Solveurs SAT et application au *model checking*

Loïg Jezequel

Université de Nantes

IRCCyN, UMR CNRS 6597

29 mars 2016

Aperçu de cet exposé

Point de départ : la planification

Plans (chemins dans des graphes) et solveurs SAT

$\hookrightarrow$ souvent très efficace en pratique

Et si...

- ...on veut plus qu'un plan ?
- ...on veut tous les plans / prouver l'absence de plan ?
- ...on veut préserver la concurrence ?

Aperçu de cet exposé

Point de départ : la planification

Plans (chemins dans des graphes) et solveurs SAT

$\hookrightarrow$ souvent très efficace en pratique

Et si...

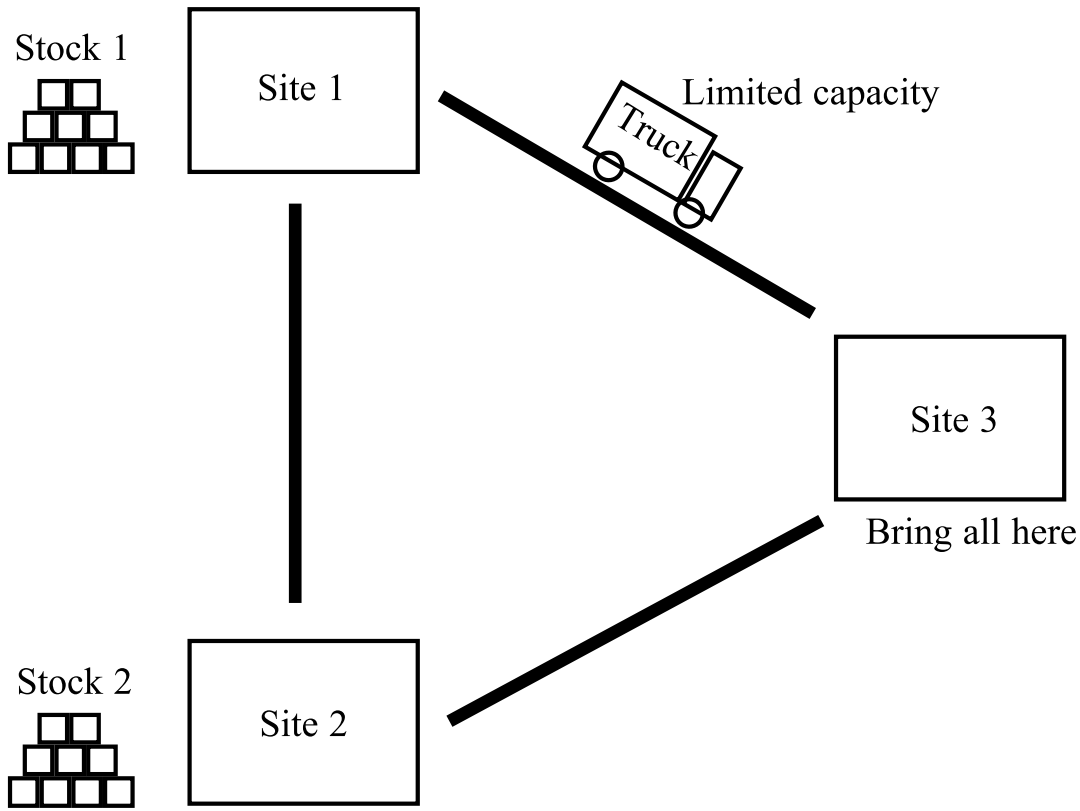
- ...on veut plus qu'un plan ?
- ...on veut tous les plans / prouver l'absence de plan ?
- ...on veut préserver la concurrence ?

Une idée

Dépliages (de graphes, de réseaux de Petri) et solveurs SAT

Planification et solveurs SAT

La planification



Objectif

Trouver un *plan* : une séquence d'actions faisant passer le système de son état initial à un de ses états cibles.

STRIPS : représentation des problèmes de planification

Syntaxe

(A, O, I, G) avec :

- A un ensemble d'atomes
- $\ell(A) = \{a, \neg a \ : \ a \in A\}$ l'ensemble de littéraux correspondant
- $O \subseteq 2^{\ell(A)} \times 2^{\ell(A)}$ un ensemble d'opérateurs
- $I \subseteq A$ la représentation d'un état initial
- $G \subseteq \ell(A)$ la représentation d'un ensemble d'états cibles

Sémantique

État : sous-ensemble de $\ell(A)$ contenant exactement un représentant de chaque élément de A

Transition : de s à s' par $o = (p, e) \in O$ si et seulement si $s \supseteq p$, $s' \supseteq e$ et $s(a) = s'(a)$ dès que $e(a)$ non défini

SAT

Le problème de base

- V ensemble de variables
- $\ell(V)$ l'ensemble de littéraux correspondant
- $\vee$ ou logique (disjonction)
- $\wedge$ et logique (conjonction)

Question

Étant donnée une formule, existe-t-il une valuation des variables qui la satisfasse ? (En pratique on donnera une telle valuation.)

SAT

Le problème de base

- V ensemble de variables
- $\ell(V)$ l'ensemble de littéraux correspondant
- $\vee$ ou logique (disjonction)
- $\wedge$ et logique (conjonction)

Question

Étant donnée une formule, existe-t-il une valuation des variables qui la satisfasse ? (En pratique on donnera une telle valuation.)

Forme normale conjonctive

Clause : disjonction de littéraux

Formule CNF : conjonction de clauses

La planification en tant qu'instance de SAT

Idée de départ [Kautz et Selman, ECAI 92]

STRIPS et SAT sont très proches, une transformation syntaxique de STRIPS en SAT peut être effectuée

Mais il y a un problème...

STRIPS est « dynamique », SAT est « statique »

La planification en tant qu'instance de SAT

Idée de départ [Kautz et Selman, ECAI 92]

STRIPS et SAT sont très proches, une transformation syntaxique de STRIPS en SAT peut être effectuée

Mais il y a un problème...

STRIPS est « dynamique », SAT est « statique »

Pour s'en sortir

Introduction de nouvelles variables pour coder la dynamique : un littéral de STRIPS sera instancié plusieurs fois en SAT

$$\ell@t_0, \ell@t_1, \ell@t_2, \ell@t_3, \ldots$$

De STRIPS à SAT, la base

Choix d'un horizon temporel k

Intuition : on cherche un plan qui soit une séquence de k actions

Instanciation des variables

Pour chaque atome $a \in A$ et chaque opérateur $o \in O$, à chaque point de temps $t_i \in \{t_0, \ldots, t_k\}$ on crée une variable : $a@t_i$ ou $o@t_i$

Définition des clauses

Les clauses servent à représenter :

- l'état initial (à partir des $a@t_0$)
- les états cibles (à partir des $a@t_k$)
- les contraintes pour utiliser un opérateur à chaque point de temps t_i (à partir des $a@t_i$, $a@t_{i+1}$ et $o@t_i$)

De STRIPS à SAT, les clauses – état initial, états cibles

État initial

$$\left(\bigwedge_{a \in I} a@t_0\right) \wedge \left(\bigwedge_{a \in A \setminus I} \neg a@t_0\right)$$

De STRIPS à SAT, les clauses – état initial, états cibles

État initial

$$\left(\bigwedge_{a \in I} a@t_0\right) \wedge \left(\bigwedge_{a \in A \setminus I} \neg a@t_0\right)$$

États cibles

$$\left(\bigwedge_{\ell \in G} \ell@t_k\right)$$

remarque : $(\neg a)@t_k$ abus de notation pour $\neg(a@t_k)$

De STRIPS à SAT, les clauses – opérateurs

Opérateur $o = (p, e)$ tirable au temps t_i

$$o@t_i \implies \bigwedge_{\ell \in p} \ell@t_i$$

De STRIPS à SAT, les clauses – opérateurs

Opérateur $o = (p, e)$ tirable au temps t_i

$$o@t_i \implies \bigwedge_{\ell \in p} \ell@t_i$$

Effets d'un tire de $o = (p, e)$ au temps t_i

$$o@t_i \implies \bigwedge_{\ell \in e} \ell@t_{i+1}$$

De STRIPS à SAT, les clauses – opérateurs

Opérateur $o = (p, e)$ tirable au temps t_i

$$o@t_i \implies \bigwedge_{\ell \in p} \ell@t_i$$

Effets d'un tire de $o = (p, e)$ au temps t_i

$$o@t_i \implies \bigwedge_{\ell \in e} \ell@t_{i+1}$$

Un seul opérateur o au temps t_i

$$o@t_i \implies \bigwedge_{o' \neq o \in O} \neg o'@t_i$$

De STRIPS à SAT, les clauses – les atomes

Explication de $\ell @ t_i$

$$\ell @ t_i \implies \ell @ t_{i-1} \vee \bigvee_{\substack{o \in O \\ o=(p,e) \\ \ell \in e}} o @ t_{i-1}$$

De STRIPS à SAT, exemple

STRIPS

$$A = \{a, b, c\}$$

$$O = \{ \quad o_1 = (\{a\}, \{\neg b\}) \quad o_2 = (\{a, b\}, \{c\}) \quad \}$$

$$I = \{a, b\}$$

$$G = \{\neg b, c\}$$

SAT (horizon $k = 1$)

De STRIPS à SAT, exemple

STRIPS

$$A = \{a, b, c\}$$

$$O = \{ \quad o_1 = (\{a\}, \{\neg b\}) \quad o_2 = (\{a, b\}, \{c\}) \quad \}$$

$$I = \{a, b\}$$

$$G = \{\neg b, c\}$$

SAT (horizon $k = 1$)

$$a@t_0$$
$$b@t_0$$
$$\neg(c@t_0)$$

De STRIPS à SAT, exemple

STRIPS

$$A = \{a, b, c\}$$

$$O = \{ \quad o_1 = (\{a\}, \{\neg b\}) \quad o_2 = (\{a, b\}, \{c\}) \quad \}$$

$$I = \{a, b\}$$

$$G = \{\neg b, c\}$$

SAT (horizon $k = 1$)

$$a@t_0$$
$$b@t_0$$
$$\neg(c@t_0)$$

$$\neg(b@t_2)$$
$$c@t_2$$

De STRIPS à SAT, exemple

STRIPS

$$A = \{a, b, c\}$$

$$O = \{ \quad o_1 = (\{a\}, \{\neg b\}) \quad o_2 = (\{a, b\}, \{c\}) \quad \}$$

$$I = \{a, b\}$$

$$G = \{\neg b, c\}$$

SAT (horizon $k = 1$)

$$a@t_0$$
$$b@t_0$$
$$\neg(c@t_0)$$
$$o_1@t_0 \implies a@t_0$$
$$o_1@t_0 \implies \neg(b@t_1)$$

$$\neg(b@t_2)$$
$$c@t_2$$

De STRIPS à SAT, exemple

STRIPS

$$A = \{a, b, c\}$$

$$O = \{ \quad o_1 = (\{a\}, \{\neg b\}) \quad o_2 = (\{a, b\}, \{c\}) \quad \}$$

$$I = \{a, b\}$$

$$G = \{\neg b, c\}$$

SAT (horizon $k = 1$)

$$a@t_0$$
$$b@t_0$$
$$\neg(c@t_0)$$
$$o_1@t_0 \implies a@t_0$$
$$o_1@t_0 \implies \neg(b@t_1)$$
$$o_2@t_0 \implies a@t_0 \quad \wedge \quad b@t_0$$
$$o_2@t_0 \implies c@t_1$$

$$\neg(b@t_2)$$
$$c@t_2$$

De STRIPS à SAT, exemple

STRIPS

$$A = \{a, b, c\}$$

$$O = \{ \quad o_1 = (\{a\}, \{\neg b\}) \quad o_2 = (\{a, b\}, \{c\}) \quad \}$$

$$I = \{a, b\}$$

$$G = \{\neg b, c\}$$

SAT (horizon $k = 1$)

$$a@t_0$$
$$b@t_0$$
$$\neg(c@t_0)$$
$$o_1@t_0 \implies a@t_0$$
$$o_1@t_0 \implies \neg(b@t_1)$$
$$o_2@t_0 \implies a@t_0 \quad \wedge \quad b@t_0$$
$$o_2@t_0 \implies c@t_1$$
$$o_1@t_0 \implies \neg(o_2@t_0)$$

$$\neg(b@t_2)$$
$$c@t_2$$

De STRIPS à SAT, exemple

STRIPS

$$A = \{a, b, c\}$$

$$O = \{ \quad o_1 = (\{a\}, \{\neg b\}) \quad o_2 = (\{a, b\}, \{c\}) \quad \}$$

$$I = \{a, b\}$$

$$G = \{\neg b, c\}$$

SAT (horizon $k = 1$)

$$a@t_0$$
$$b@t_0$$
$$\neg(c@t_0)$$
$$o_1@t_0 \implies a@t_0$$
$$o_1@t_0 \implies \neg(b@t_1)$$
$$o_2@t_0 \implies a@t_0 \quad \wedge \quad b@t_0$$
$$o_2@t_0 \implies c@t_1$$
$$o_1@t_0 \implies \neg(o_2@t_0)$$
$$a@t_1 \implies a@t_0$$
$$\neg(a@t_1) \implies \neg(a@t_0)$$

$$\neg(b@t_2)$$
$$c@t_2$$

De STRIPS à SAT, exemple

STRIPS

$$A = \{a, b, c\}$$

$$O = \{ \quad o_1 = (\{a\}, \{\neg b\}) \quad o_2 = (\{a, b\}, \{c\}) \quad \}$$

$$I = \{a, b\}$$

$$G = \{\neg b, c\}$$

SAT (horizon $k = 1$)

$$a@t_0$$
$$b@t_0$$
$$\neg(c@t_0)$$
$$o_1@t_0 \implies a@t_0$$
$$o_1@t_0 \implies \neg(b@t_1)$$
$$o_2@t_0 \implies a@t_0 \wedge b@t_0$$
$$o_2@t_0 \implies c@t_1$$
$$o_1@t_0 \implies \neg(o_2@t_0)$$
$$a@t_1 \implies a@t_0$$
$$\neg(a@t_1) \implies \neg(a@t_0)$$
$$b@t_1 \implies b@t_0$$
$$\neg(b@t_1) \implies \neg(b@t_0) \vee o_1@t_0$$

$$\neg(b@t_2)$$
$$c@t_2$$

De STRIPS à SAT, exemple

STRIPS

$$A = \{a, b, c\}$$

$$O = \{ \quad o_1 = (\{a\}, \{\neg b\}) \quad o_2 = (\{a, b\}, \{c\}) \quad \}$$

$$I = \{a, b\}$$

$$G = \{\neg b, c\}$$

SAT (horizon $k = 1$)

$$
\begin{array}{lcll}
a@t_0 & & & \\
b@t_0 & & & \\
\neg(c@t_0) & & & \\
o_1@t_0 & \Longrightarrow & a@t_0 & \\
o_1@t_0 & \Longrightarrow & \neg(b@t_1) & \\
o_2@t_0 & \Longrightarrow & a@t_0 & \wedge \quad b@t_0 \\
o_2@t_0 & \Longrightarrow & c@t_1 & \\
o_1@t_0 & \Longrightarrow & \neg(o_2@t_0) & \\
a@t_1 & \Longrightarrow & a@t_0 & \\
\neg(a@t_1) & \Longrightarrow & \neg(a@t_0) & \\
b@t_1 & \Longrightarrow & b@t_0 & \\
\neg(b@t_1) & \Longrightarrow & \neg(b@t_0) & \vee \quad o_1@t_0 \\
c@t_1 & \Longrightarrow & c@t_0 & \vee \quad o_2@t_0 \\
\neg(c@t_1) & \Longrightarrow & \neg(c@t_1) & \\
\neg(b@t_2) & & & \\
c@t_2 & & &
\end{array}
$$

De STRIPS à SAT, exemple

STRIPS

$$A = \{a, b, c\}$$

$$O = \{ \quad o_1 = (\{a\}, \{\neg b\}) \quad o_2 = (\{a, b\}, \{c\}) \quad \}$$

$$I = \{a, b\}$$

$$G = \{\neg b, c\}$$

Solution ?

Horizon $k = 1$: non

Horizon $k = 2$:

$a@t_0, b@t_0, o_2@t_0,$
$a@t_1, b@t_1, c@t_1, o_1@t_1,$
$a@t_2, c@t_2$

SAT (horizon $k = 1$)

$$
\begin{array}{lcll}
a@t_0 & & & \\
b@t_0 & & & \\
\neg(c@t_0) & & & \\
o_1@t_0 & \Longrightarrow & a@t_0 & \\
o_1@t_0 & \Longrightarrow & \neg(b@t_1) & \\
o_2@t_0 & \Longrightarrow & a@t_0 \quad \wedge & b@t_0 \\
o_2@t_0 & \Longrightarrow & c@t_1 & \\
o_1@t_0 & \Longrightarrow & \neg(o_2@t_0) & \\
a@t_1 & \Longrightarrow & a@t_0 & \\
\neg(a@t_1) & \Longrightarrow & \neg(a@t_0) & \\
b@t_1 & \Longrightarrow & b@t_0 & \\
\neg(b@t_1) & \Longrightarrow & \neg(b@t_0) \quad \vee & o_1@t_0 \\
c@t_1 & \Longrightarrow & c@t_0 \quad \vee & o_2@t_0 \\
\neg(c@t_1) & \Longrightarrow & \neg(c@t_1) & \\
\neg(b@t_2) & & & \\
c@t_2 & & &
\end{array}
$$

En pratique

Algorithme de planification simple à partir d'un solveur SAT

1. Partir de l'horizon $k = 0$
2. Faire le codage SAT
3. Appeler le solveur SAT
4. Si une solution est trouvée, la transcrire en un plan
5. Sinon, augmenter k et recommencer à l'étape 2

En pratique

Algorithme de planification simple à partir d'un solveur SAT

1. Partir de l'horizon $k = 0$
2. Faire le codage SAT
3. Appeler le solveur SAT
4. Si une solution est trouvée, la transcrire en un plan
5. Sinon, augmenter k et recommencer à l'étape 2

Aller plus loin (voir notamment les travaux de J. Rintanen)

- Prendre en compte l'exclusion mutuelle entre littéraux
- Améliorer la transformation :
  plans séquentiels $\rightarrow$ plans $\forall$-*step* $\rightarrow$ plans $\exists$-*step*
- Paralléliser la recherche aux différents horizons
- Spécialiser les solveurs

Dépliages

Dépliages : définition informelle

C'est quoi un dépliage ?

Une représentation sans cycles des comportements d'un système

Dépliage et concurrence

- Graphe $\rightarrow$ arbre ou DAG
- Réseau de Petri $\rightarrow$ arbre, DAG ou réseau d'occurrences

Préfixes finis complets

En général, un dépliage est infini, ce qui n'est pas très pratique
$\hookrightarrow$ on lui préfère un de ses préfixes, suffisamment informatif

Exemple : qui contienne un représentant de chaque sommet atteignable d'un graphe

Dépliages de problèmes STRIPS sous forme d'arbres

STRIPS

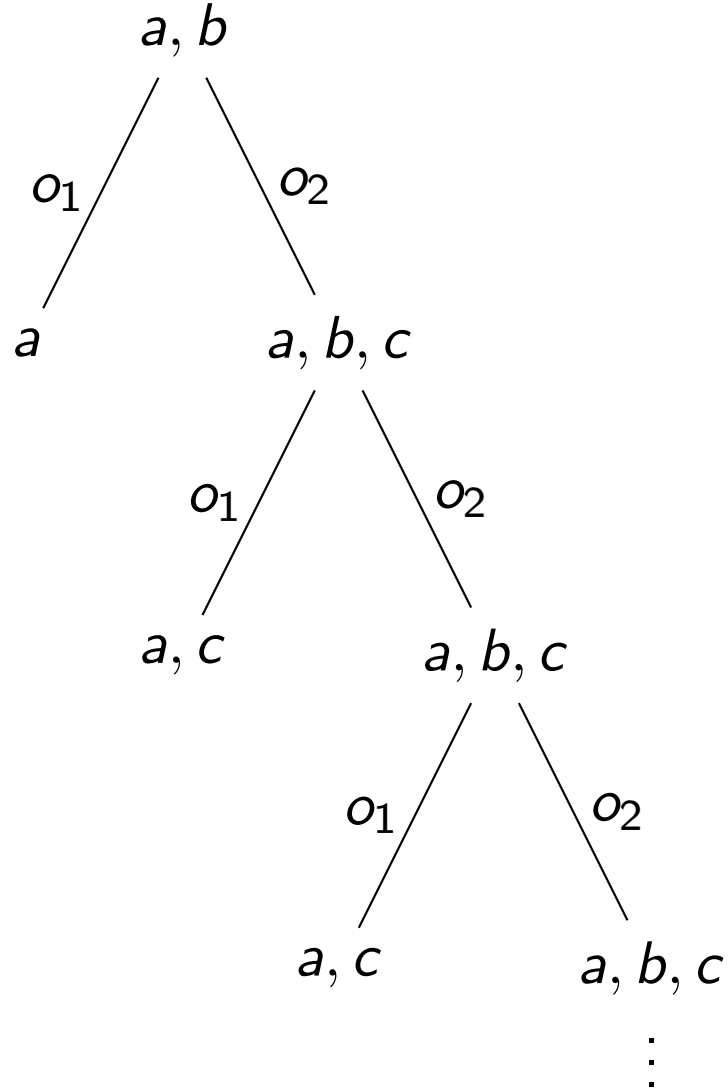
$$A = \{a, b, c\}$$

$$O = \{ \quad o_1 = (\{a\}, \{\neg b\}) \quad o_2 = (\{a, b\}, \{c\}) \quad \}$$

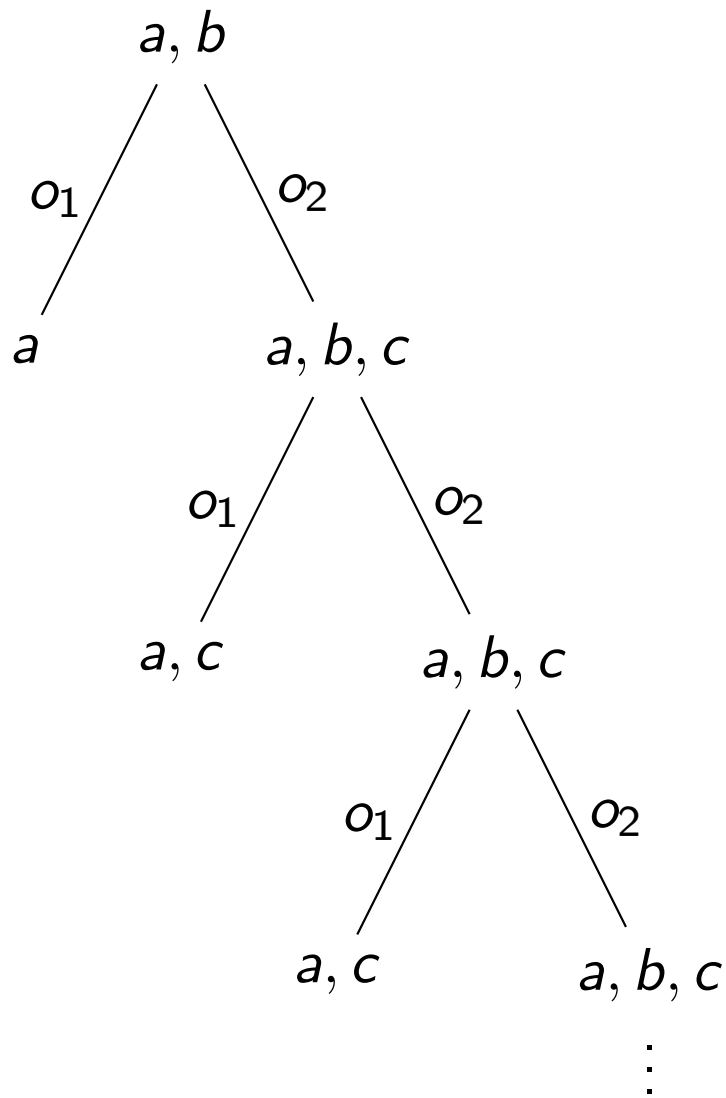
$$I = \{a, b\}$$

$$G = \{\neg b, c\}$$

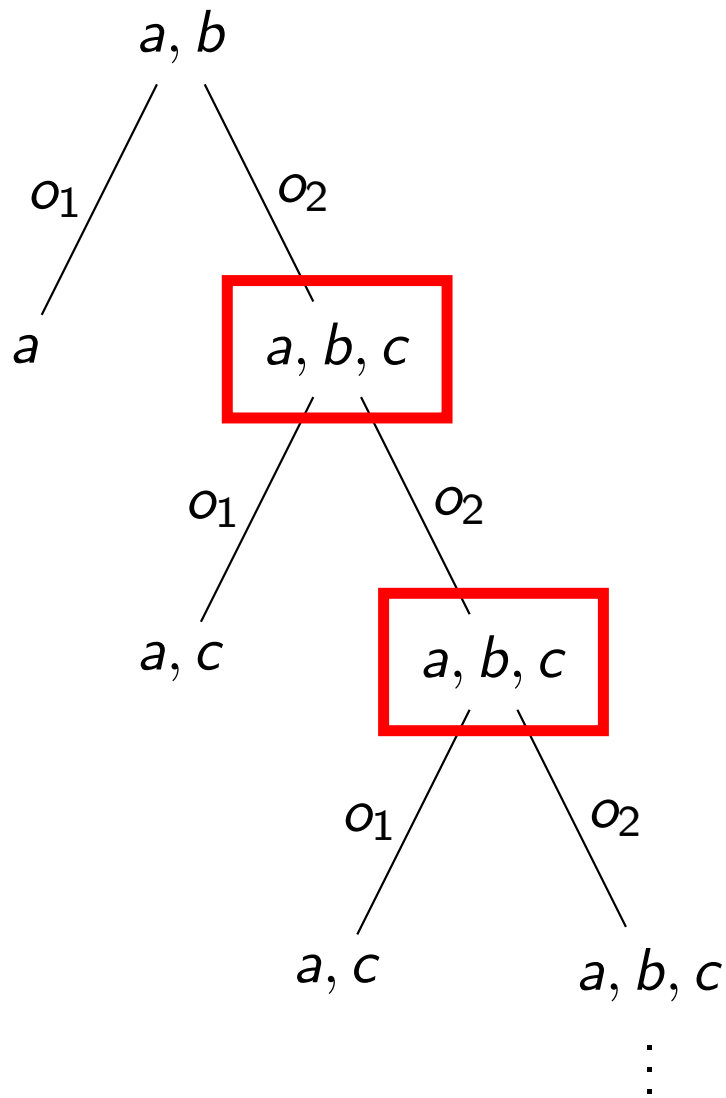
Dépliage



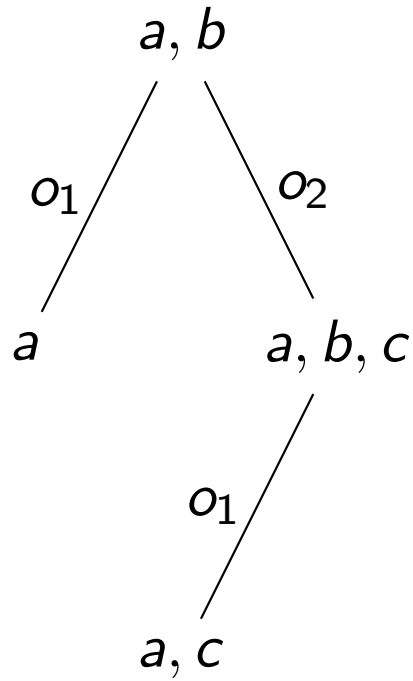
Dépliages de problèmes STRIPS et atteignabilité



Dépliages de problèmes STRIPS et atteignabilité



Dépliages de problèmes STRIPS et atteignabilité



Préfixe fini complet

Dépliages et solveurs SAT

Le dépliage en tant qu'instance de SAT

Idée de départ

La résolution de problèmes STRIPS par des solveurs SAT s'avère très efficace en pratique

$\hookrightarrow$ Peut-on étendre l'approche aux calculs de dépliages ?

Principe

Introduction de nouvelles variables pour coder le passé causal : un littéral de STRIPS sera instancié plusieurs fois en SAT

$$\ell, \ell@o_1, \ell@o_1 o_2, \ell@o_1 o_2 o_3, \ldots$$

Le dépliage en tant qu'instance de SAT

Idée de départ

La résolution de problèmes STRIPS par des solveurs SAT s'avère très efficace en pratique

$\hookrightarrow$ Peut-on étendre l'approche aux calculs de dépliages ?

Principe

Introduction de nouvelles variables pour coder le passé causal : un littéral de STRIPS sera instancié plusieurs fois en SAT

$$\ell, \ell@o_1, \ell@o_1 o_2, \ell@o_1 o_2 o_3, \ldots$$

Problème potentiel

Explosion des nombres de variables et de clauses... mais, en général, trouver un plan peut aussi demander un horizon k très grand

Dépliage et SAT, les clauses – état initial

État initial

$$\left(\bigwedge_{a \in I} a@\varepsilon\right) \wedge \left(\bigwedge_{a \in A \setminus I} \neg a@\varepsilon\right)$$

Dépliage et SAT, les clauses – pré-conditions

- $\pi = o_{i_1} o_{i_2} \ldots o_{i_k}$, une séquence d'opérateurs
- $o = (p, e)$, un opérateur

Dépliage et SAT, les clauses – pré-conditions

- $\pi = o_{i_1} o_{i_2} \ldots o_{i_k}$, une séquence d'opérateurs
- $o = (p, e)$, un opérateur

Si p est vérifiée après π on doit ajouter o

$$\bigwedge_{\ell \in p} \ell @ \pi \implies o @ \pi$$

Sinon on ne doit pas ajouter o

$$o @ \pi \implies \bigwedge_{\ell \in p} \ell @ \pi$$

Dépliage et SAT, les clauses – pré-conditions

- $\pi = o_{i_1} o_{i_2} \ldots o_{i_k}$, une séquence d'opérateurs
- $o = (p, e)$, un opérateur

Si p est vérifiée après π on doit ajouter o

$$\bigwedge_{\ell \in p} \ell @ \pi \implies o @ \pi$$

Sinon on ne doit pas ajouter o

$$o @ \pi \implies \bigwedge_{\ell \in p} \ell @ \pi$$

Tout ceci, seulement si π est possible

$$\neg o_{i_k} @ o_{i_1} o_{i_2} \ldots o_{i_{k-1}} \implies \neg o @ \pi$$

Dépliage et SAT, les clauses – effets

- $\pi = o_{i_1} o_{i_2} \ldots o_{i_k}$, une séquence d'opérateurs
- $o = (p, e)$, un opérateur

Appliquer les effets de o

$$o@\pi \implies \bigwedge_{\ell \in e} \ell@\pi o$$

Dépliage et SAT, les clauses – effets

- $\pi = o_{i_1} o_{i_2} \dots o_{i_k}$, une séquence d'opérateurs
- $o = (p, e)$, un opérateur

Appliquer les effets de o

$$o@\pi \implies \bigwedge_{\ell \in e} \ell@\pi o$$

Ne pas modifier les autres littéraux

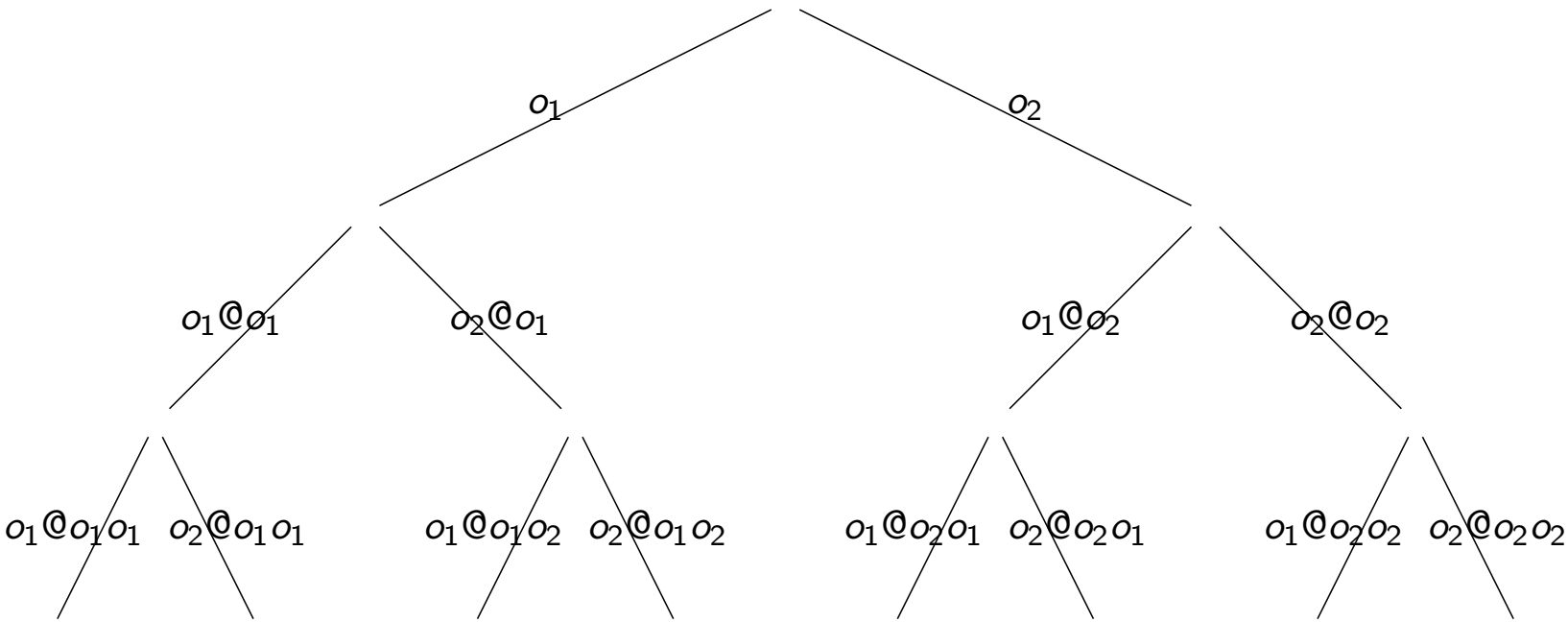
$$(\forall a, a \notin e, \neg a \notin e)$$

$$a@\pi \implies a@\pi o$$

$$\neg a@\pi \implies \neg a@\pi o$$

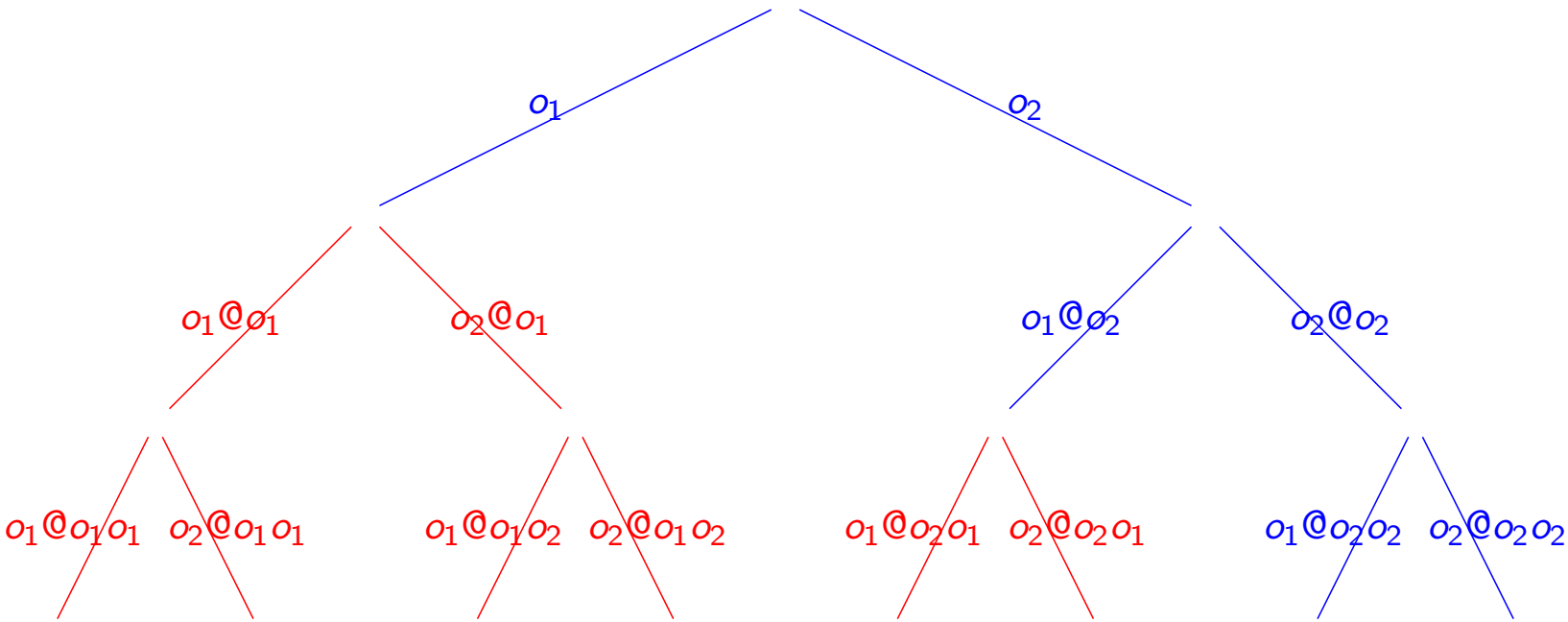
Retrouver le dépliage depuis une solution du problème SAT

$I = \{a, b\}$
$O = \{o_1 = (\{a\}, \{\neg b\}), o_2 = (\{a, b\}, \{c\})\}$



Retrouver le dépliage depuis une solution du problème SAT

$I = \{a, b\}$
$O = \{o_1 = (\{a\}, \{\neg b\}), o_2 = (\{a, b\}, \{c\})\}$



Quelques mesures

Influence du nombre de variables STRIPS sur la taille du problème SAT

Paramètres fixés : 5 opérateurs STRIPS, horizon/profondeur 3

STRIPS	**Variables**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**+1**
Plan	Variables	27	31	35	39	43	47	51	+4
	Clauses	83	90	97	104	111	118	125	+7
Dépliage	Variables	779	935	1091	1247	1403	1559	1715	+156
	Clauses	1553	1864	2175	2486	2797	3108	3419	+311

Influence du nombre d'opérateurs STRIPS sur la taille du problème SAT

Paramètres fixés : 3 variables STRIPS, horizon/profondeur 3

STRIPS	**Opérateurs**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**+1**
Plan	Variables	21	24	27	30	33	36	39	+3
	Clauses	50	65	83	104	128	155	185	-
Dépliage	Variables	199	424	779	1294	1999	2924	4099	-
	Clauses	393	843	1553	2583	3993	5843	8193	-

Influence de l'horizon/profondeur sur la taille du problème SAT

Paramètres fixés : 3 variables STRIPS, 5 opérateurs STRIPS

STRIPS	**Horizon**	**3**	**4**	**5**	**6**	**7**	**+1**
Plan	Variables	27	35	43	51	59	+8
	Clauses	83	109	135	161	187	+26
Dépliage	Variables	779	3904	19529	97654	488279	-
	Clauses	1553	7803	39053	195303	976553	-

Pour conclure

Possibilité de construire des dépliages à partir de solveurs SAT

- Même principe que SATPLAN
- Mais explosion de la taille des problèmes SAT

Pour conclure

Possibilité de construire des dépliages à partir de solveurs SAT

- Même principe que SATPLAN
- Mais explosion de la taille des problèmes SAT

Critère de complétude du préfixe

Global ou local ?

- Traditionnellement local (évènements *cut-off*)
- Intérêt d'un critère global ?

Concurrence

Coder la concurrence dans le dépliage

$\hookrightarrow$ Diminution potentielle de la profondeur

$\hookrightarrow$ Diminution importante de la taille du (plus grand) problème SAT à résoudre