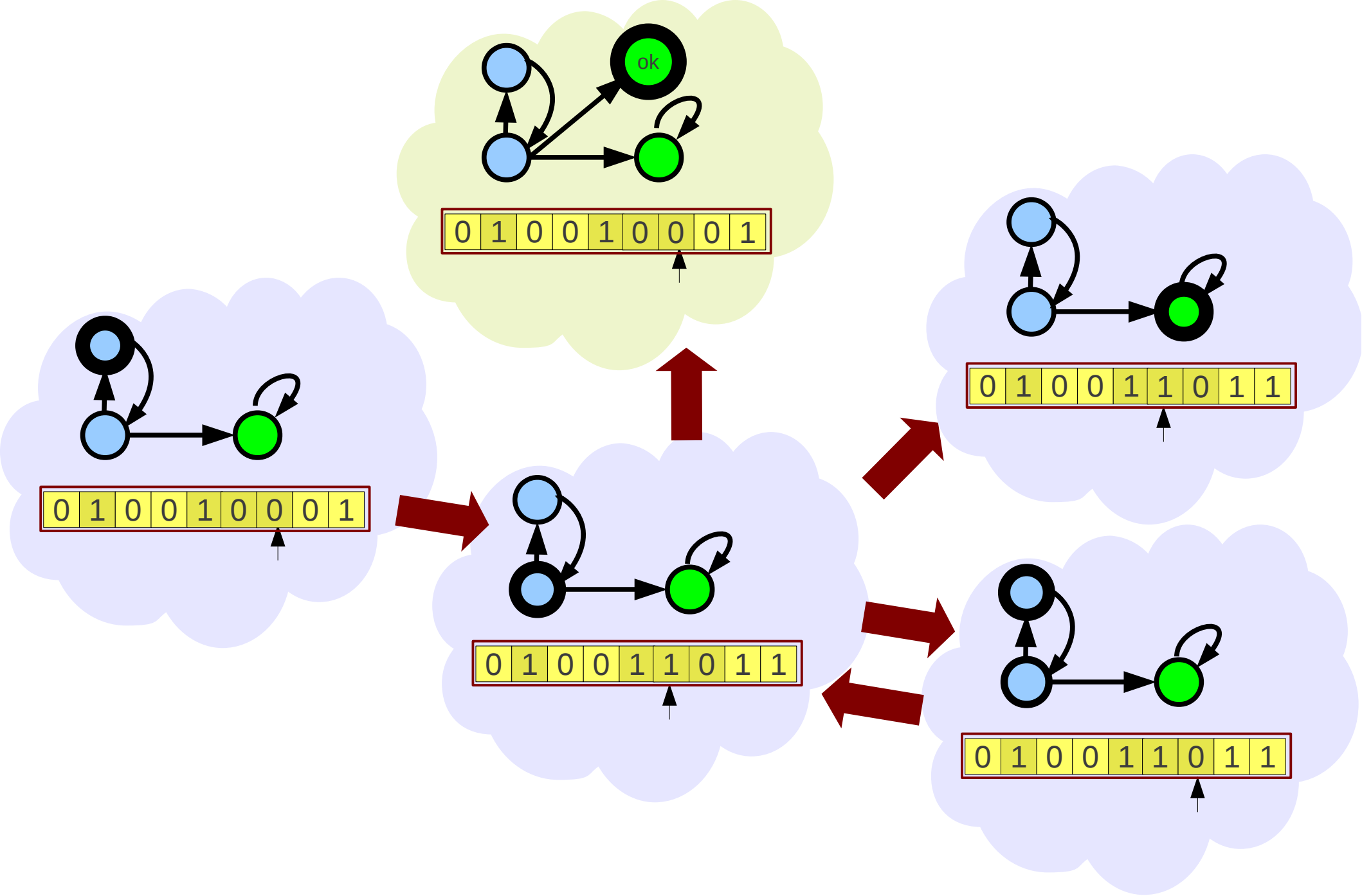
Problème de  
“recherche”

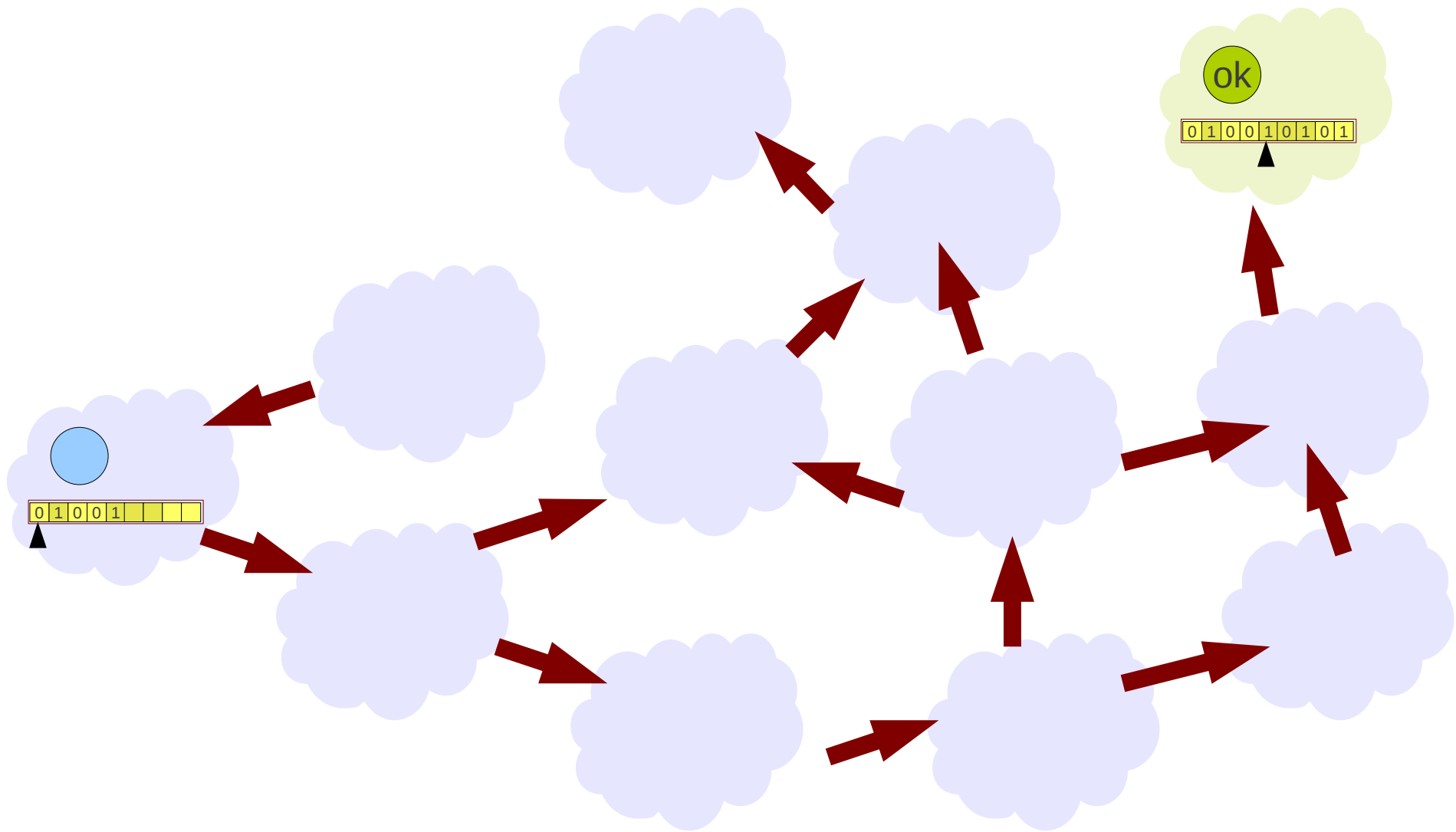
- SAT
- Couverture d'ensemble
- Chemin hamiltonien

Problème de  
“stratégie gagnante”  
dans un jeu à 2  
joueurs

- QBF
- Jeu de géographie



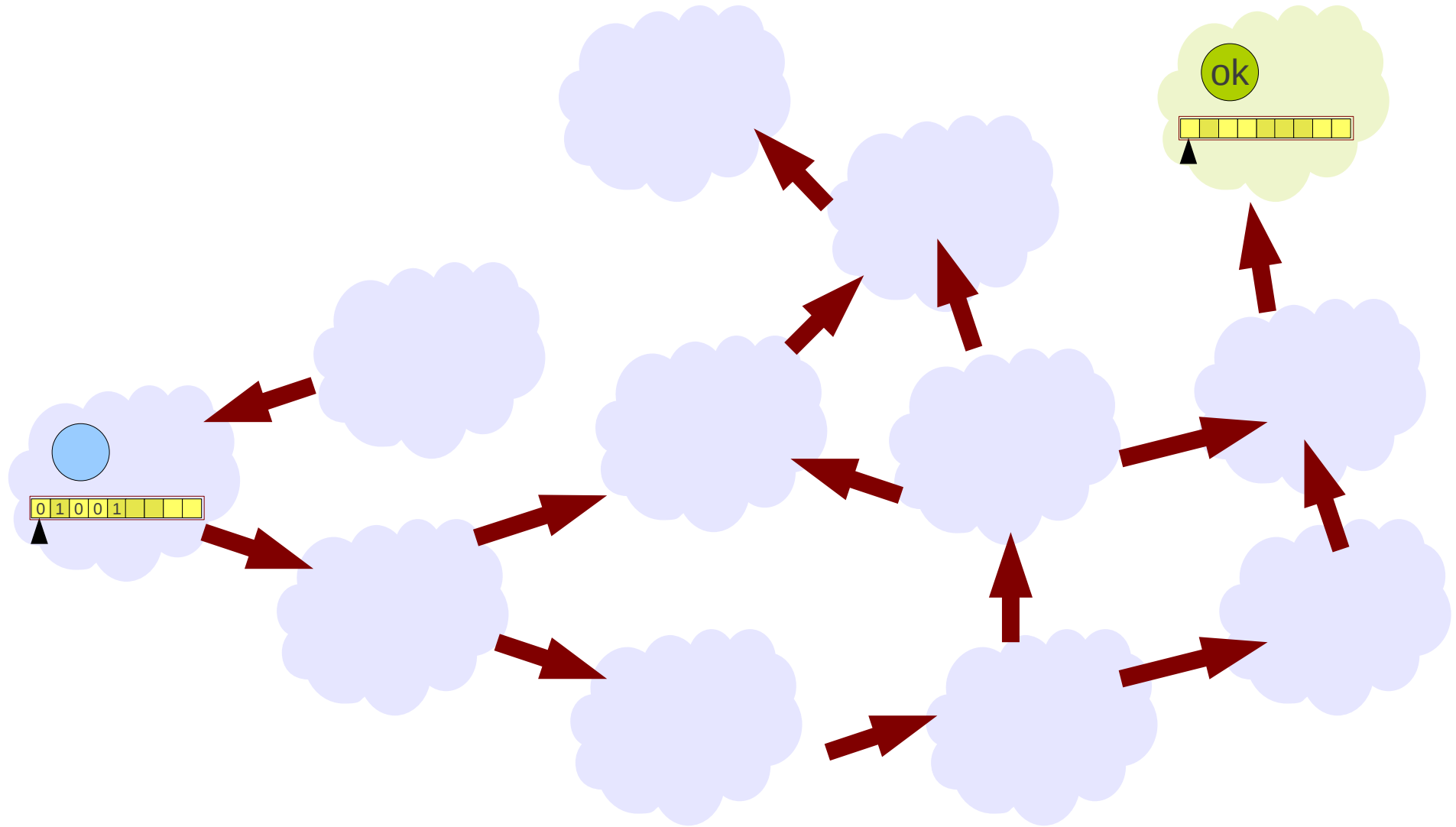




# Plan du cours

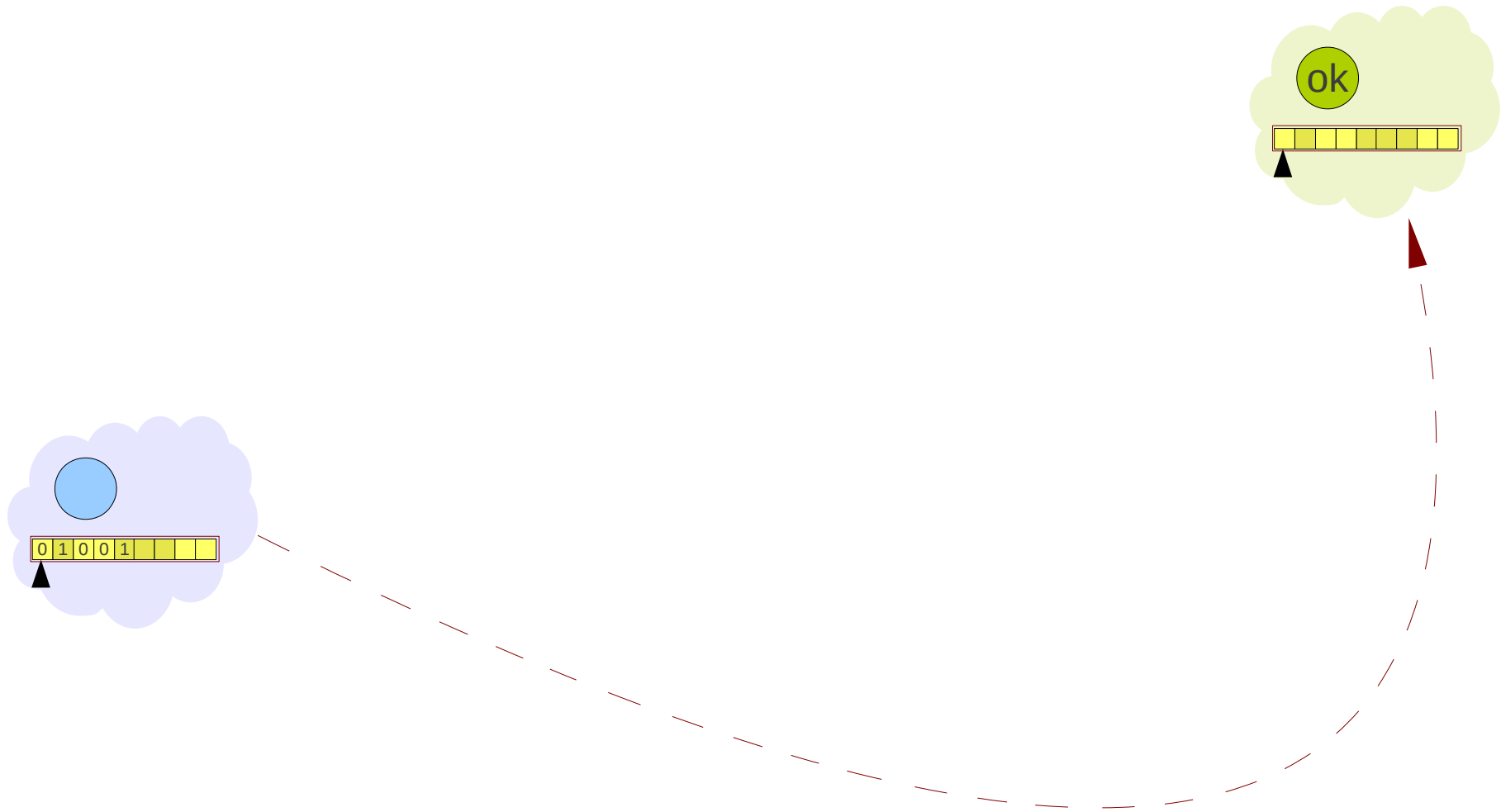
- Théorème de Savitch
- Paysage général
  - PSPACE
  - Stabilité par complémentaire
- Jeu à 2 joueurs
  - QBF
  - Jeu de géographie
- Machine alternante

# Preuve du théorème de Savitch

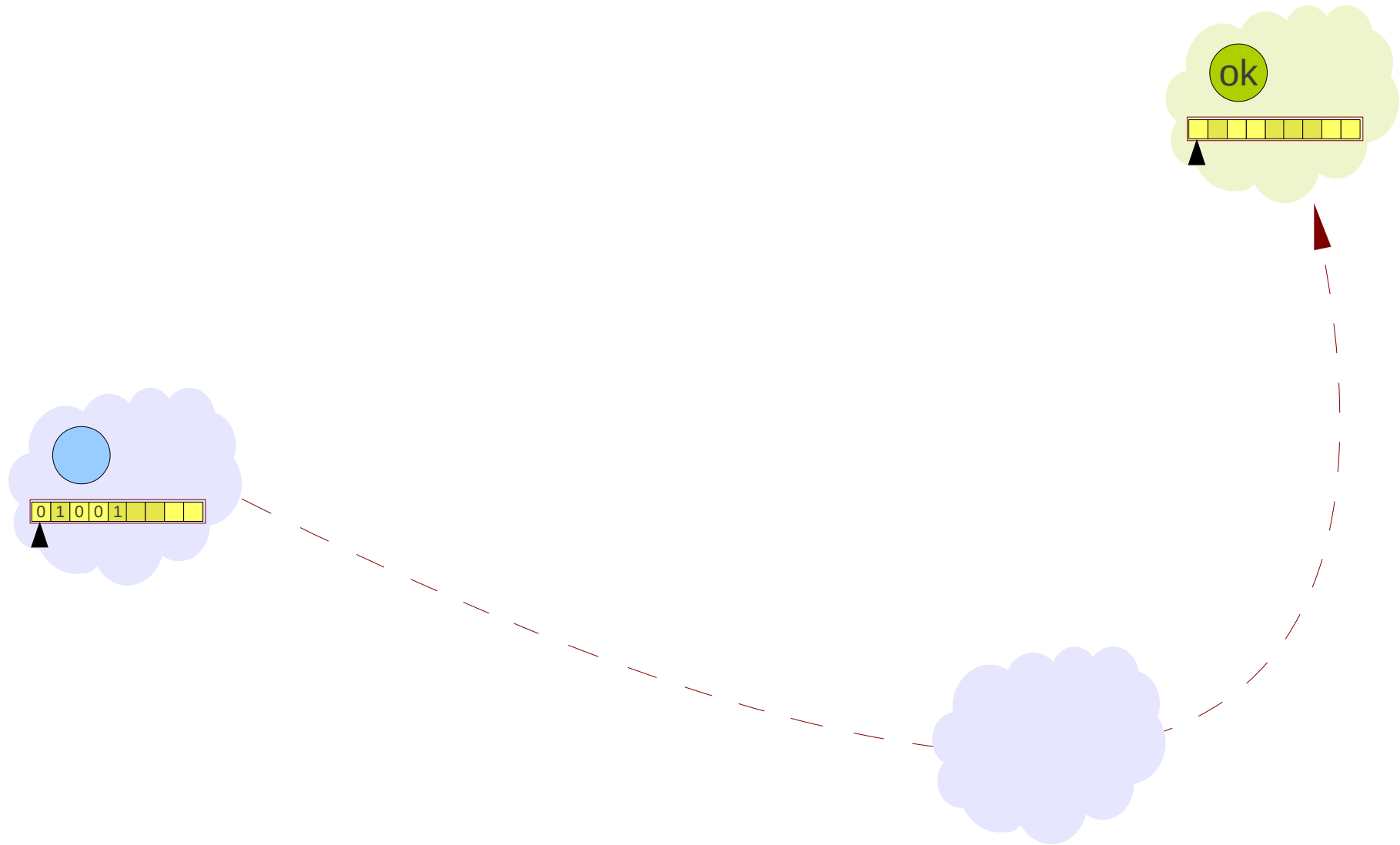




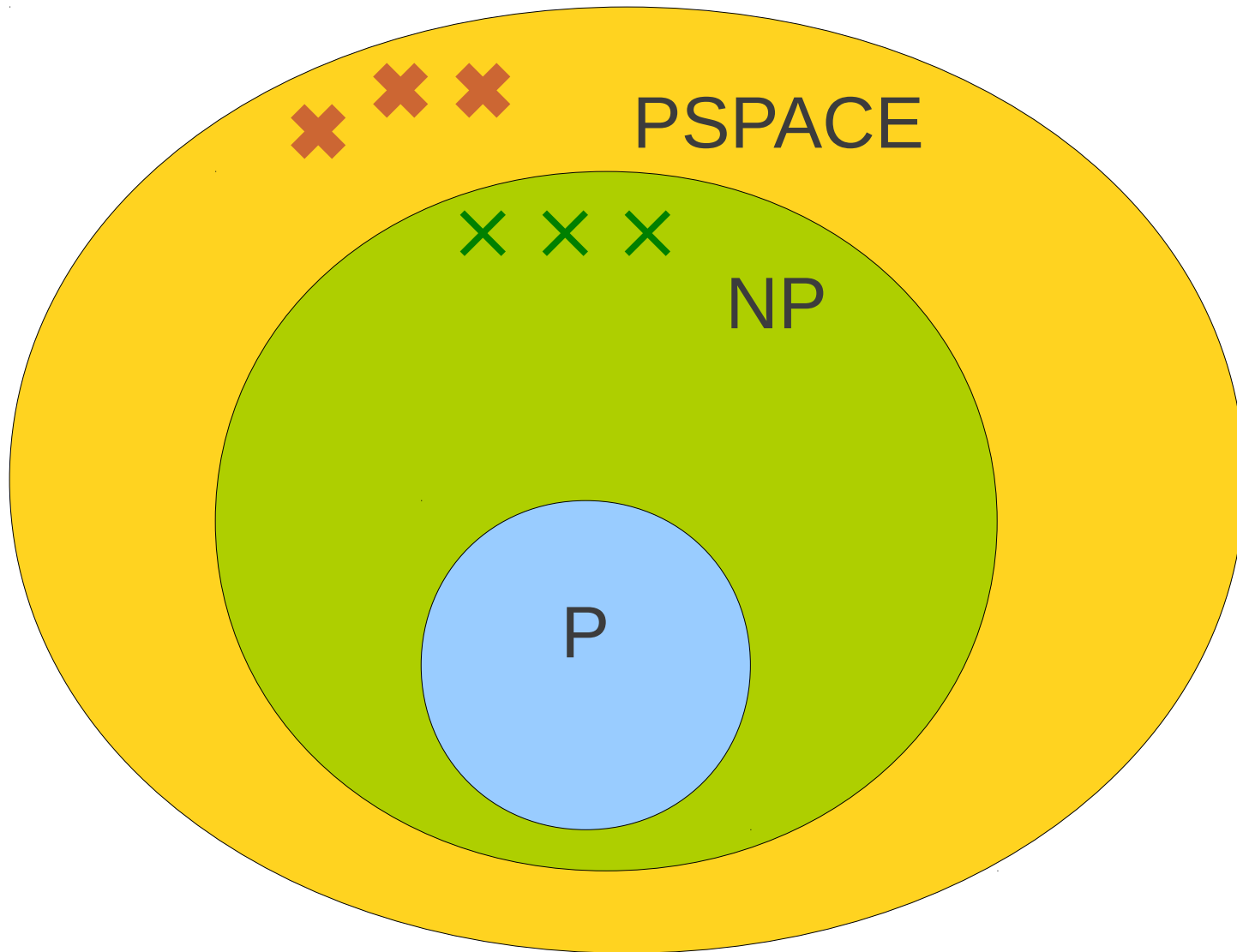
# Preuve du théorème de Savitch



# Preuve du théorème de Savitch



# Le paysage

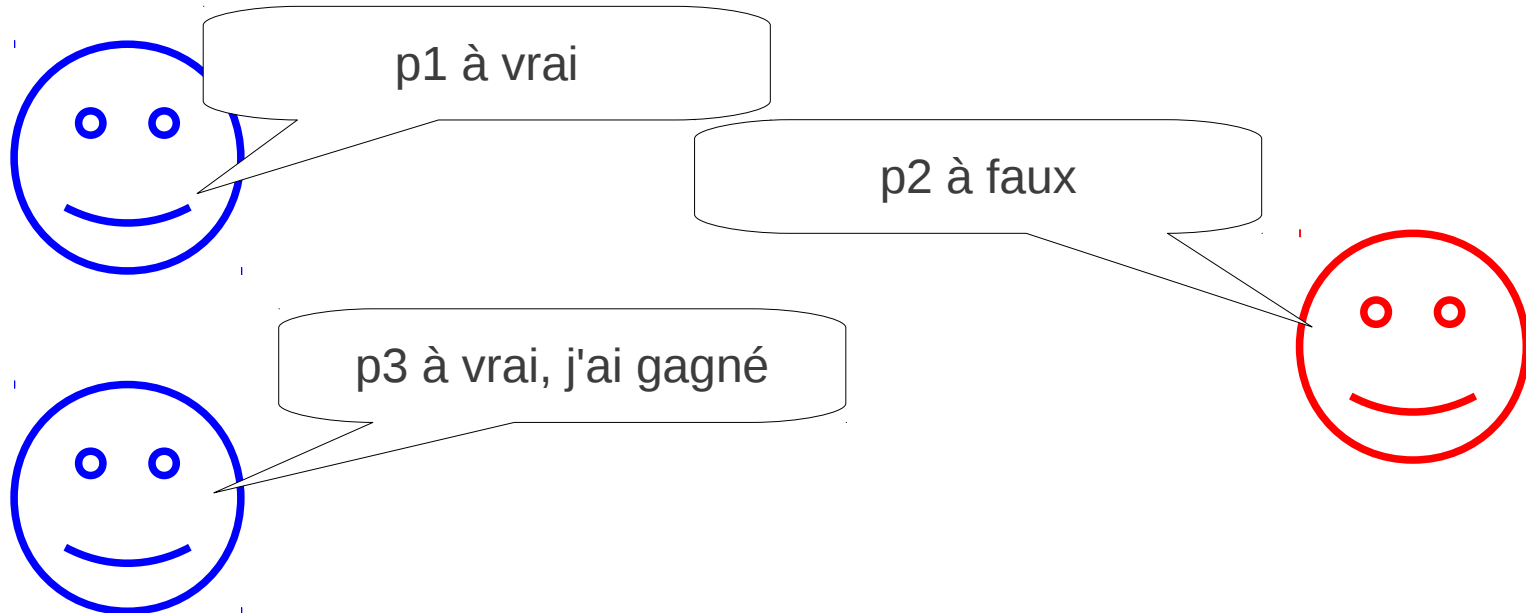


# Des problèmes PSPACE-complets

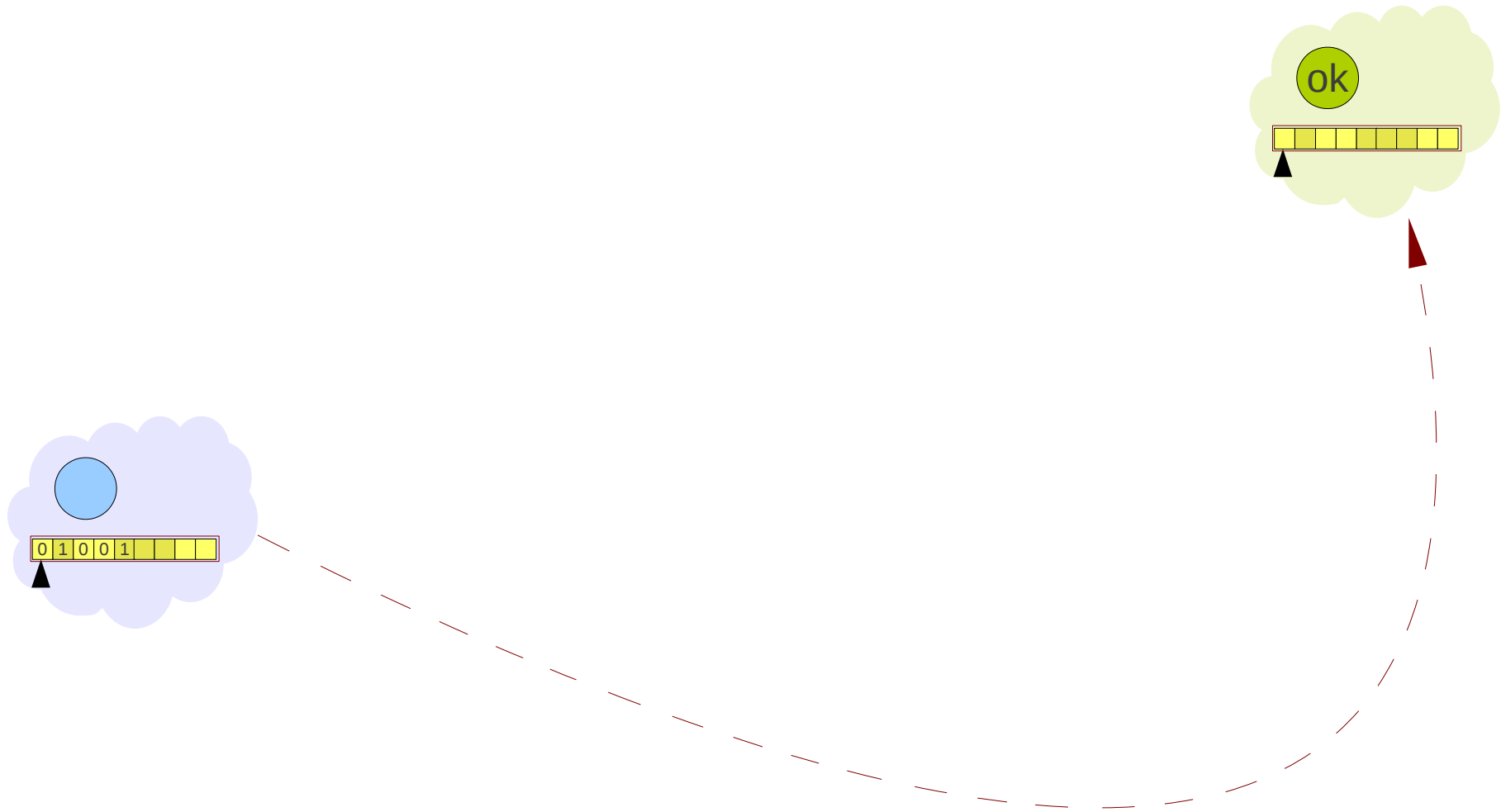
- Jeux à deux joueurs
- Voyager dans le temps (type \* de Kleene)

# Quantified binary formula

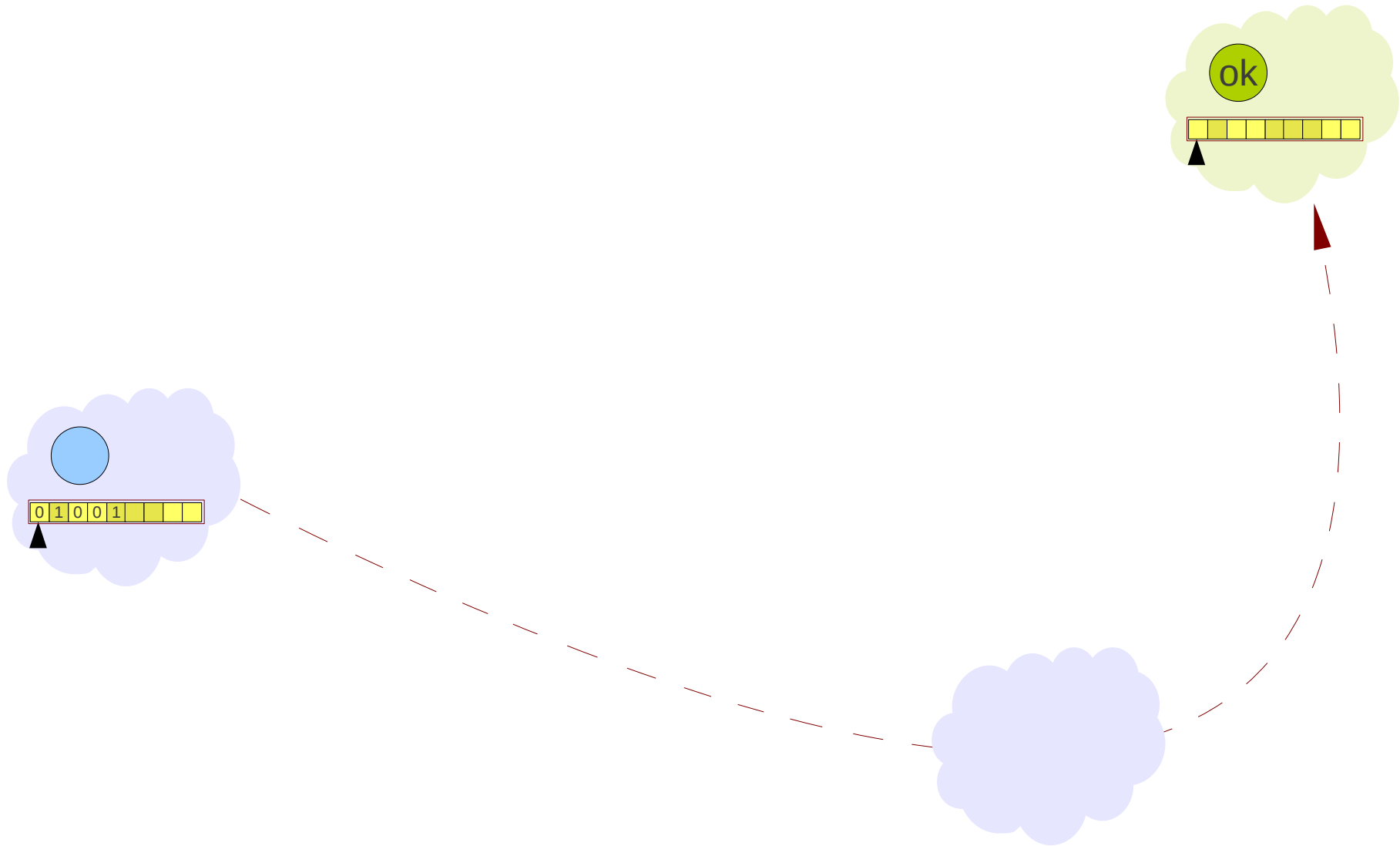
$$\phi = \exists p_1 \forall p_2 \exists p_3 \psi$$



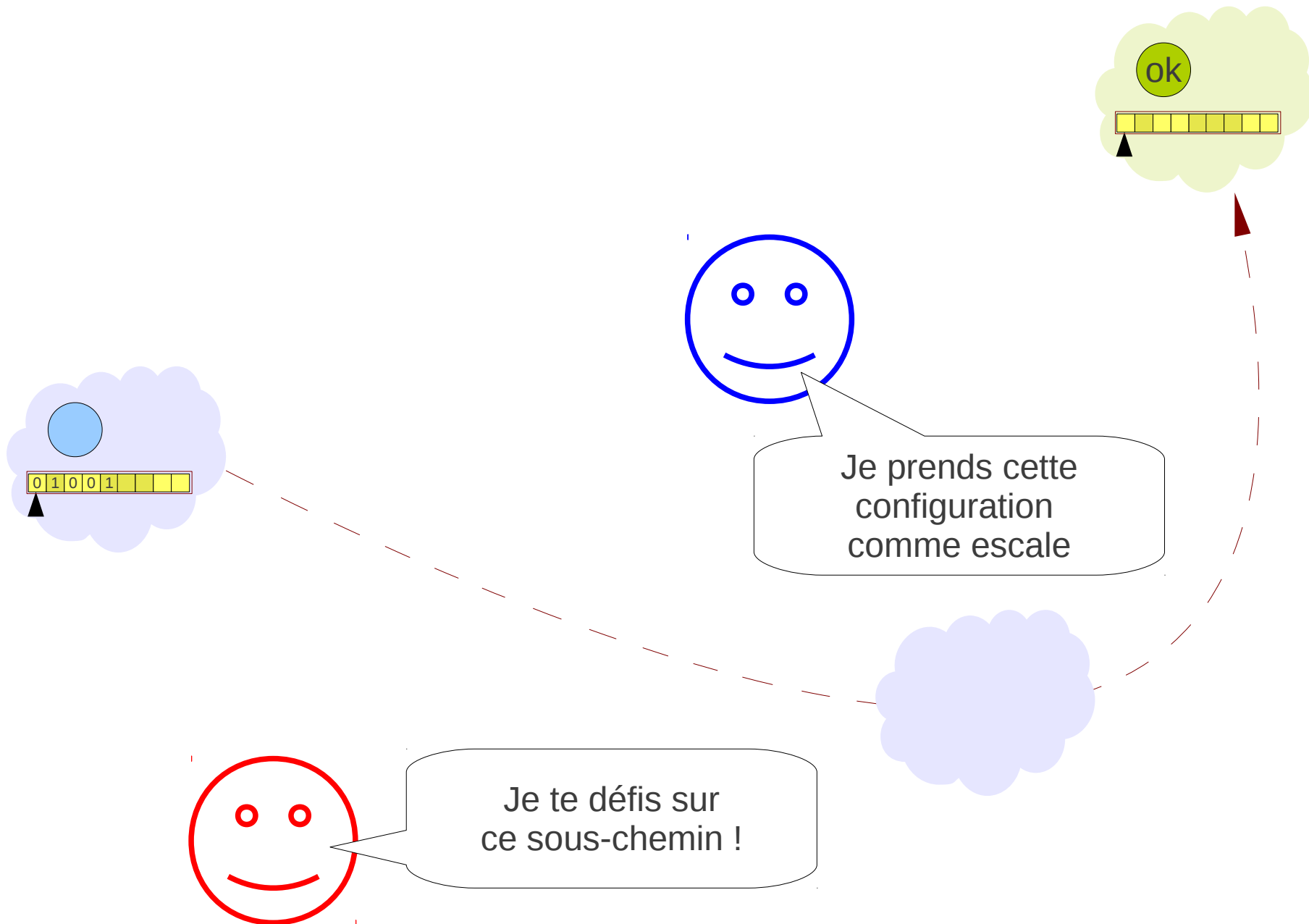
# Encoder l'existence d'un chemin



# Encoder l'existence d'un chemin

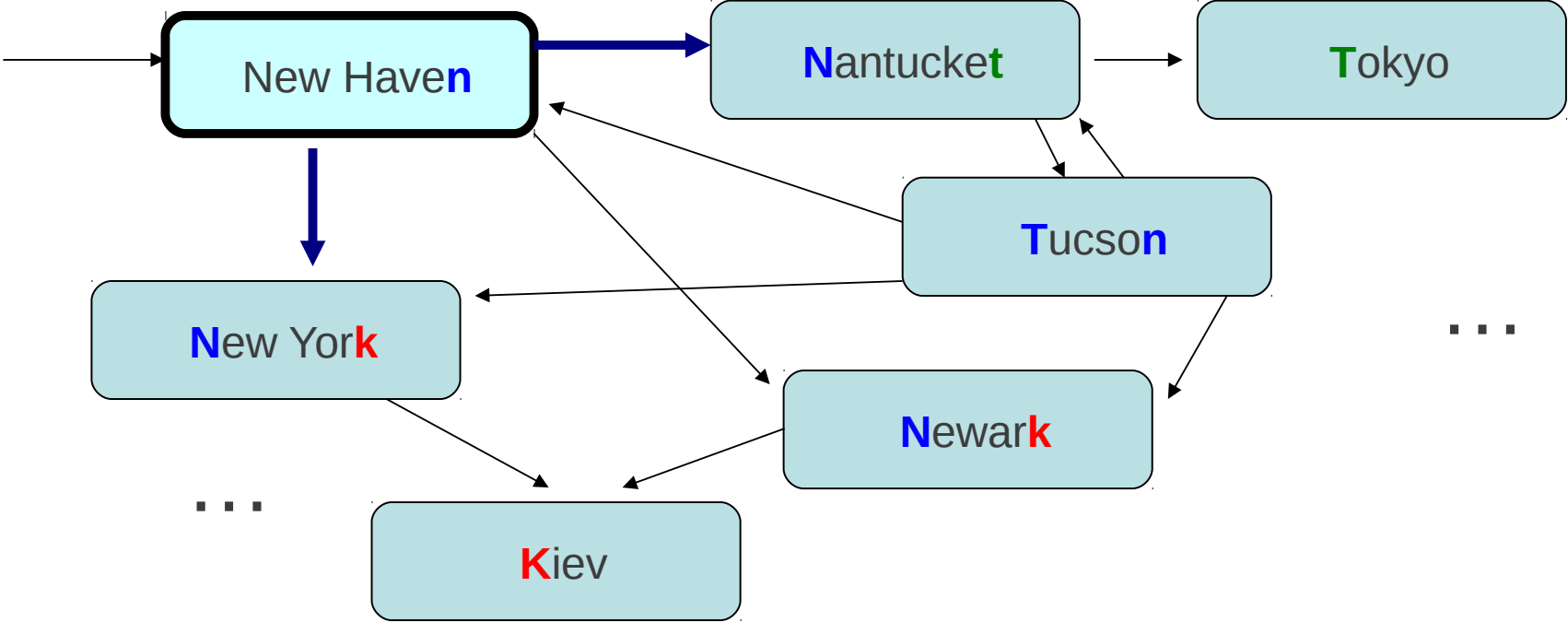


# Encoder l'existence d'un chemin

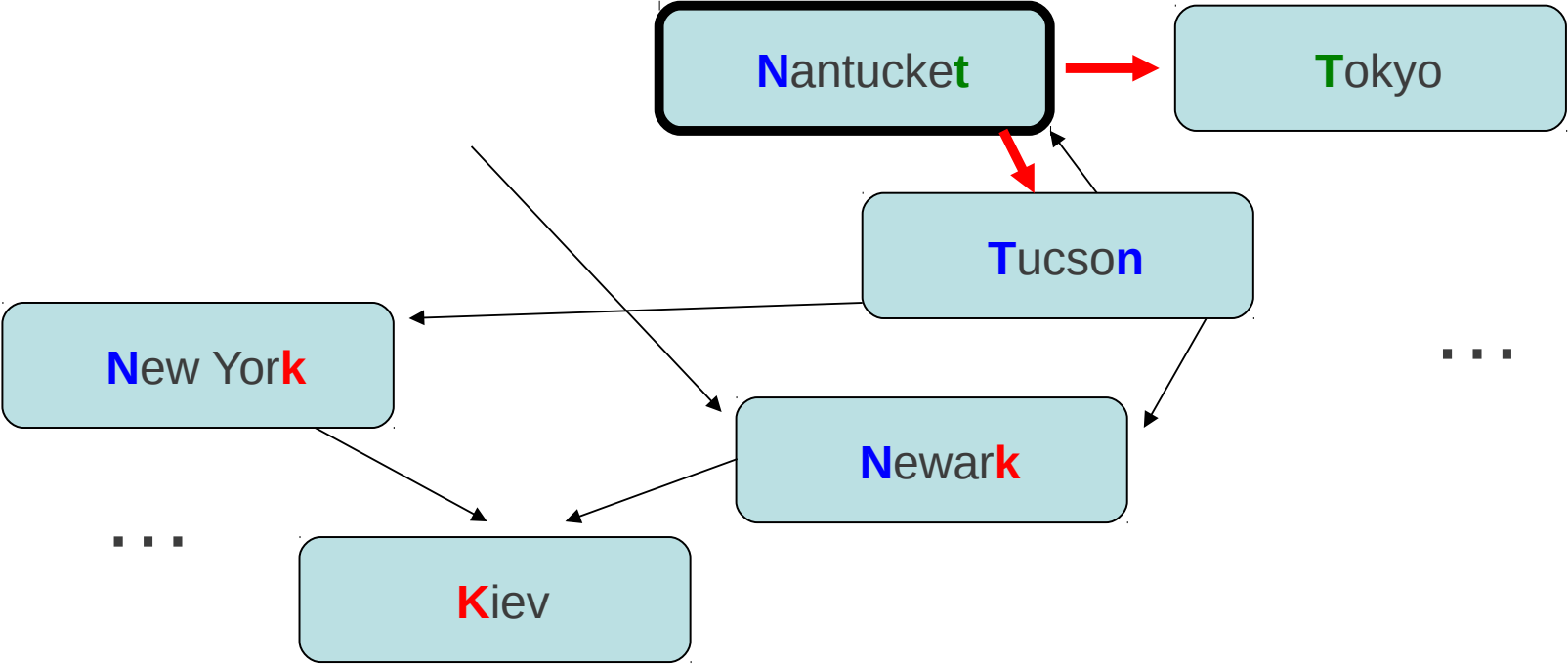




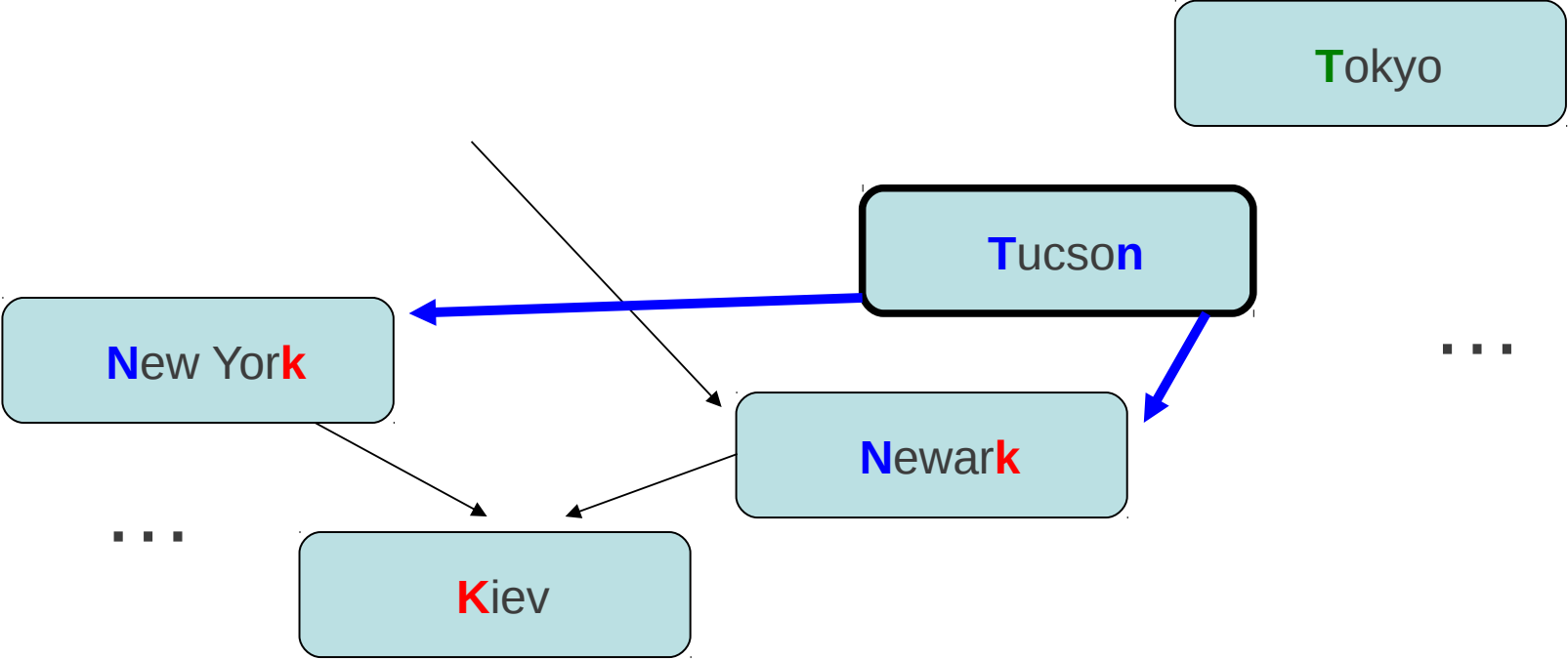
# Generalized Geography game



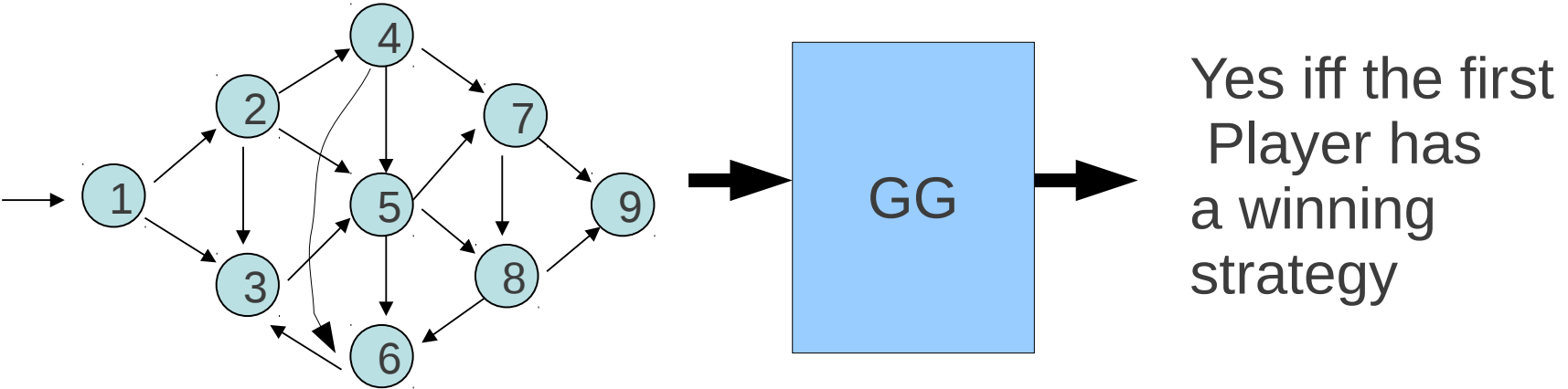
# Generalized Geography game



# Generalized Geography game

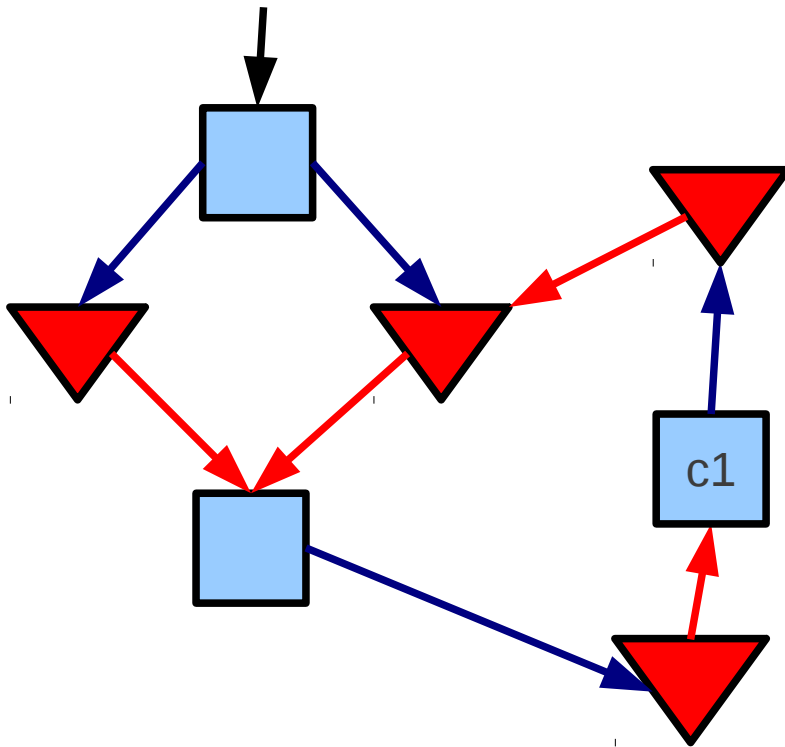


# Generalized Geography game



# Example

$$\phi = \exists p. p$$



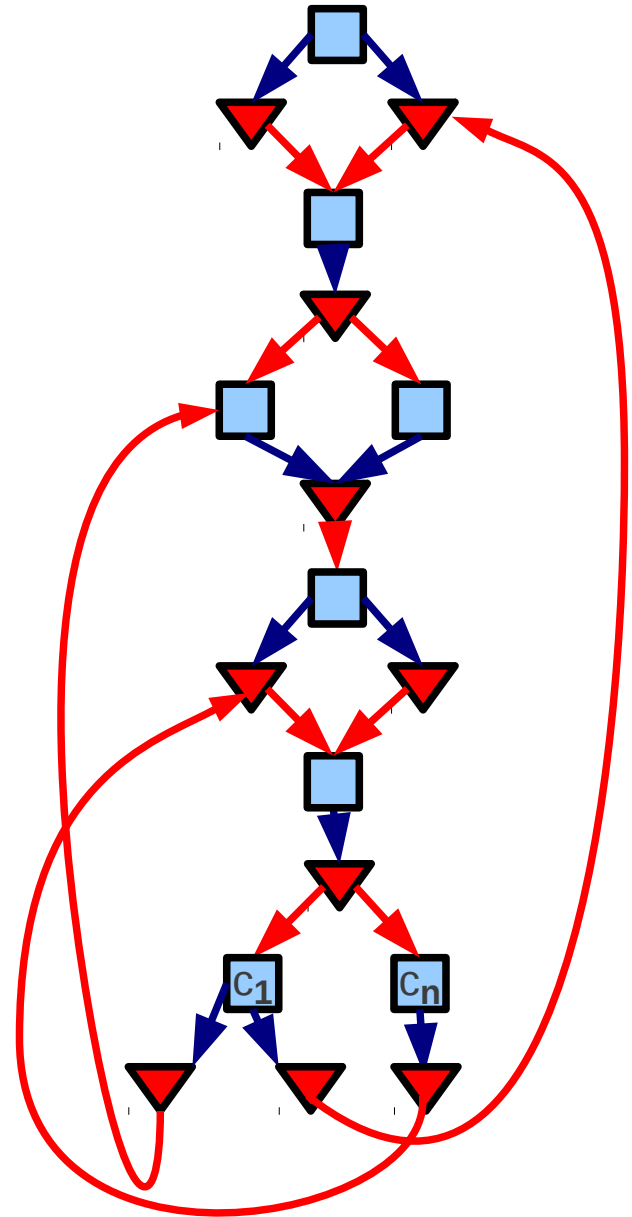
Bleu gagne

# Reduction from QBF to GG

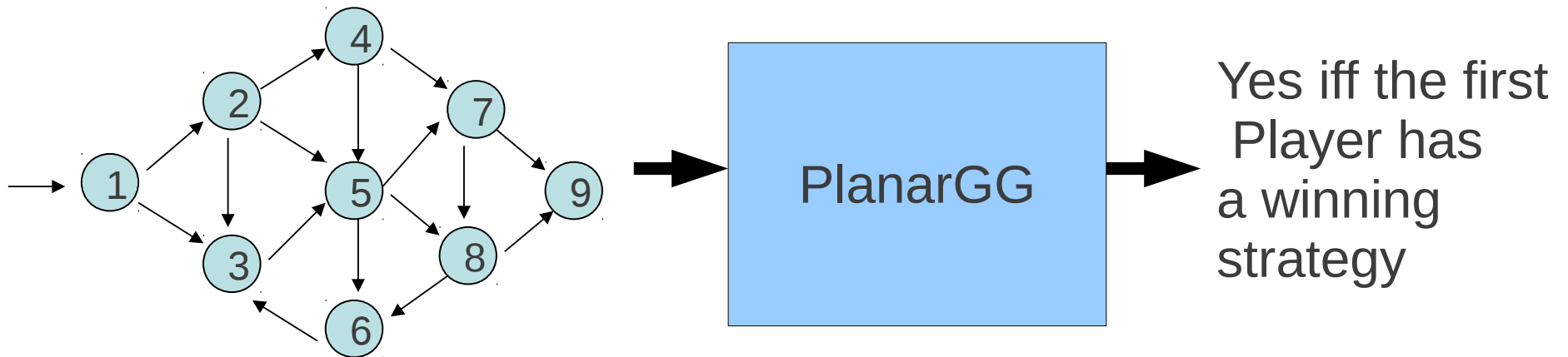
$$\phi = \exists x_1 \forall x_2 \exists x_3 \psi$$


$$c_1 \wedge \dots \wedge c_n$$

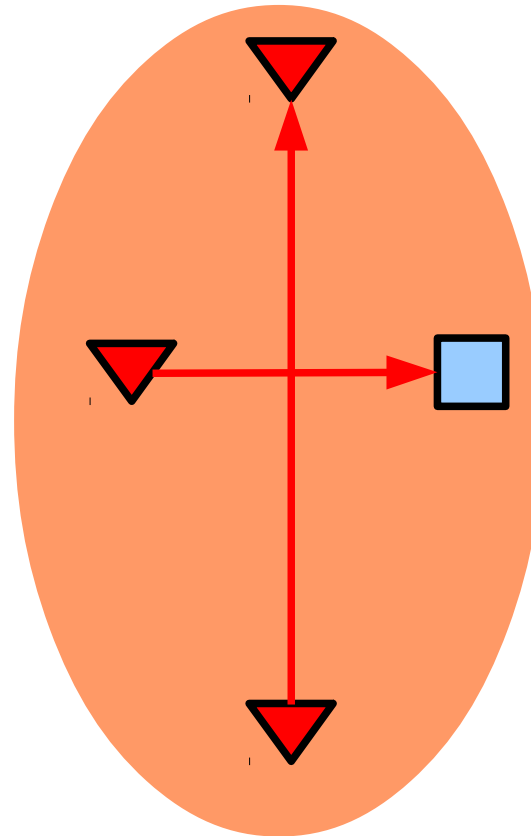
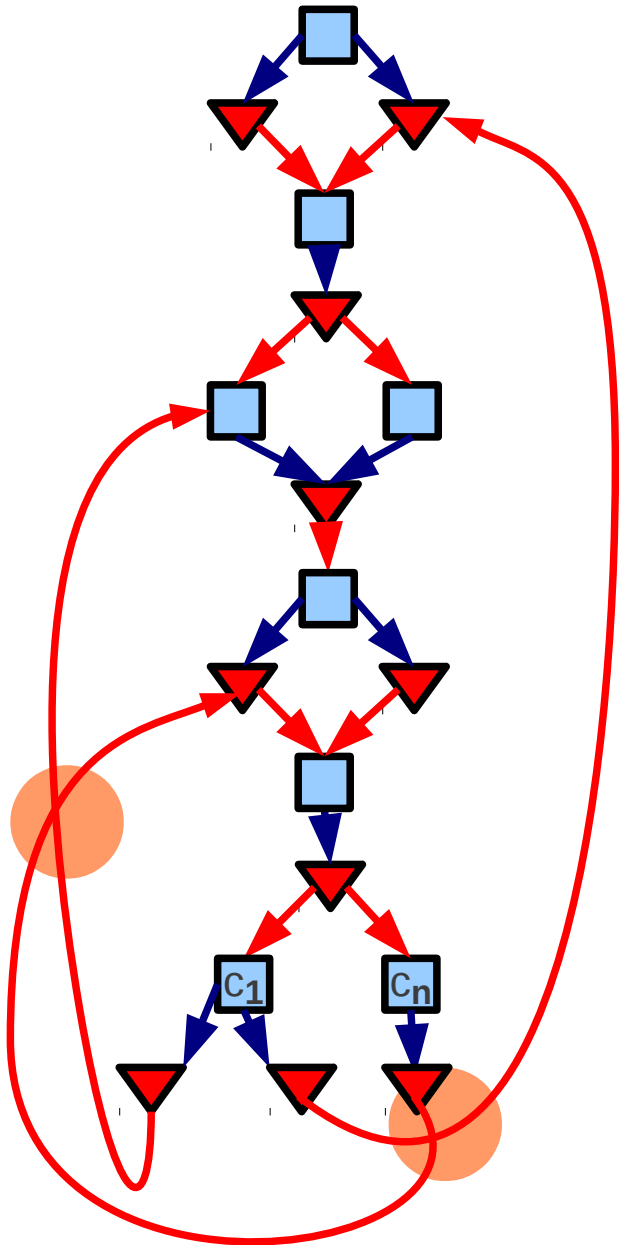
$\phi$  is true iff blue has winning strategy.



# Planar GG is also PSPACE-hard



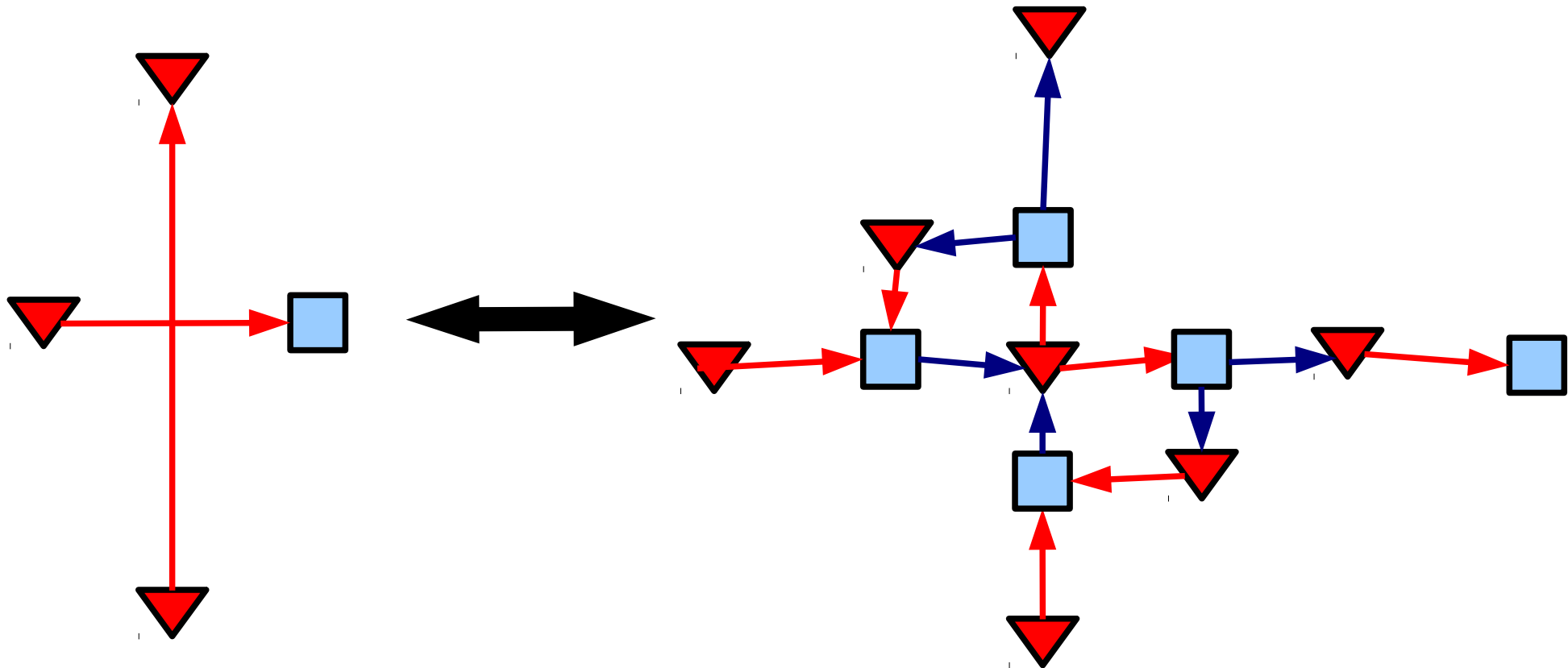
# Remove crossings



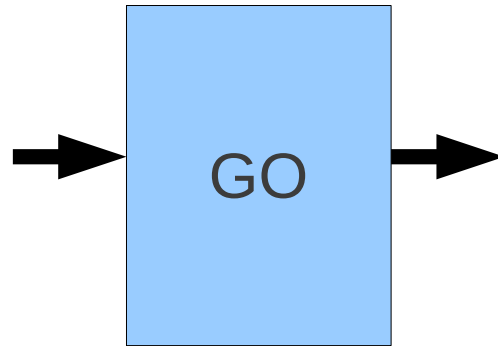
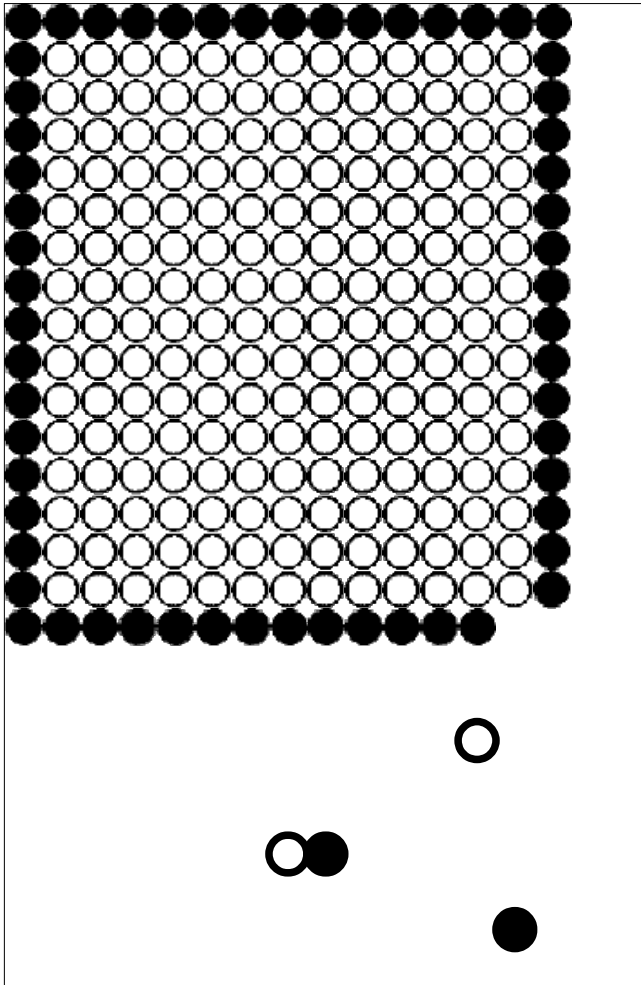
In the GG game, at most one edge will be used.



# Remove crossings



# Go winning strategy problem



Yes iff  
black has  
a winning  
strategy to take  
the white territory

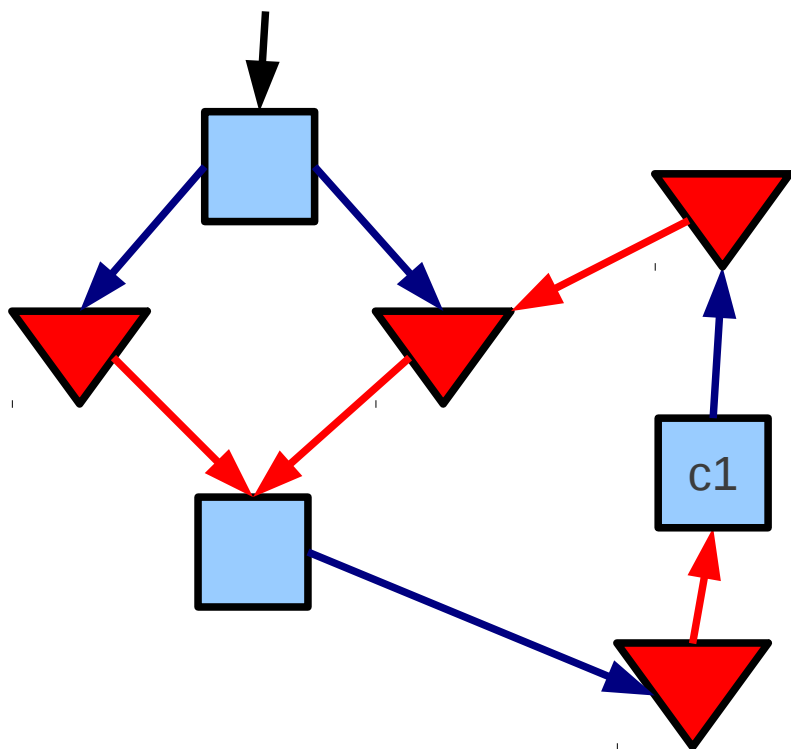
# GO is PSPACE-hard

Theorem : GO is PSPACE-hard.

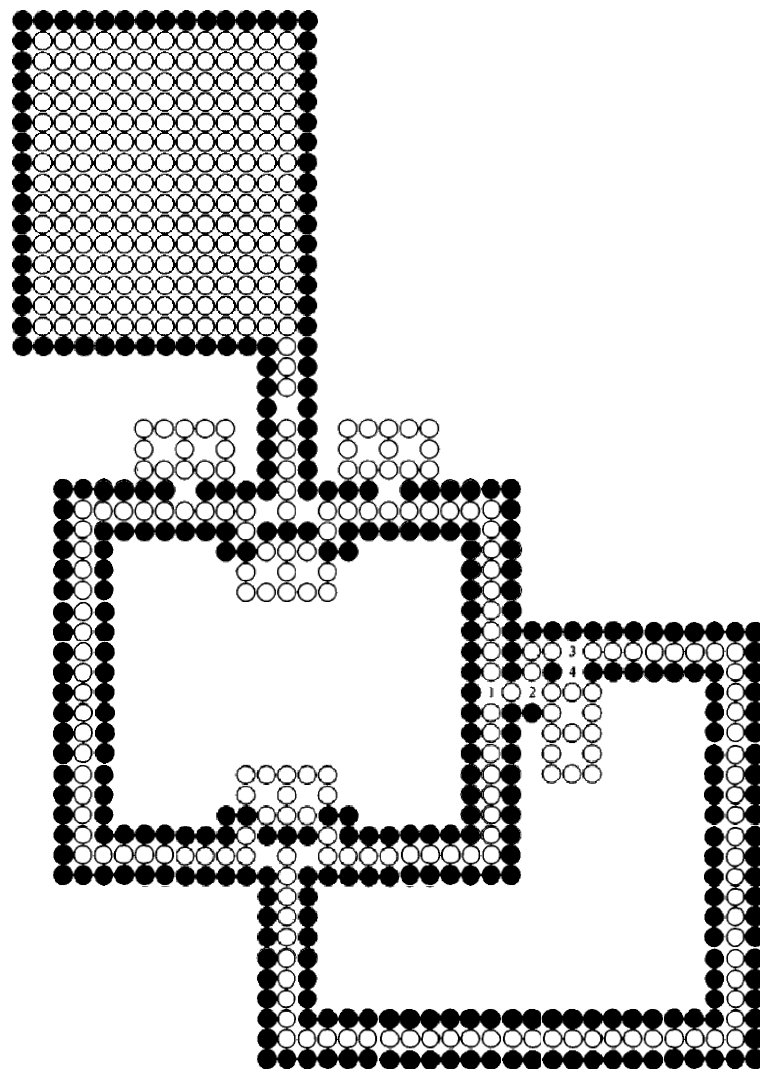
Proof : We reduce Planar GG to GO.

# We encode the graph on the goban.

$$\phi = \exists p. p$$

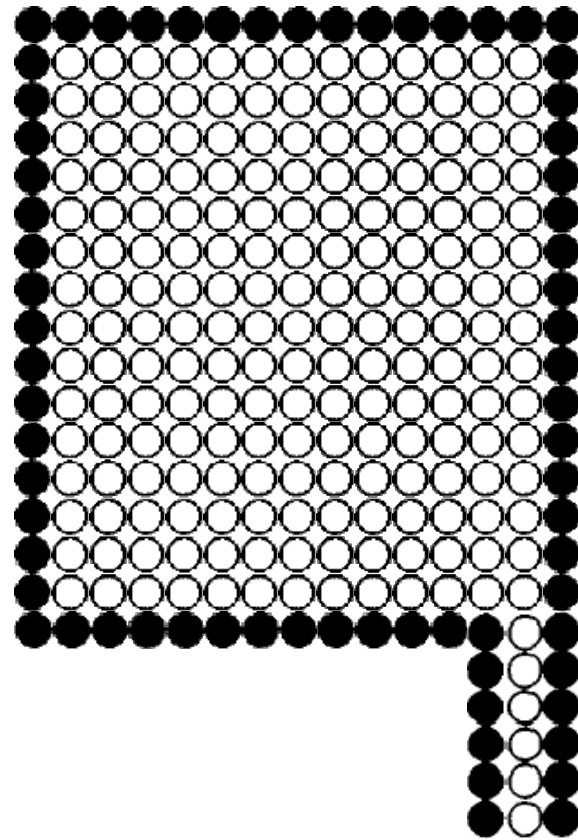
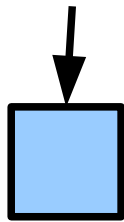


Bleu gagne

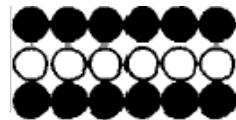
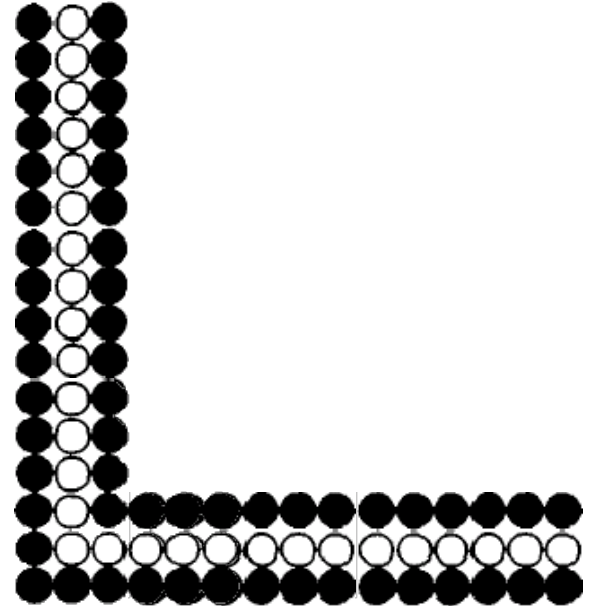
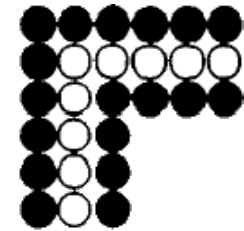
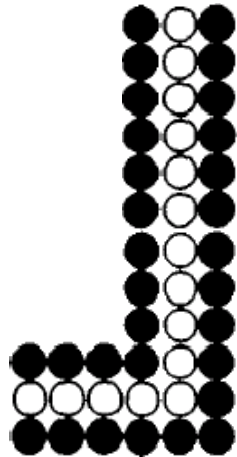
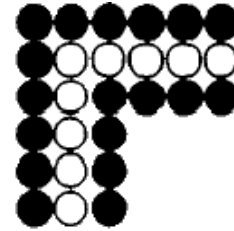
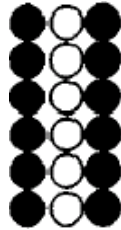
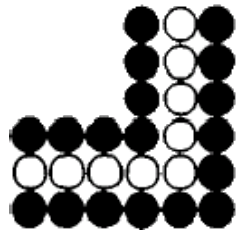


Noir gagne

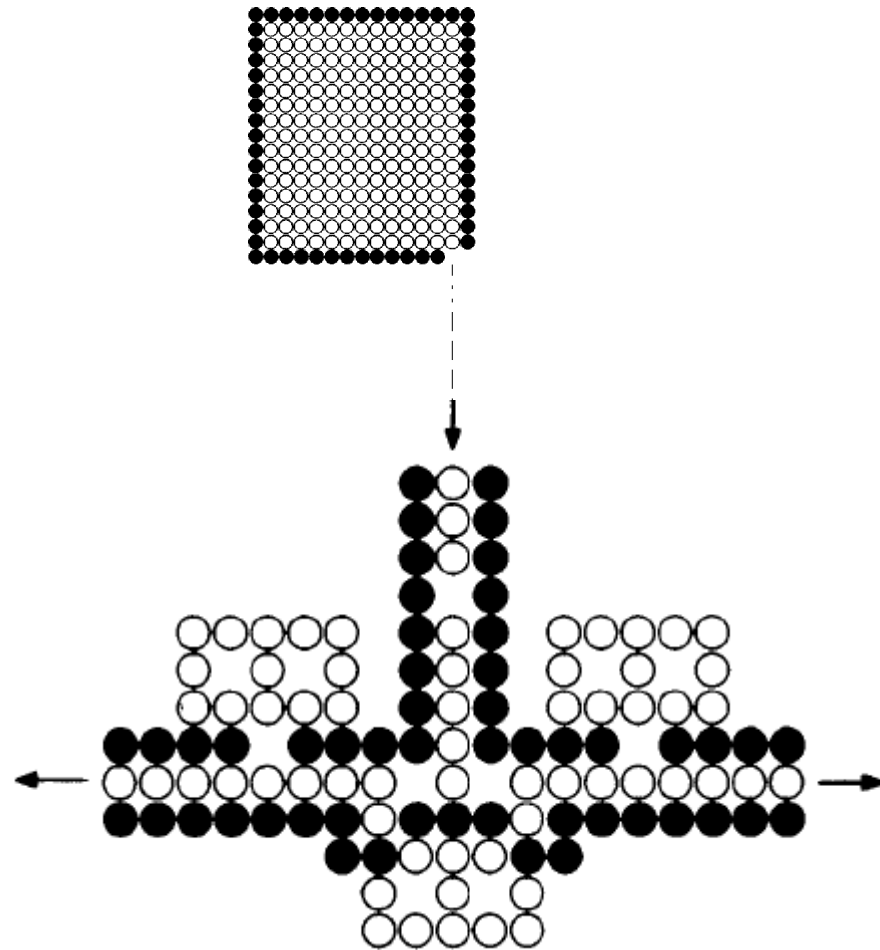
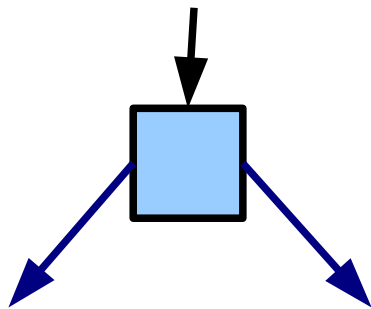
# Starting point



# Edges

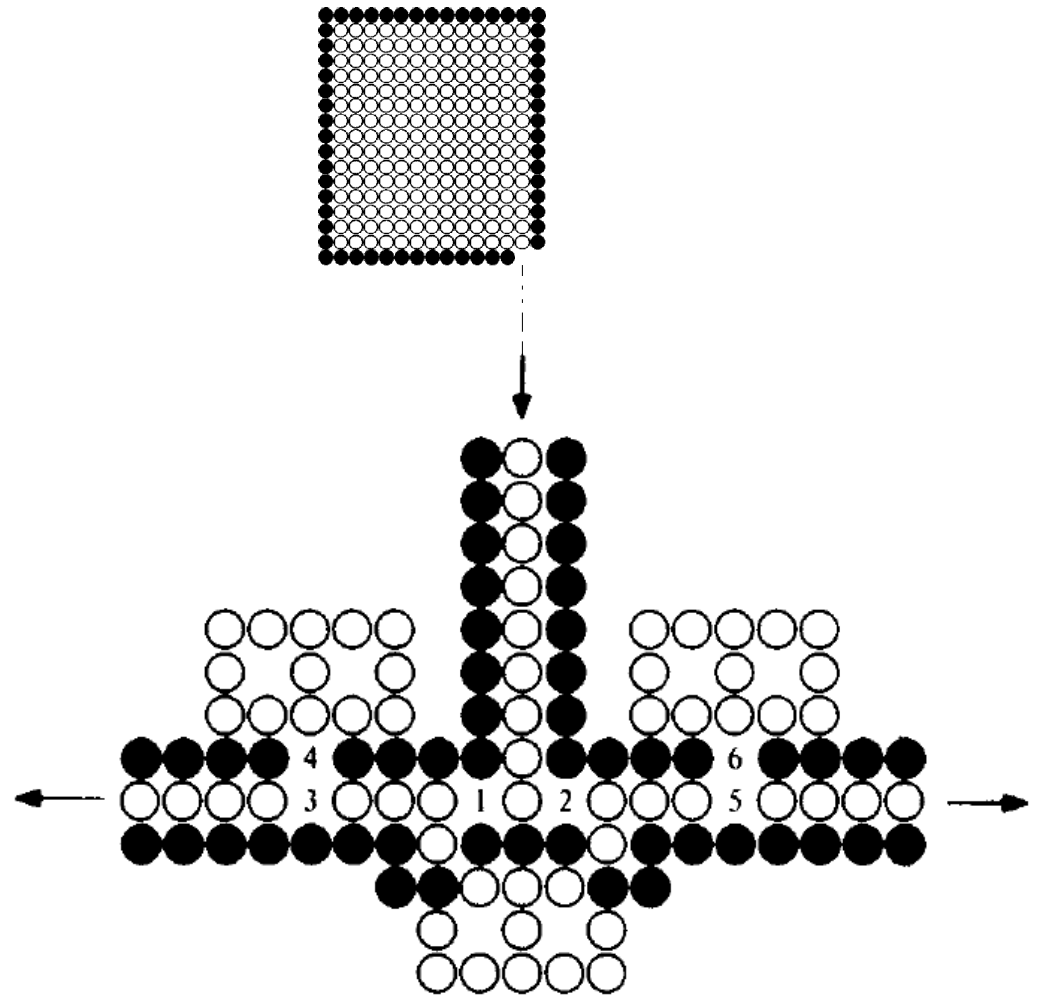
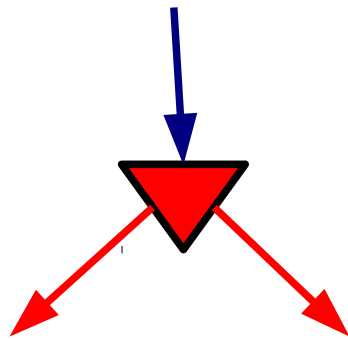


# Player E's choice



Noir décide le côté où on va

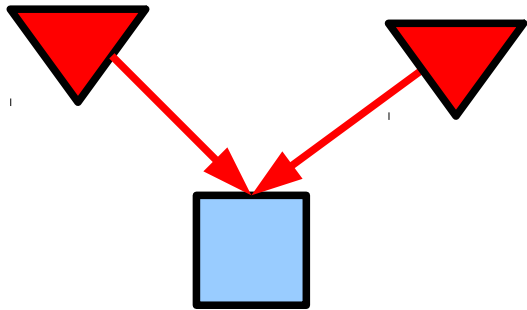
# Player A's choice



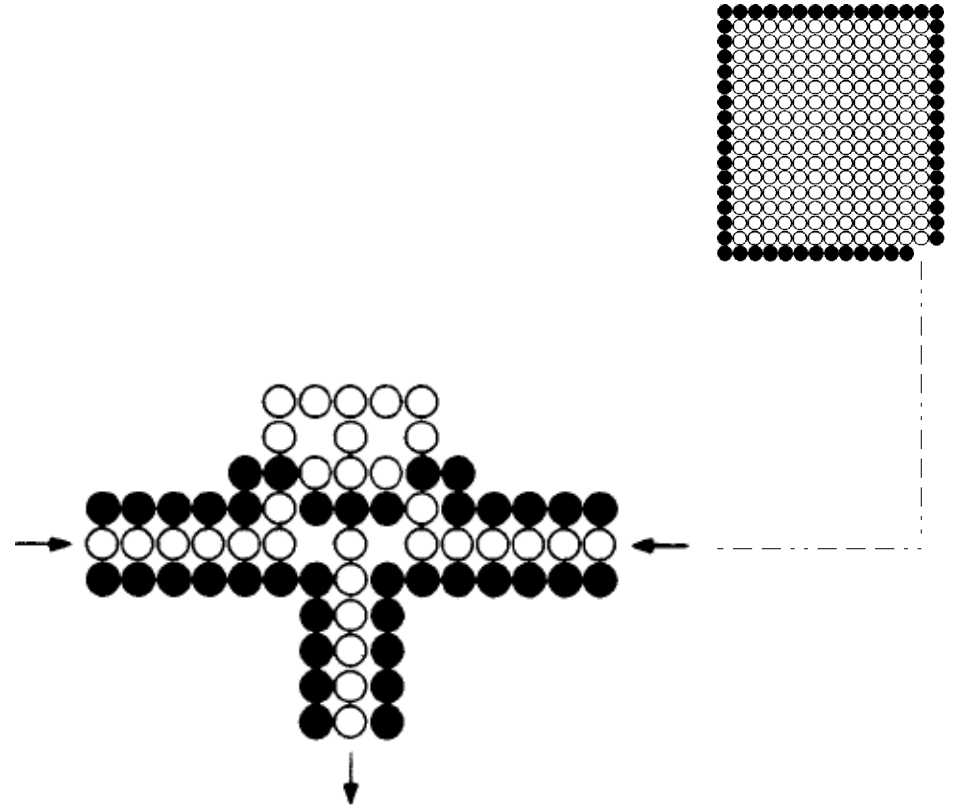
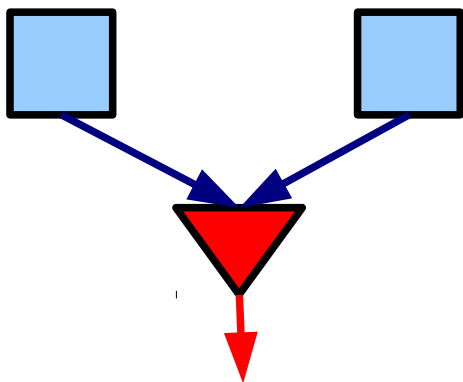
Blanc décide le côté où on va



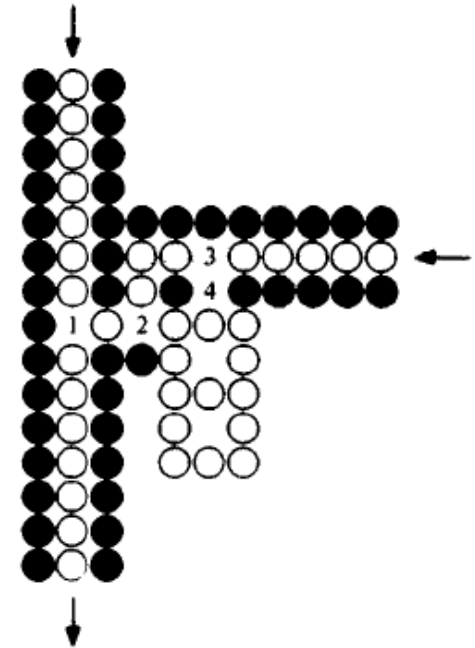
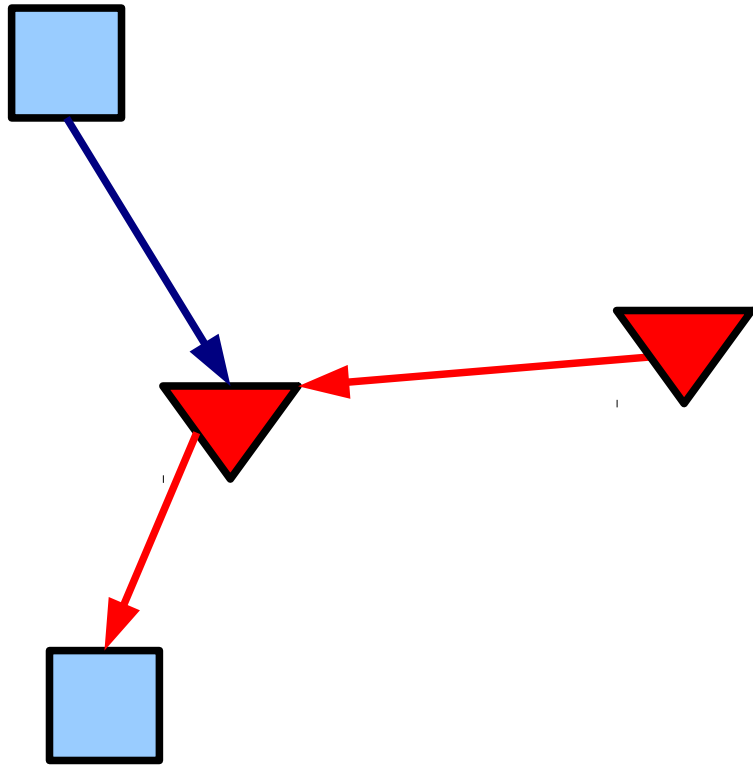
# Join



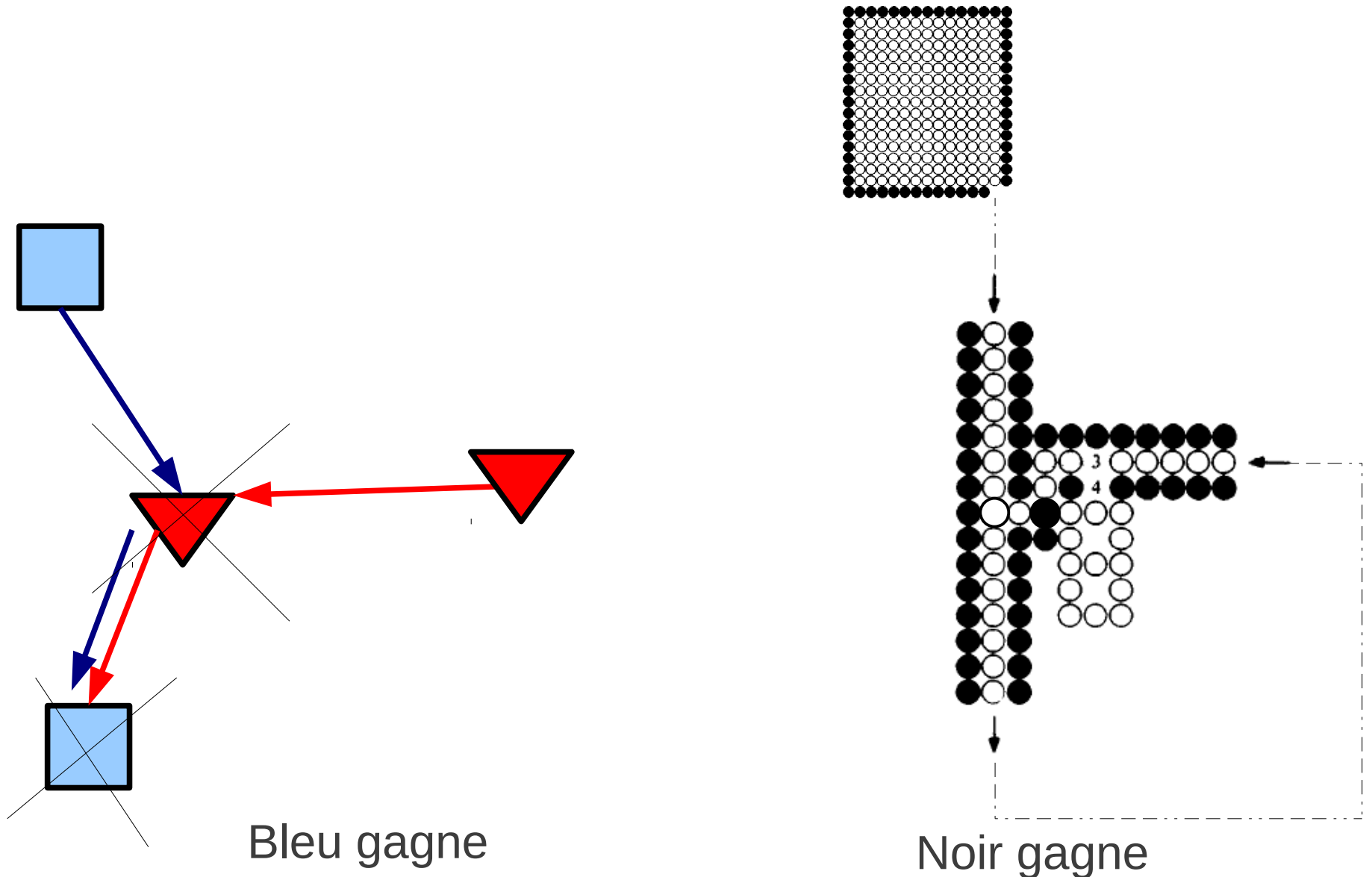
or



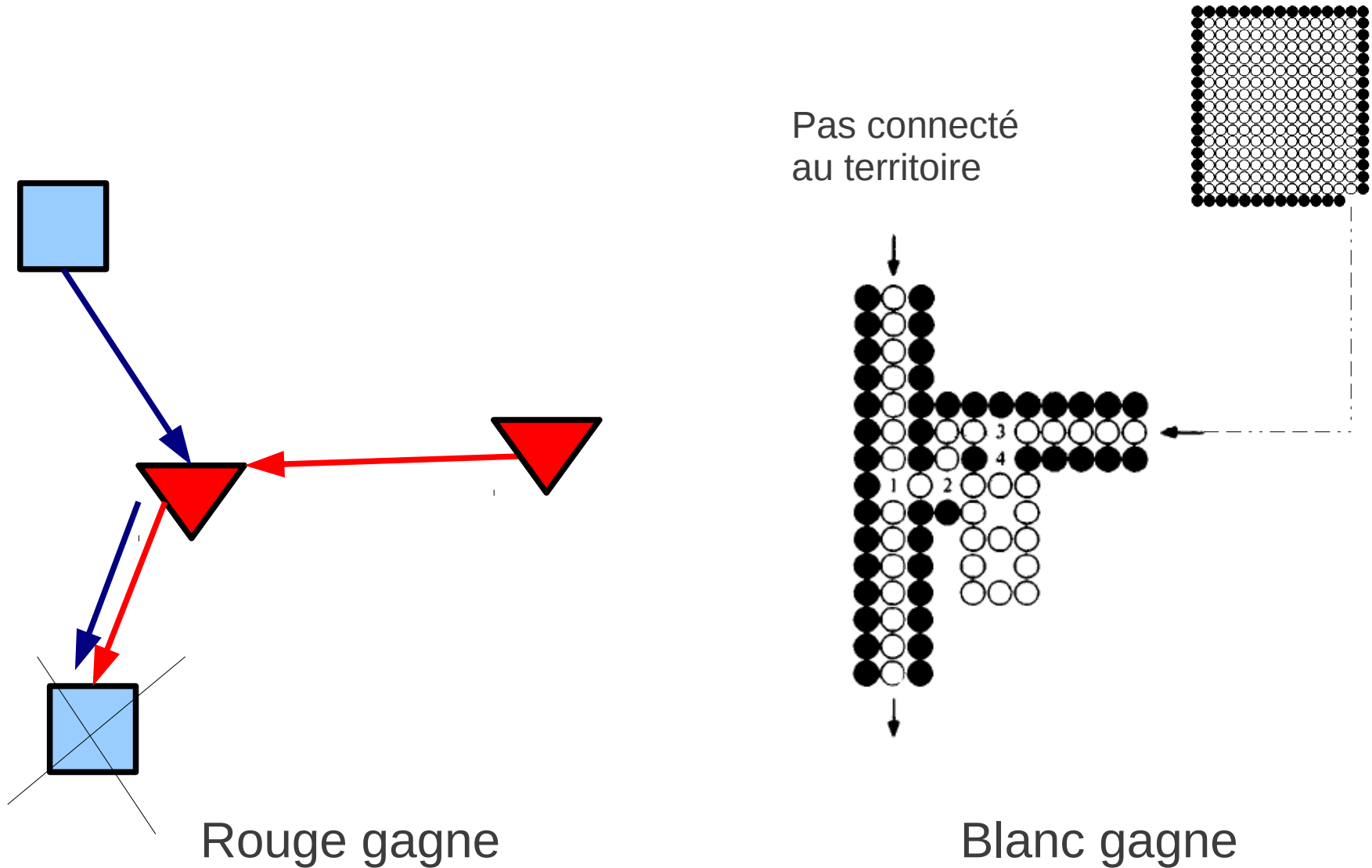
# Test



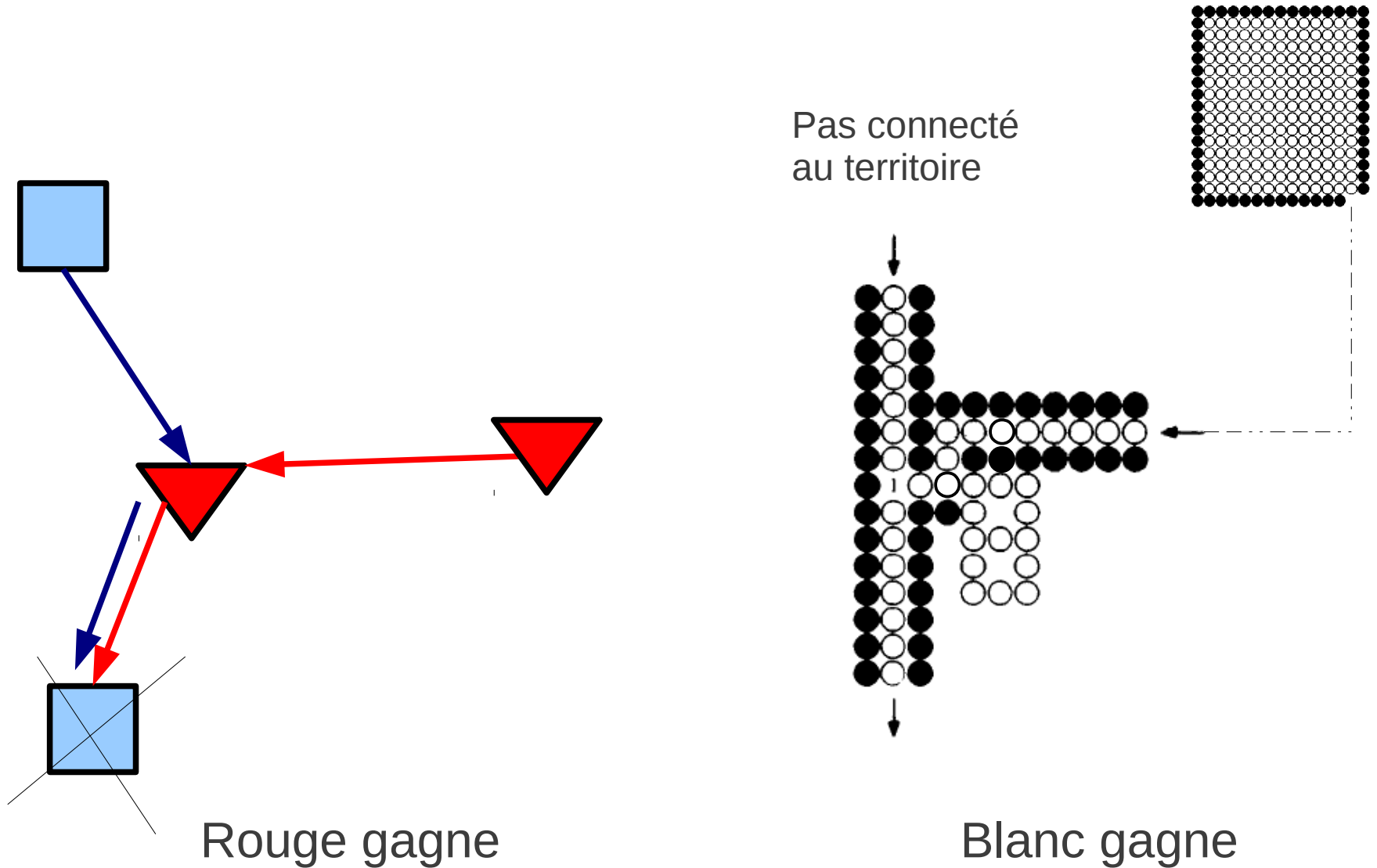
# Test : cas où connecté au territoire



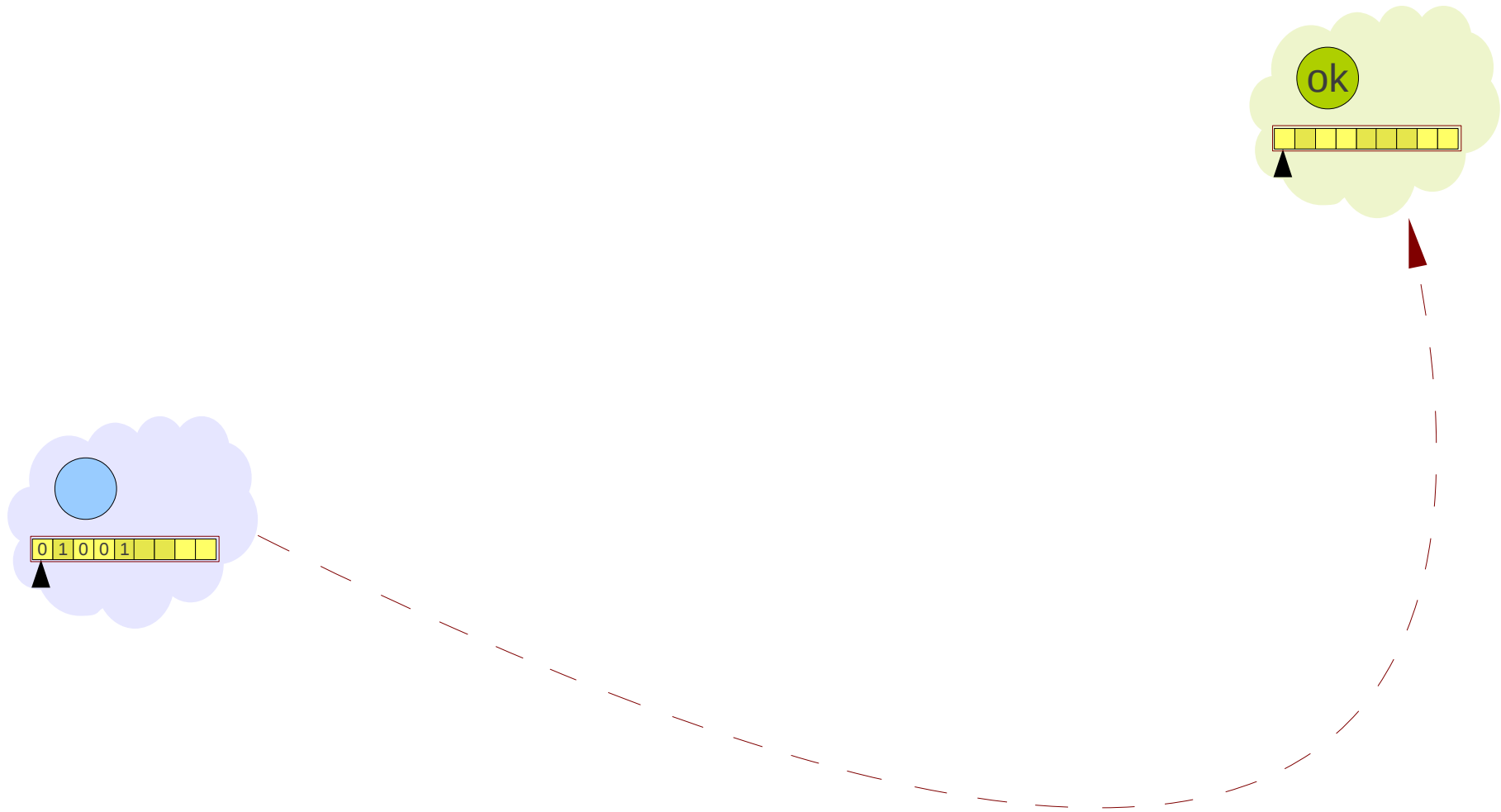
# Cas où non connecté au territoire



# Cas où non connecté au territoire



# Utiliser \* de Kleene



# Utiliser \* de Kleene

(q0, 0)	1	0	0	1				
1	(q1, 1)	0	0	1				

- 
- 
- 

(ok, )								
--------	--	--	--	--	--	--	--	--

(q0, 0)1001 \_ \_ \_ \_ 1(q1, 1)0 0 1 \_ \_ \_ \_ ..... (ok, \_) \_ \_ \_ \_ \_