Le Sokoban est PSPACE-complet

TIPE d'informatique pour les Écoles Normales Supérieures

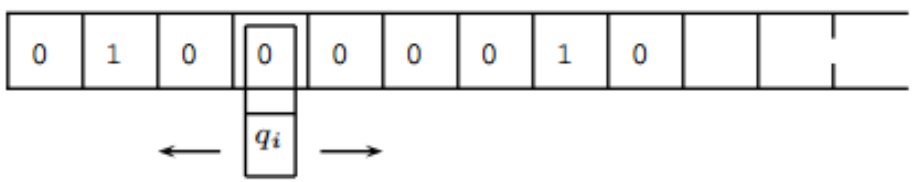
Jonathan Laurent

Juin 2013

Définition

Une machine de Turing est un sextuplet $(Q, \Sigma, \Gamma, q_0, F, \delta)$ où

- Q est un ensemble fini d'états
- Σ est un alphabet fini dit *alphabet d'entrée*. On prendra $\Sigma = \{0, 1\}$
- $\Gamma = \Sigma \cup \{\sqcup\}$ est l'*alphabet de travail*
- q_0 est l'état initial
- $F \subset Q$ un ensemble d'états bloquants. On prendra $F = \{q_+, q_-\}$
- $\delta : \Gamma \times (Q \setminus F) \longrightarrow \Gamma \times Q \times \{\leftarrow, \rightarrow\}$ est dite fonction de transition



La bande est indicée sur $\mathbb{N}$.
On convient qu'une chaine est rejetée si la tête de lecture tente de sortir de la bande au cours du calcul associé.

Classes de complexité

Et relation de difficulté sur les langages

Définition

On définit les fonctions de complexité en temps et en espace

- $T_M : n \mapsto \sup_{|x| \leqslant n} t_M(x)$
- $S_M : n \mapsto \sup_{|x| \leqslant n} s_M(x)$

où $t_M(x)$ représente le nombre de transitions effectuées par M avant arrêt sur l'entrée x, et $s_M(x)$ la longueur de bande alors consommée.

On introduit naturellement les classes P et $PSPACE$:

- $P = \{ L \mid \exists M,\ M \text{ décide } L \text{ et } T_M \text{ est bornée par un polynôme} \}$
- $PSPACE = \{ L \mid \exists M,\ M \text{ décide } L \text{ et } S_M \text{ est bornée par un polynôme} \}$

Définition

On note $L \prec_P L'$ lorsque il existe $f : \mathbb{B} \longrightarrow \mathbb{B}$ fonction calculable polynomiale telle que $\forall x \in \mathbb{B},\ x \in L \iff f(x) \in L'$.

On dit également que L est réductible en temps polynomial à L'.

Position du problème

Règles du jeu :

- Le joueur doit pousser toutes les caisses sur des cases spéciales dites *objectifs*.
- Il est impossible de tirer une caisse ou d'en pousser deux d'un même mouvement.

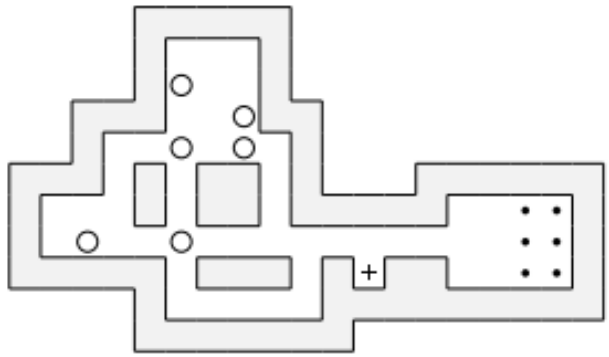


Figure : Le premier niveau du jeu original, par *Thinking Rabbit Inc.*

On définit le langage *SOK* des chaines codant pour un niveau de sokoban qui admet une solution.

À un niveau on peut associer son graphe des configurations :

- Une configuration est la donnée d'une répartition de caisses et de la position du magasinier.
- Chaque arrête indique une transition réalisable en un unique mouvement autorisé.

Le sokoban est PSPACE

Le problème du sokoban se ramène à un problème d'accessibilité dans un graphe, pour lequel on dispose d'un algorithme économe en espace :

La procédure *Access* prend pour paramètres deux configurations c et c' et un entier t. Elle retourne *vrai* si et seulement si il existe un chemin de c à c' de longueur au plus 2^t.

```
ACCESS(c, c', t) =
  if t = 0 then
    return c = c' or (c, c') ∈ E
  else
    for all c'' ∈ V do
      if ACCESS(c, c'', t − 1) and ACCESS(c'', c', t − 1) then
        return true
      end if
    end for
    return false
  end if
```

Un niveau dont la grille comporte n cases admet une solution si et seulement si il existe une configuration finale f telle que $\mathit{Access}\,(c_0, f, \lceil \log_2 (n \cdot 2^n) \rceil)$.

On a donc un algorithme pour SOK qui s'exécute en espace $O(n^2)$.

Présentation du problème QSAT

Définition

Soient $x_1, \cdots, x_n$ variables booléennes.

- On appelle *littéral* une expression du type x_i ou $\neg x_i$ pour $1 \leqslant i \leqslant n$.
- Une *clause* est une disjonction de littéraux.
- Une *CNF* est une conjonction de clauses.

Soit $\psi(x_1, \cdots, x_n)$ une CNF et $Q_i \in \{\forall, \exists\}$ pour $1 \leqslant i \leqslant n$.
Alors $Q_n x_n \cdots Q_1 x_1 \ \psi(x_1, \cdots, x_n)$ est dite *QCNF*.

Une telle expression s'évalue en 0 ou en 1 de la manière suivante, par récurrence :

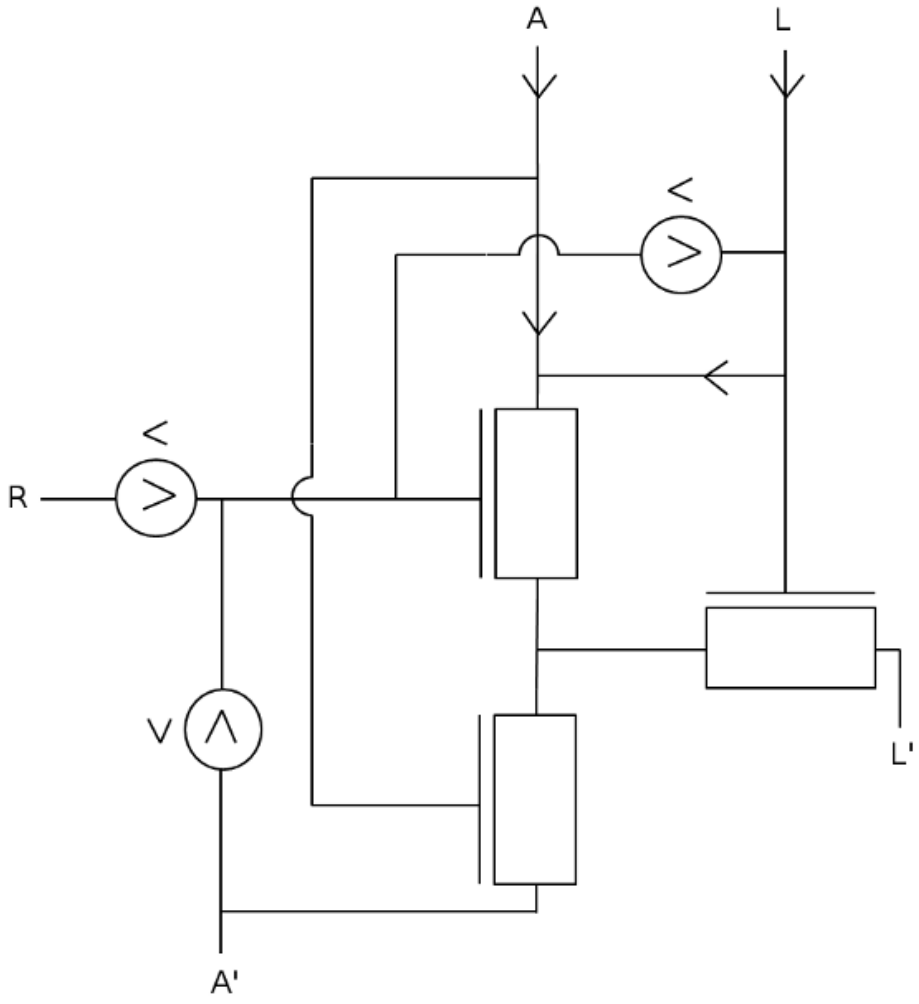
Pour $F = Q_n x_n \cdots Q_1 x_1 \ \psi(x_1, \cdots, x_n)$, on pose

$F_0 = Q_{n-1} x_{n-1} \cdots Q_1 x_1 \ \psi(x_1, \cdots x_{n-1}, 0)$
$F_1 = Q_{n-1} x_{n-1} \cdots Q_1 x_1 \ \psi(x_1, \cdots, x_{n-1}, 1)$

- Si $Q_n = \forall$, on a $F = F_0 \wedge F_1$
- Si $Q_n = \exists$, on a $F = F_0 \vee F_1$

Exemple : $\forall x \ \exists y \ \forall z \ (x \vee y \vee z) \ = 1$

Construction du verrou



	A	A’	L	L’	R
0	-	-	L’0	-	R1
1	A’0 A’1	-	L’0	-	R1

Réduction de QSAT vers SOK

Principe :

Étant donné que *QSAT* est *PSPACE* complet, il nous suffit de montrer que $QSAT \prec_P SOK$.

On se donne pour cela une *QCNF* $F = Q_n x_n \cdots Q_1 x_1 \ \psi(x_1, \cdots, x_n)$. On note $C_1, \cdots, C_p$ l'ensemble des clauses de ψ. L'objectif est de construire un niveau de sokoban qui admet une solution si et seulement si F s'évalue en 1.

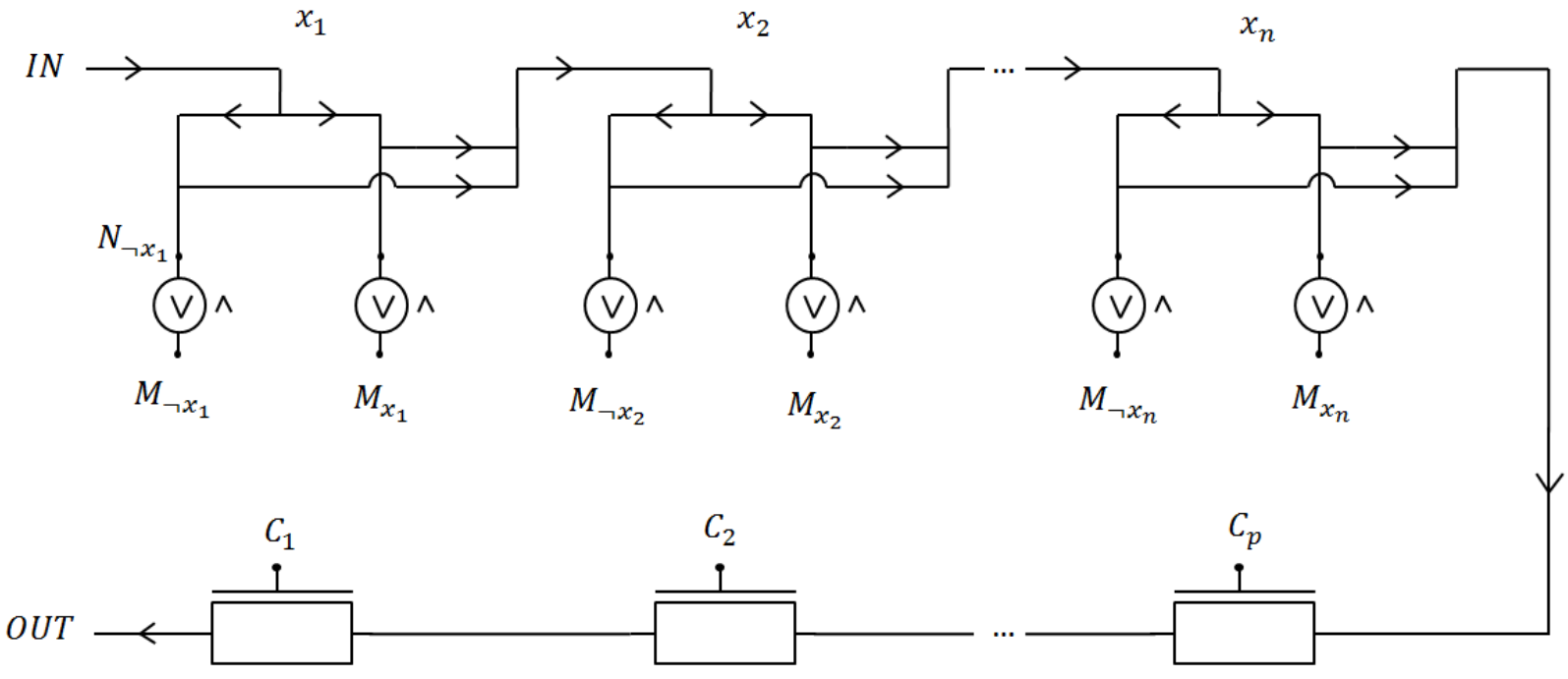
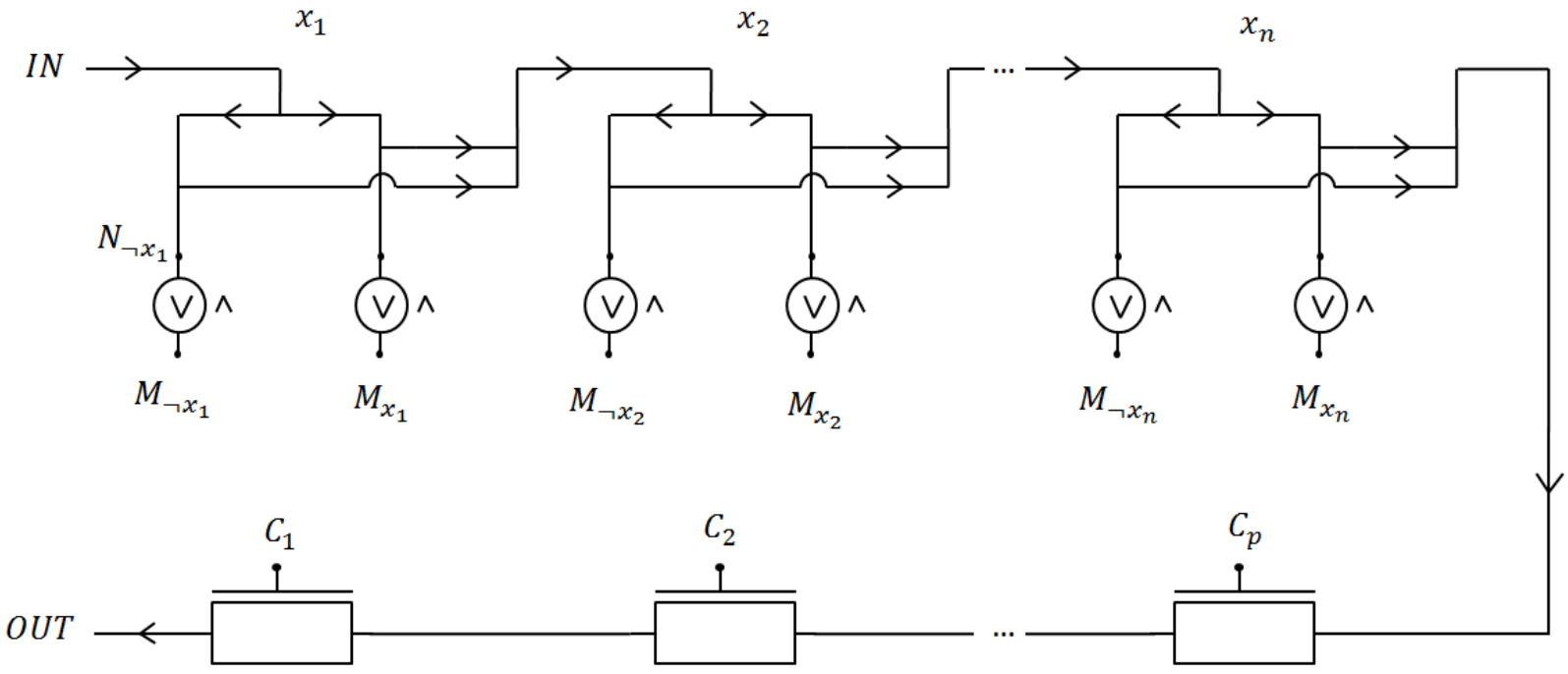


Figure : Le circuit de satisfiabilité G^{SAT}



$$F^k(x_{k+1}, \cdots, x_n) = Q_k x_k \; Q_{k-1} x_{k-1} \; \cdots \; Q_1 x_1 \; \psi(x_1, \cdots, x_n)$$

$$F^k(x_{k+1}, \cdots, x_n) = \begin{cases} F^{k-1}(0, x_{k+1}, \cdots, x_n) \vee F^{k-1}(1, x_{k+1}, \cdots, x_n) & \text{si } Q_k = \exists \\ F^{k-1}(0, x_{k+1}, \cdots, x_n) \wedge F^{k-1}(1, x_{k+1}, \cdots, x_n) & \text{si } Q_k = \forall \end{cases}$$

La classe de complexité NP

Intuitivement, on dit d'un problème qu'il est NP lorsqu'il est possible de vérifier rapidement la validité d'une solution.

Définition

On dit qu'un langage L est NP lorsqu'il existe une machine de Turing polynomiale M, dite *vérificateur*, et un polynôme R tels que :

$L = \{x \mid$ il existe une preuve y telle que $|y| \leqslant R(|x|)$ et M accepte $\langle x, y \rangle\}$

Remarque : NP est également la classe des problèmes solubles en temps polynomial sur une machine de Turing non déterministe.

Fait

$P \subset NP \subset PSPACE$

Question ouverte : a-t-on $P = NP$?

Le Théorème de Cook Levin

Théorème

Le problème SAT de la satisfiabilité des CNF est NP-complet.

Démonstration :

SAT est trivialement dans NP. Montrons que SAT est NP-difficile.
Soit $L \in NP$. On dispose donc de R polynôme et de $M = (Q, \Sigma, \Gamma, q_0, F, \delta)$ vérificateur tels que :

$L = \{x \mid$ il existe une preuve y telle que $|y| \leqslant R(|x|)$ et M accepte $\langle x, y \rangle\}$

On se donne alors P polynôme tel que $\forall n,\ T_M(n + 1 + R(n)) \leqslant P(n)$

Pour une instance x de taille n, on va construire en temps polynomial par rapport à n une expression booléenne B_x telle que $x \in L \iff B_x$ satisfiable.

Pour $q \in Q$, $s \in \Gamma$, $|i| \leqslant P(n)$, et $0 \leqslant k \leqslant P(n)$, on définit :

Variables	**Interprétation**	**Quantité**
$T_{i,s}^{[k]}$	La cellule i contient le symbole s à l'étape k	$O(P(n)^2)$
$H_i^{[k]}$	La tête de lecture est sur la cellule i à l'étape k	$O(P(n)^2)$
$Q_q^{[k]}$	L'état courant est q à l'étape k	$O(P(n))$

Variables	**Interprétation**	**Quantité**
$T_{i,s}^{[k]}$	La cellule i contient le symbole s à l'étape k	$O(P(n)^2)$
$H_i^{[k]}$	La tête de lecture est sur la cellule i à l'étape k	$O(P(n)^2)$
$Q_q^{[k]}$	L'état courant est q à l'étape k	$O(P(n))$

On définit ensuite B_x comme la conjonction des expressions suivantes :

- $Q_{q_0}^{[0]} \wedge H_0^{[0]}$ — 1
- $T_{i,s}^{[0]}$, si la cellule i contient initialement le symbole s — $O(P(n))$
- $T_{i,s}^{[k]} \Rightarrow \neg T_{i,s'}^{[k]}$ pour $s \neq s'$ — $O(P(n)^2)$
- $Q_q^{[k]} \Rightarrow \neg Q_{q'}^{[k]}$ pour $q \neq q'$ — $O(P(n))$
- $H_i^{[k]} \Rightarrow \neg H_{i'}^{[k]}$ pour $i \neq i'$ — $O(P(n)^3)$
- $\left(T_{i,s}^{[k]} \wedge T_{i,s'}^{[k+1]}\right) \Rightarrow H_i^{[k]}$ pour $s \neq s'$ — $O(P(n)^2)$
- $\left(T_{i,s}^{[k]} \wedge Q_q^{[k]} \wedge H_i^{[k]}\right) \Rightarrow \left(T_{i,s'}^{[k+1]} \wedge Q_{q'}^{[k+1]} \wedge H_{i+1}^{[k+1]}\right)$ si $\delta(s,q) = (s', q', \rightarrow)$ — $O(P(n)^2)$
- $Q_{q_+}^{[P(n)]}$ — 1

Le Théorème de Savitch

Dont une conséquence est l'égalité remarquable *NPSPACE = PSPACE*

Théorème (Savitch, 1970)

Soit $s : \mathbb{N} \to \mathbb{R}_+$ telle que $s(n) \geqslant n$ pour n assez grand. Alors toute machine de Turing non déterministe sans calcul infini et qui fonctionne en espace borné par $s(n)$ est équivalente à une machine de Turing déterministe M' en espace $O(s^2(n))$.

Démonstration :

- $M = (Q, \Sigma, \Gamma, q_0, F, \delta)$, machine de Turing non déterministe telle que $S_M(n) \leqslant s(n)$. On impose à M d'effacer la bande avant de s'arrêter.
- On associe un mot sur $Q \cup \Gamma$ à chaque configuration de M. Ainsi, pour une entrée ω, la configuration initiale est $q_0\omega$, et l'unique configuration acceptante est q_+.

$\textsc{Access}(c, c', t) =$
if $t = 0$ **then**
  return $c = c'$ **or** $(c, c') \in E$
else
  for all $c'' \in V$ **do**
    if $\textsc{Access}(c, c'', t-1)$ **and** $\textsc{Access}(c'', c', t-1)$ **then**
      return true
    end if
  end for
  return false
end if

```
ACCESS(c, c', t) =
  if t = 0 then
    return c = c' or (c, c') ∈ E
  else
    for all c'' ∈ V do
      if ACCESS(c, c'', t − 1) and ACCESS(c'', c', t − 1) then
        return true
      end if
    end for
    return false
  end if
```

On a $T_M(n) \leqslant |\Gamma \cup Q|^{s(n)}$, d'où l'existence de K tel que $T_M(n) \leqslant 2^{K \cdot s(n)}$.

Notons m la taille de la plus grande configuration accessible depuis $q_0\omega$. $m \leqslant s(n)$

Alors ω est acceptée si et seulement si $Access(q_0\omega, q_+, m, 2^{K \cdot m})$. Cet appel à *Access* se fait en espace $O(m^2)$.

Il reste à montrer que m est constructible en espace $O(m^2)$. On note pour cela :

- C_k le sous-graphe des configurations de taille au plus k
- N_k le nombre de sommets accessibles dans C_k depuis $q_0\omega$

$(N_k)_{k \geqslant n+1}$ est croissante et majorée par $s(n)$. $m = \min\{k \mid N_k = N_{k+1}\}$.
En outre, on peut évaluer N_k avec *Access* en espace $O(k^2)$.

QSAT est PSPACE-difficile

Soit $L \in PSPACE$, décidé par $M = (Q, \Sigma, \Gamma, q_0, F, \delta)$, et P polynôme bornant la complexité en espace de M. Soit ω entrée de taille s

- On représente une configuration de M par une chaîne de $(Q \cup \Gamma)^*$.
  Par exemple, la configuration initiale est représentée par $q_0\omega$
  Sans perte de généralité, on suppose que l'unique configuration acceptante est q_+
- On peut également représenter une configuration par un ensemble de $|Q \cup \Gamma| \cdot (P(n) + 1)$ variables booléennes $c = \{c_{i,s} \mid 0 \leqslant i \leqslant P(n), s \in Q \cup \Gamma\}$
- On a $T_M(n) \leqslant |\Gamma \cup Q|^{P(n)}$, d'où l'existence de K tel que $T_M(n) \leqslant 2^{K \cdot P(n)}$.

Principe de la démonstration

On va construire par récurrence $\phi_{c,d,i}$, où $\phi_{c,d,i}$ vraie si et seulement si M peut passer de la configuration c à la configuration d en moins de 2^i mouvements.
ω est donc acceptée ssi $\phi_{q_0\omega,\, q_+,\, \lceil \log_2(T_M(n)) \rceil \cdot}$

Deux configurations c et d sont équivalentes si et seulement si :

$$(c \Leftrightarrow d) =^{def} \bigwedge_{i=0}^{P(n)} \bigwedge_{s \in Q \cup \Gamma} (c_{i,s} \Leftrightarrow d_{i,s})$$

Si $\delta(q, a) = (q', b, \leftarrow)$, on pose

$$\phi_{c,d,0}^{(q,a,q',b,\rightarrow)} =$$

$$\bigvee_{j=1}^{P(n)} \left(c_{j,q} \wedge c_{j+1,a} \wedge {}^{d_{j-1,q'}} \wedge d_{j+1,b} \wedge \left(\bigvee_{e \in \Gamma} c_{j-1,e} \wedge d_{j,e} \right) \wedge \left(\bigwedge_{\substack{i \in \{0, \cdots, j-2, j+2, \cdots P(n)\} \\ s \in Q \cup \Gamma}} (c_{i,s} \Leftrightarrow d_{i,s}) \right) \right)$$

On procède de même pour les déplacements à droite. On a alors

$$\phi_{c,d,0} = (c_{i,s} \Leftrightarrow d_{i,s}) \vee \left(\bigvee_{\delta(q,a) = (q',b,\mathcal{D})} \phi_{c,d,0}^{(q,a,q',b,\mathcal{D})} \right)$$

et

$$\phi_{c,d,i+1} = \exists m \, \forall c' \, \forall d' \, ((c \Leftrightarrow c' \wedge m \Leftrightarrow d') \vee (m \Leftrightarrow c' \wedge d \Leftrightarrow d')) \Rightarrow \phi_{c',d',i}$$

La taille des $\phi_{c,d,i}$ est linéaire en i. En outre, on peut mettre ces expressions sous forme de *CNF* en temps polynomial.

Quelques considérations sur les graphes planaires

Ou pourquoi le gadget du pont plan est indispensable à notre construction

Théorème

Soit $G = (S, A)$ un graphe planaire connexe, sans triangles. Alors $|A| \leqslant 2|S| - 4$

Démonstration :
Notons F l'ensemble des faces de G.

- D'après la formule d'Euler : $|S| - |A| + |F| = 2$
- En outre, $\sum_{f \in F} \deg f = 2|A|$.
- Comme G est sans triangles, $\deg f \geqslant 4 \ \forall f \in F$

On a donc $2|A| \geqslant 4|F|$.
Enfin, en réinjectant dans l'équation d'Euler : $|A| \leqslant 2|S| - 4$.

Application :
Pour $p \geqslant 3$, $X_p = [\![1\,;\ p]\!]$, on construit le graphe bipartite $G_p = (S_p, A_p)$ défini par

- $S_p = \{U_i \mid i \in X_p\} \ \cup \ \{V_Y \mid Y \in \mathcal{P}_3(X_p)\}$
- $A_p = \{\{U_i, V_X\} \mid i \in X\}$

G_p, qui est connexe et sans triangles, est contenu dans certaines instances du circuit G^{SAT}. Or, $|S_p| = C_p^3 + p$ et $|A_p| = 3 \cdot C_p^3$. Ainsi, G_p n'est plus planaire pour $p \geqslant 5$.