

Histoire de cruche

PCC - 1

Jean-Marc.Vincent@univ-grenoble-alpes.fr¹

¹Laboratoire LIG
Équipe-Projet INRIA POLARIS
Université Grenoble-Alpes

Algorithmique et Modélisation
L3-INFO

- I. **LE PROBLÈME : Une histoire de cruches**
- II. **UN MODÈLE : un petit automate**
- III. **SYNTÈSE : Un problème classique**



METTRE LA PRESSION

Dans le film d'action Die Hard 3, Bruce Willis et Jeremy Irons doivent résoudre un problème pour désamorcer une bombe



Ils ont 5mn pour résoudre le problème ([version française sans le problème](#)).

Question

Saurez-vous en faire autant, en 5mn (et sans la pression de faire tout exploser) ? (Si vous l'avez déjà fait, obtenez 6l avec des seaux de 5l et 7l)

MODÉLISATION

La difficulté principale de ce problème est sa modélisation, c'est à dire sa représentation sous forme abstraite.

- ▶ ce que l'on a à disposition : 2 seaux de capacité respectives 5l et 3l, un robinet et une évacuation d'eau
- ▶ des opérations : remplir, vider, transvaser du seau de 5 vers seau de 3 ou l'inverse

Problème : trouver la séquence d'opérations partant des 2 seaux vides pour arriver à 4l dans un seau.

MODÉLISATION PAR UN AUTOMATE

Un exemple de modélisation

Pour modéliser le problème par un automate, nous devons spécifier :

- ▶ **État** : l'état de l'automate est un couple de 2 valeurs entières $[s_a, s_b]$ ou s_a modélise le nombre de litres dans le seau A de capacité $C_A = 5l$ donc $0 \leq s_a \leq C_A$ (respectivement s_b pour le seau B , $0 \leq s_b \leq 3 = C_B$)
- ▶ **Transitions** : correspondent aux différentes opérations et leur conditions de déclenchement. À partir de l'état $[s_a, s_b]$ les 6 opérations possibles produisent :
 - ★ $[C_A, s_b]$ pour l'opération remplir le seau A
 - ★ $[s_a, C_B]$ pour l'opération remplir le seau B
 - ★ $[0, s_b]$ pour l'opération vider le seau A
 - ★ $[s_a, 0]$ pour l'opération vider le seau B
 - ★ $[s_a - z, s_b + z]$ pour l'opération transvaser du seau A vers le seau B avec $z = \min(s_a, C_B - s_b)$
 - ★ $[s_a + z, s_b - z]$ pour l'opération transvaser du seau B vers le seau A avec $z = \min(C_A - s_a, s_b)$
- ▶ **État initial** : $[0, 0]$ les seaux sont vides
- ▶ **État finaux** : $[4, \cdot]$ le seau A contient 4l ; il y a 4 états finaux $[4, 0]$, $[4, 1]$, $[4, 2]$ et $[4, 3]$

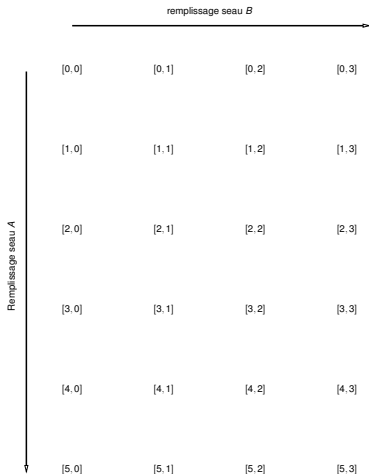
La question se reformule :

Existe-t'il une séquence de transitions conduisant de l'état initial à l'un des états finaux ?

si oui donner une séquence des transitions pour aller de l'état initial à un état final, en bonus on pourrait chercher une séquence la plus efficace (la plus courte, ou celle qui consomme le moins d'eau, ...)

GRAPHE ASSOCIÉ À L'AUTOMATE

Une représentation de l'automate



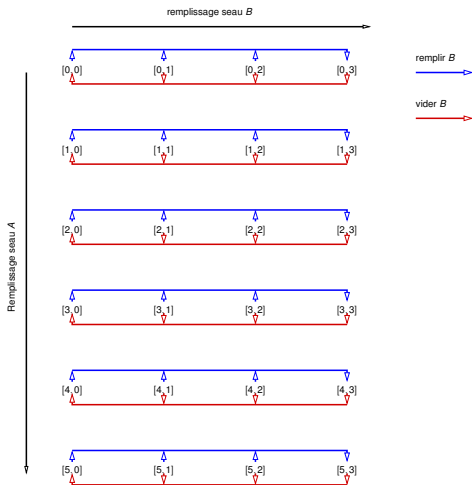
Remarques

- ▶ ensemble des états visualisés sur une grille (cf votre démarche)
- ▶ traitement systématique de toutes les transitions

GRAPHE ASSOCIÉ À L'AUTOMATE

Une représentation de l'automate

Remarques

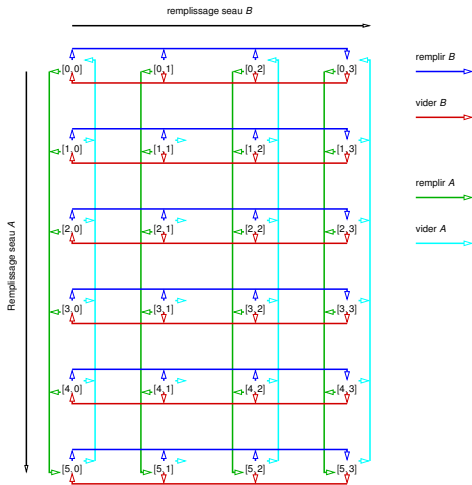


- ▶ ensemble des états visualisés sur une grille (cf votre démarche)
- ▶ traitement systématique de toutes les transitions

GRAPHE ASSOCIÉ À L'AUTOMATE

Une représentation de l'automate

Remarques

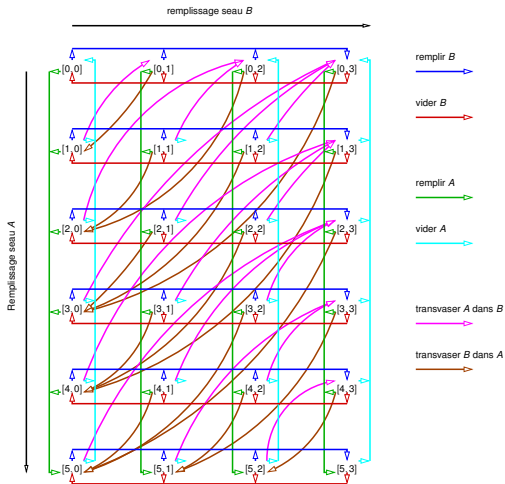


- ▶ ensemble des états visualisés sur une grille (cf votre démarche)
- ▶ traitement systématique de toutes les transitions

GRAPHE ASSOCIÉ À L'AUTOMATE

Une représentation de l'automate

Remarques

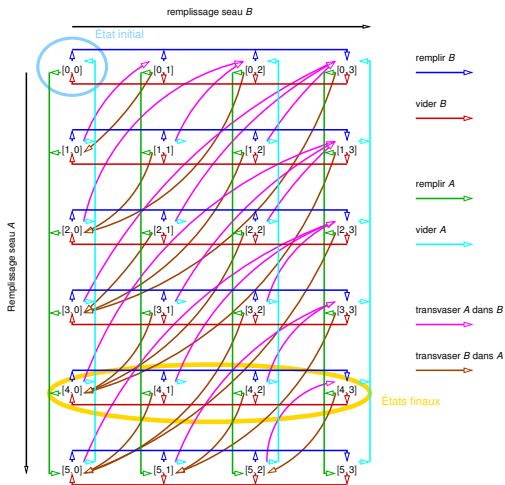


- ensemble des états visualisés sur une grille (cf votre démarche)
- traitement systématique de toutes les transitions

GRAPHE ASSOCIÉ À L'AUTOMATE

Une représentation de l'automate

Remarques

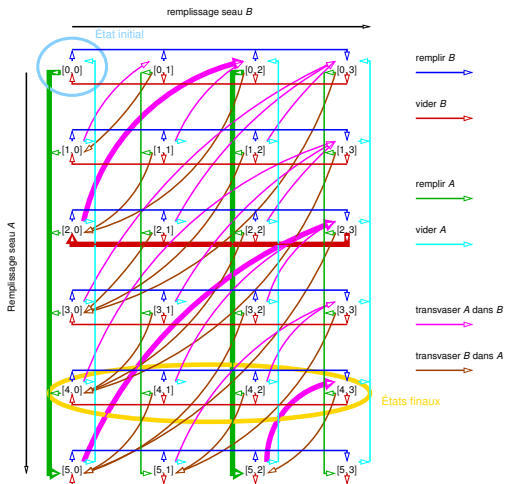


- ▶ ensemble des états visualisés sur une grille (cf votre démarche)
- ▶ traitement systématique de toutes les transitions

GRAPHE ASSOCIÉ À L'AUTOMATE

Une représentation de l'automate

Remarques



- ▶ ensemble des états visualisés sur une grille (cf votre démarche)
- ▶ traitement systématique de toutes les transitions
- ▶ Par tâtonnement on trouve une première solution

$$[0, 0] \xrightarrow{r_A} [5, 0] \xrightarrow{t_{A,B}} [2, 3]$$

$$\xrightarrow{v_B} [2, 0] \xrightarrow{t_{A,B}} [0, 2]$$

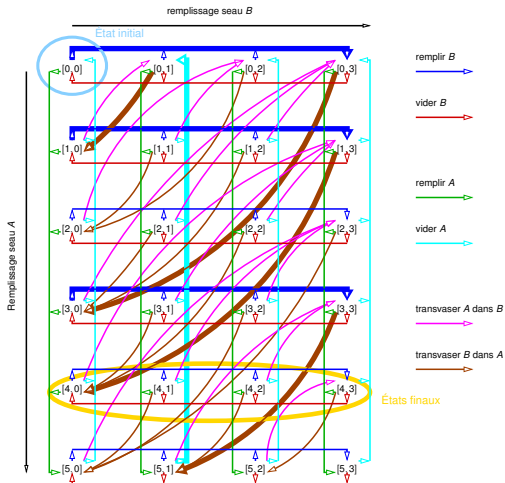
$$\xrightarrow{r_A} [5, 2] \xrightarrow{t_{A,B}} [4, 3]$$

6 opérations effectuées

GRAPHE ASSOCIÉ À L'AUTOMATE

Une représentation de l'automate

Remarques



- ▶ ensemble des états visualisés sur une grille (cf votre démarche)
- ▶ traitement systématique de toutes les transitions
- ▶ Par tâtonnement on trouve une première solution

$$\begin{aligned}
 [0, 0] &\xrightarrow{r_A} [5, 0] \xrightarrow{t_{A,B}} [2, 3] \\
 &\xrightarrow{v_B} [2, 0] \xrightarrow{t_{A,B}} [0, 2] \\
 &\xrightarrow{r_A} [5, 2] \xrightarrow{t_{A,B}} [4, 3]
 \end{aligned}$$

6 opérations effectuées

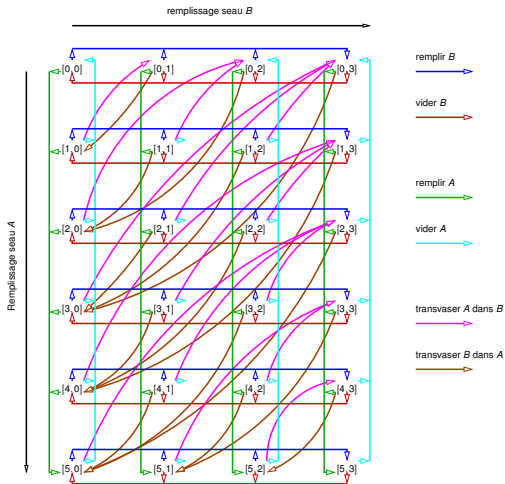
- ▶ Une autre solution

$$\begin{aligned}
 [0, 0] &\xrightarrow{r_B} [0, 3] \xrightarrow{t_{B,A}} [3, 0] \\
 &\xrightarrow{r_B} [3, 3] \xrightarrow{t_{B,A}} [5, 1] \\
 &\xrightarrow{v_A} [0, 1] \xrightarrow{t_{B,A}} [1, 0] \\
 &\xrightarrow{r_B} [1, 3] \xrightarrow{t_{B,A}} [4, 0]
 \end{aligned}$$

8 opérations effectuées

GRAPHE ASSOCIÉ À L'AUTOMATE

Une représentation de l'automate



curieux : certains états ne sont pas atteignables à partir de [0, 0] :

[1, 1], [1, 2], [2, 1], [2, 2], [3, 1], [3, 2], [4, 1 et 4, 2]

Remarques

- ▶ ensemble des états visualisés sur une grille (cf votre démarche)
- ▶ traitement systématique de toutes les transitions
- ▶ Par tâtonnement on trouve une première solution

$$[0, 0] \xrightarrow{r_A} [5, 0] \xrightarrow{t_{A,B}} [2, 3]$$

$$\xrightarrow{v_B} [2, 0] \xrightarrow{t_{A,B}} [0, 2]$$

$$\xrightarrow{r_A} [5, 2] \xrightarrow{t_{A,B}} [4, 3]$$

6 opérations effectuées

- ▶ Une autre solution

$$[0, 0] \xrightarrow{r_B} [0, 3] \xrightarrow{t_{B,A}} [3, 0]$$

$$\xrightarrow{r_B} [3, 3] \xrightarrow{t_{B,A}} [5, 1]$$

$$\xrightarrow{v_A} [0, 1] \xrightarrow{t_{B,A}} [1, 0]$$

$$\xrightarrow{r_B} [1, 3] \xrightarrow{t_{B,A}} [4, 0]$$

8 opérations effectuées

APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).

$[0, 0]$

remplir B



vider B



remplir A



vider A



transvaser A dans B

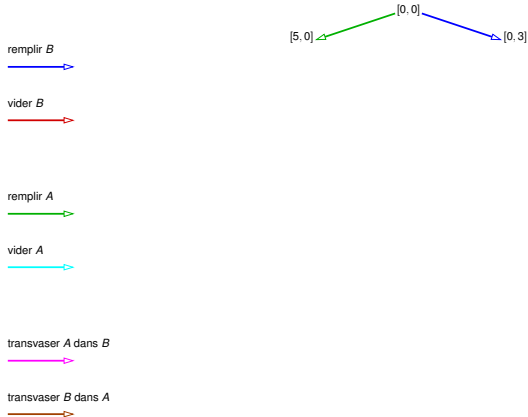


transvaser B dans A



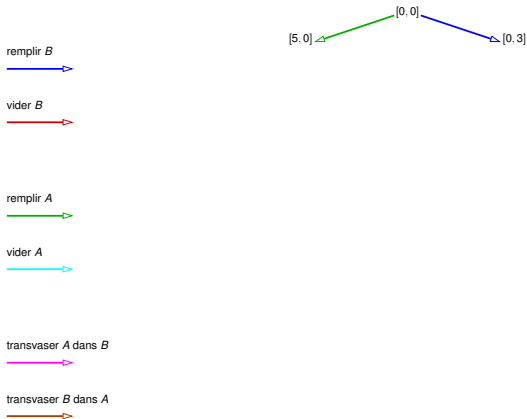
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



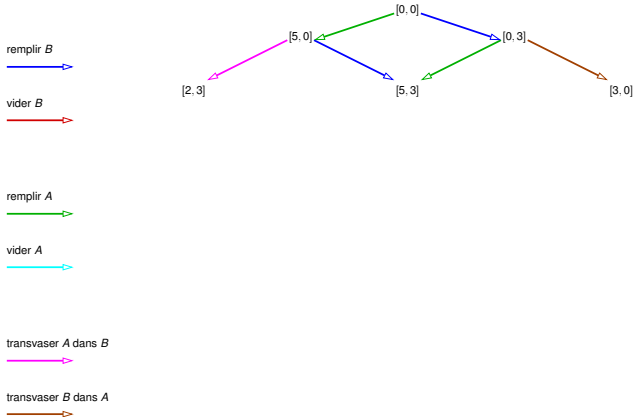
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



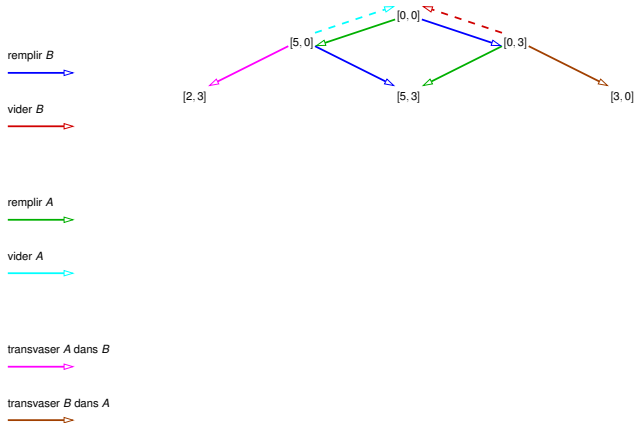
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



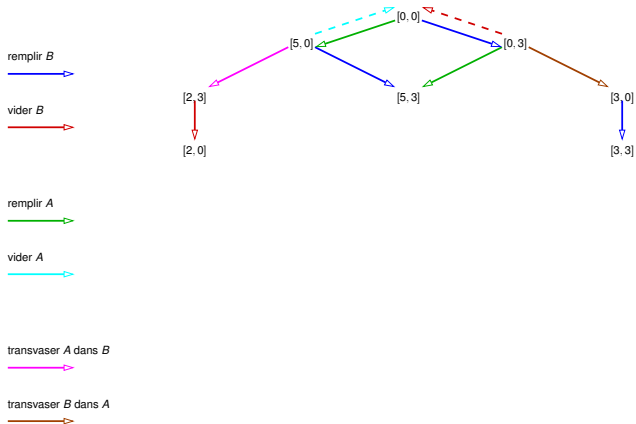
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



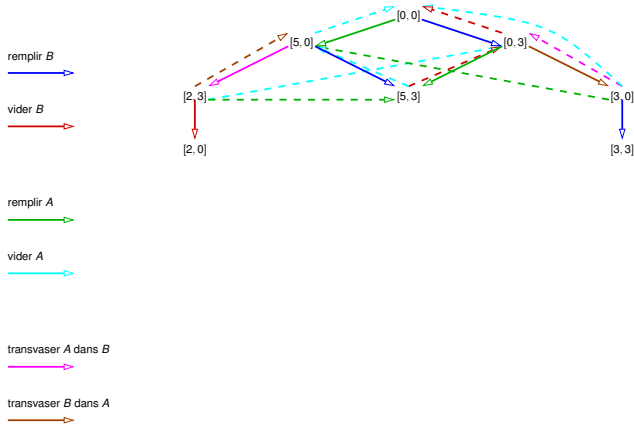
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



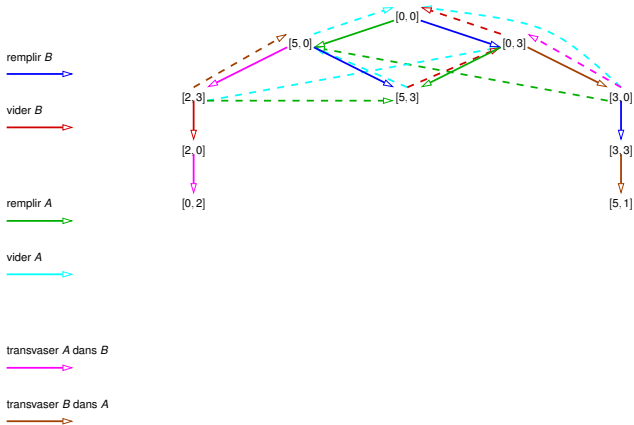
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



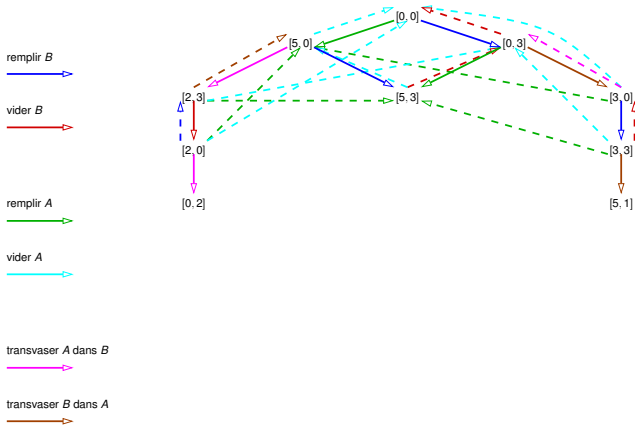
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



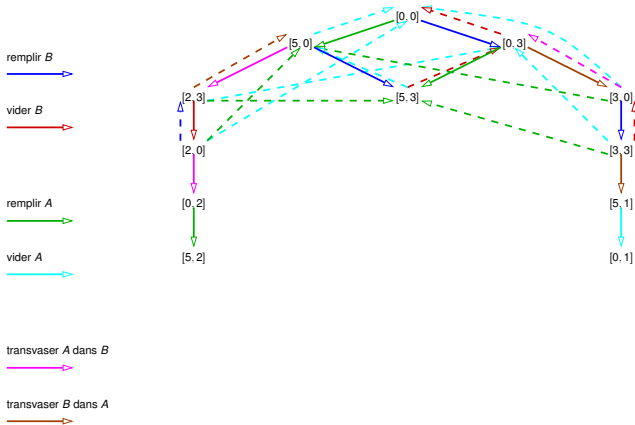
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



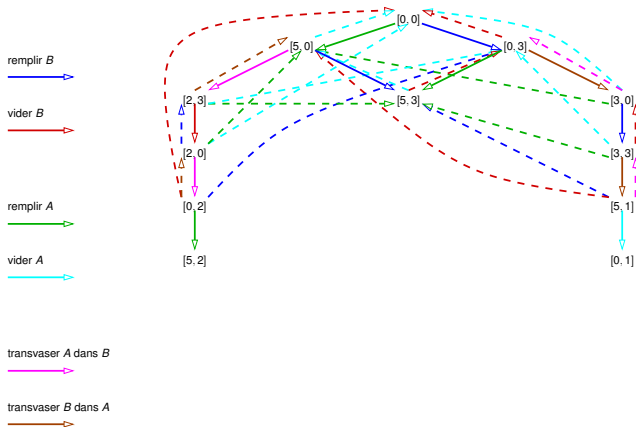
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



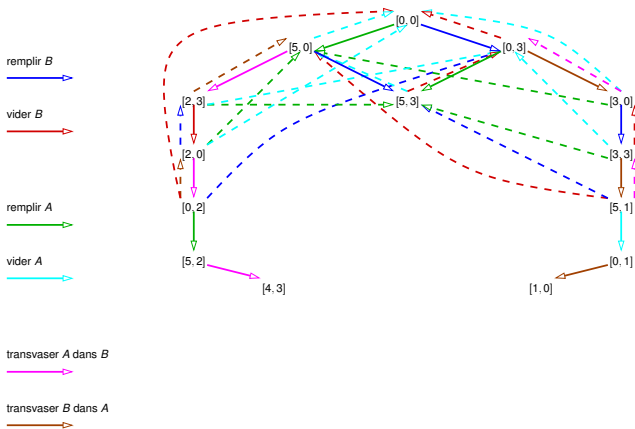
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



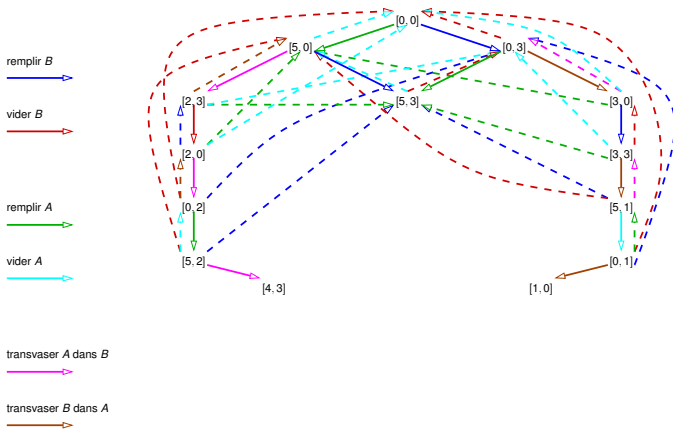
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



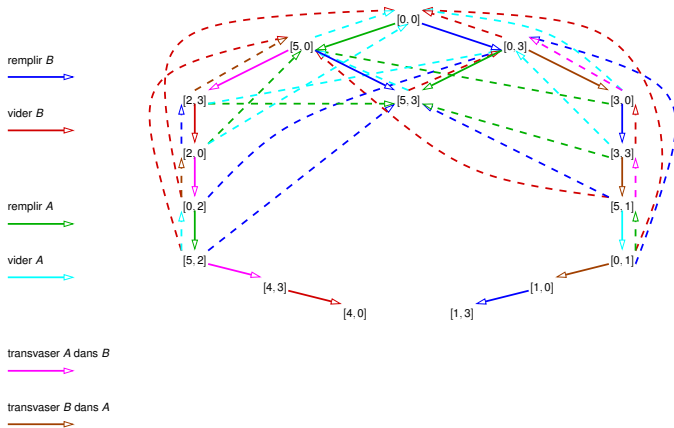
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



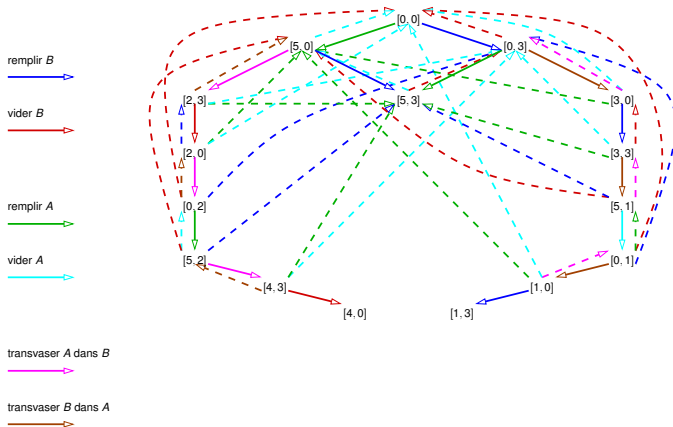
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



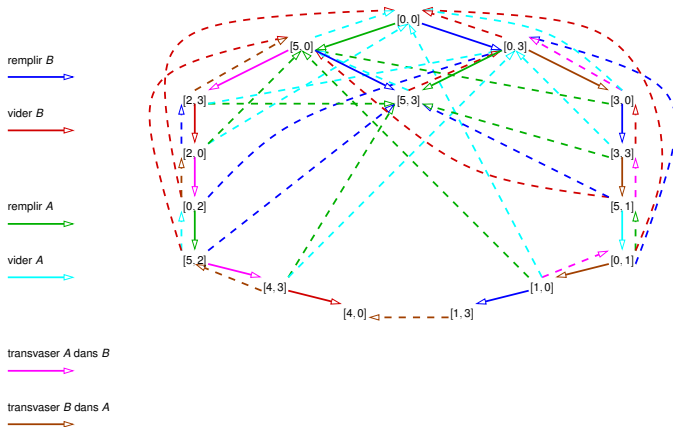
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



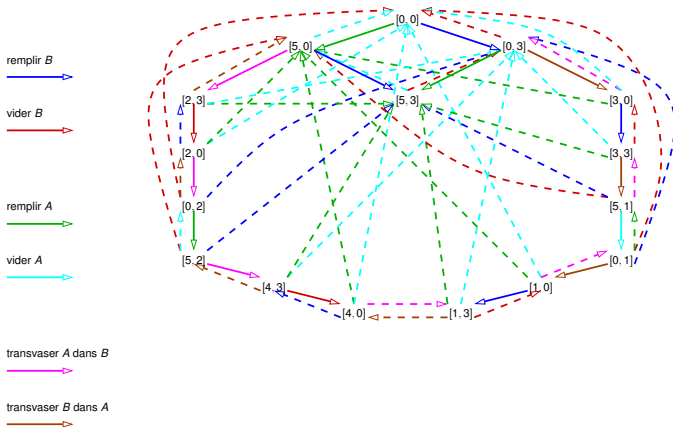
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



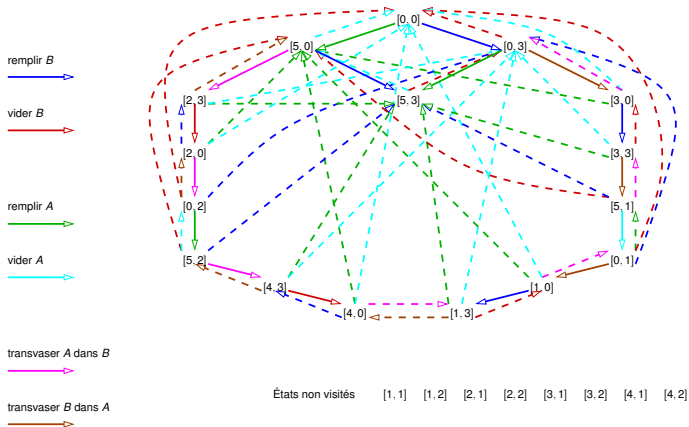
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



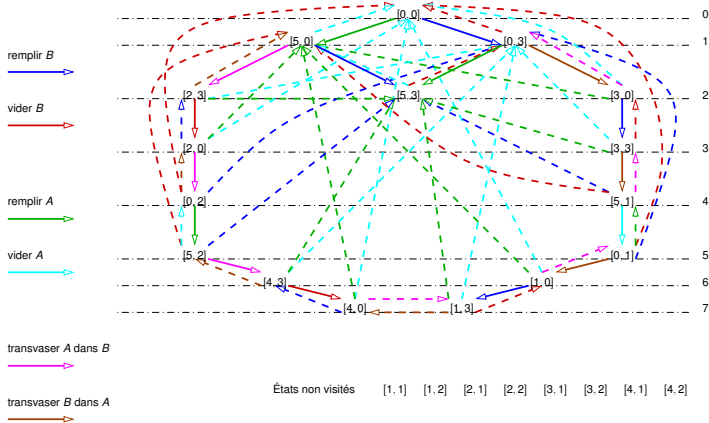
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



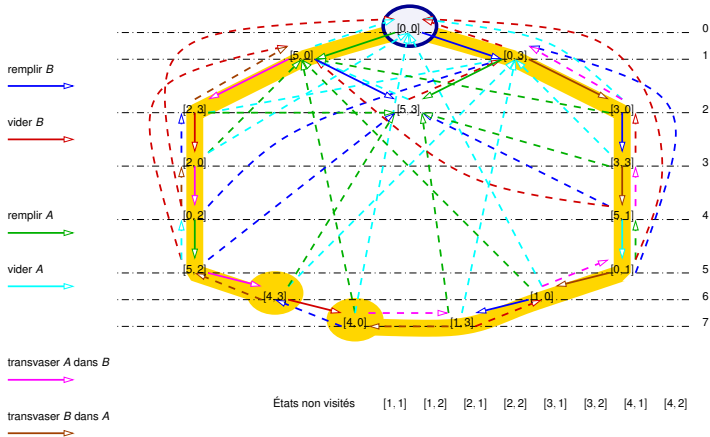
APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



APPROCHE CONSTRUCTIVE

L'idée est de partir de l'état $[0, 0]$ et d'explorer de proche en proche. On examine les voisins au fur et à mesure qu'ils sont rencontrés (et de manière exhaustive).



SYNTHÈSE

Modélisation

- ▶ **Automate fini déterministe** pour décrire la dynamique du système
 - ⇒ Ensemble fini d'états, alphabet fini, fonction de transition, état initial, états finaux
 - ⇒ graphe associé \mathcal{G} : sommets du graphe (états) \mathcal{X} , fonction de transition (arcs) \mathcal{A} , étiquettes des arcs (lettres) \mathcal{V}
- ▶ Reformulation du problème dans le **formalisme des graphes** :
Étant donné un sommet $x \in \mathcal{G}$ et un ensemble de sommets $\{y_1, \dots, y_k\} \subset \mathcal{X}$
 - i. **Existe**-t'il un chemin de x vers l'un des y_j ?
 - ii. Lorsqu'un tel chemin existe, donner la **séquence** des arcs empruntés (suite des transitions)
 - iii. Parmi tous les chemins donner un chemin de **coût minimal** (suppose que l'on se donne une fonction de coût)

SYNTÈSE

Modélisation

- ▶ **Automate fini déterministe** pour décrire la dynamique du système
 - ⇒ Ensemble fini d'états, alphabet fini, fonction de transition, état initial, états finaux
 - ⇒ graphe associé \mathcal{G} : sommets du graphe (états) \mathcal{X} , fonction de transition (arcs) \mathcal{A} , étiquettes des arcs (lettres) \mathcal{V}
- ▶ Reformulation du problème dans le **formalisme des graphes** :
 Étant donné un sommet $x \in \mathcal{G}$ et un ensemble de sommets $\{y_1, \dots, y_k\} \subset \mathcal{X}$
 - i. **Existe**-t'il un chemin de x vers l'un des y_j ?
 - ii. Lorsqu'un tel chemin existe, donner la **séquence** des arcs empruntés (suite des transitions)
 - iii. Parmi tous les chemins donner un chemin de **coût minimal** (suppose que l'on se donne une fonction de coût)

Classes de problèmes sur les chemins dans les graphes

Étant donné un graphe $\mathcal{G} = (\mathcal{X}, \mathcal{A})$ éventuellement arcs (ou sommets) valués

- i. **Existence de chemins** (réponse : oui/non)
 - entre x et y sommets donnés du graphe
 - entre un sommet x et tous les autres sommets
 - pour tout couple de sommets du graphe
- ii. **Construction de chemins** (réponse : séquence d'arcs/sommets)
- iii. **Construction de chemins optimaux** (réponse : séquence d'arcs/sommets/valeurs)
chemins les plus courts, de poids minimum,...

UN MODÈLE ARITHMÉTIQUE

Le problème repose sur des additions et soustractions d'entiers, il est donc naturel de regarder les propriétés arithmétiques des couples d'entiers obtenus par les transformations vider, remplir et transvaser.

Propriétés de (x, y) partant de l'état initial $(0, 0)$ pour arriver à c

- ▶ Invariant : x et y sont des combinaisons linéaires entières de C_A et C_B
- ▶ x et y sont des multiples de $\text{pgcd}(C_A, C_B)$
- ▶ Il existe (u', v') tels que $u' C_A + v' C_B = c$ (penser au théorème de Bezout)

En déduire un algorithme glouton pour résoudre le problème.

Merci à Denis Trystram