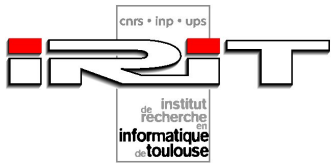
Compilation de connaissances avec automates à intervalles, et application à la planification

Alexandre Niveau (Onera/DCSD)
Hélène Fargier (IRIT/RPDMP), Cédric Pralet & Gérard Verfaillie (Onera/DCSD)



2 juin 2010

Plan

1. Introduction au problème
2. Automates à intervalles
3. Exploitation d'une politique
4. Automates à intervalles convergents
5. Construction de FIAs
6. Résultats

Conduite d'engins autonomes

Conduite d'engins autonomes : *tâches de suivi et de décision* en fonction des observations et objectifs courants ; à effectuer *en ligne*, avec les moyens de calcul embarqués.

Conduite d'engins autonomes

Conduite d'engins autonomes : *tâches de suivi et de décision* en fonction des observations et objectifs courants ; à effectuer *en ligne*, avec les moyens de calcul embarqués.

Cependant, elles sont généralement combinatoires, très gourmandes en temps de calcul.

- Intégralement réalisées en ligne, *la réactivité* du système ne sera pas assurée.
- Intégralement réalisées hors ligne (en anticipant toutes les situations possibles), *l'embarquabilité* sera compromise.

Conduite d'engins autonomes

Conduite d'engins autonomes : *tâches de suivi et de décision* en fonction des observations et objectifs courants ; à effectuer *en ligne*, avec les moyens de calcul embarqués.

Cependant, elles sont généralement combinatoires, très gourmandes en temps de calcul.

- Intégralement réalisées en ligne, *la réactivité* du système ne sera pas assurée.
- Intégralement réalisées hors ligne (en anticipant toutes les situations possibles), *l'embarquabilité* sera compromise.

Pour résoudre cette contradiction, on peut utiliser des techniques de *compilation de connaissances*.

Compilation de connaissances

Compilation de connaissances

- L'idée est de *transformer* le problème hors ligne, pour en faciliter la résolution en ligne.
- On « traduit » le problème dans un certain formalisme.
- Cette représentation permet de répondre efficacement aux requêtes.

Compilation de connaissances

Compilation de connaissances

- L'idée est de *transformer* le problème hors ligne, pour en faciliter la résolution en ligne.
- On « traduit » le problème dans un certain formalisme.
- Cette représentation permet de répondre efficacement aux requêtes.

En effectuant cette compilation hors ligne, on déporte donc une partie de l'effort de calcul avant la mise en situation du système.

Démarche

Cadre : problèmes de *planification forte non-déterministe*.

On s'intéresse ici à l'embarquement d'une solution complète, sous la forme d'une *politique de décision*, c'est-à-dire une fonction qui, à chaque état atteignable, associe des actions.

Démarche

Cadre : problèmes de *planification forte non-déterministe*.

On s'intéresse ici à l'embarquement d'une solution complète, sous la forme d'une *politique de décision*, c'est-à-dire une fonction qui, à chaque état atteignable, associe des actions.

$\longrightarrow$ Fonction à valeur booléenne, dépendant de variables

- d'*état*, dont chaque instanciation correspond à un état observable ;
- de *décision*, dont chaque instanciation correspond à une action.

Pour des instanciations $\vec{s}$ et $\vec{d}$ de ces variables, la fonction renvoie "vrai" ssi il convient de prendre la décision $\vec{d}$ dans l'état $\vec{s}$.

Objectif de l'article

Nombreuses structures de compilation utilisables :

- automates
- diagrammes de décision binaires (BDDs, FBDDs, OBDDs), algébriques (ADDs), et leurs affiliés
- formes normales négatives (NNFs), et leurs affiliés

Objectif de l'article

Nombreuses structures de compilation utilisables :

- automates
- diagrammes de décision binaires (BDDs, FBDDs, OBDDs), algébriques (ADDs), et leurs affiliés
- formes normales négatives (NNFs), et leurs affiliés

Cependant elles imposent de n'utiliser que des variables Booléennes ou à petit domaine énuméré.

Or, la conduite de systèmes autonomes peut nécessiter des variables continues ou à grand domaine énuméré, telles que le temps ou l'énergie.

Objectif de l'article

Nombreuses structures de compilation utilisables :

- automates
- diagrammes de décision binaires (BDDs, FBDDs, OBDDs), algébriques (ADDs), et leurs affiliés
- formes normales négatives (NNFs), et leurs affiliés

Cependant elles imposent de n'utiliser que des variables Booléennes ou à petit domaine énuméré.

Or, la conduite de systèmes autonomes peut nécessiter des variables continues ou à grand domaine énuméré, telles que le temps ou l'énergie.

Objectif : définir de nouvelles structures représentant des fonctions booléennes dépendant de variables continues (ou à grand domaine énuméré).

Plan

1 Introduction au problème

2 Automates à intervalles

3 Exploitation d'une politique

4 Automates à intervalles convergents

5 Construction de FIAs

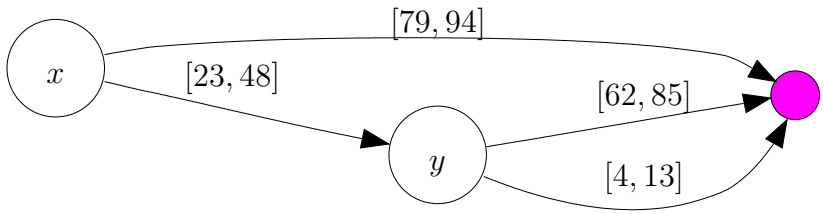
6 Résultats

Automates à intervalles

Nous avons décrit la structure des *automates à intervalles*.
Hypothèse : domaine d'une variable est une union *finie* d'intervalles fermés de $\mathbb{R}$.

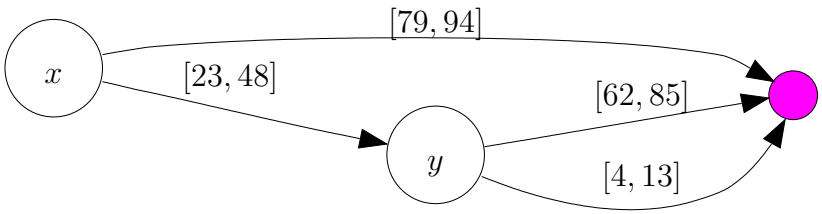
Automates à intervalles

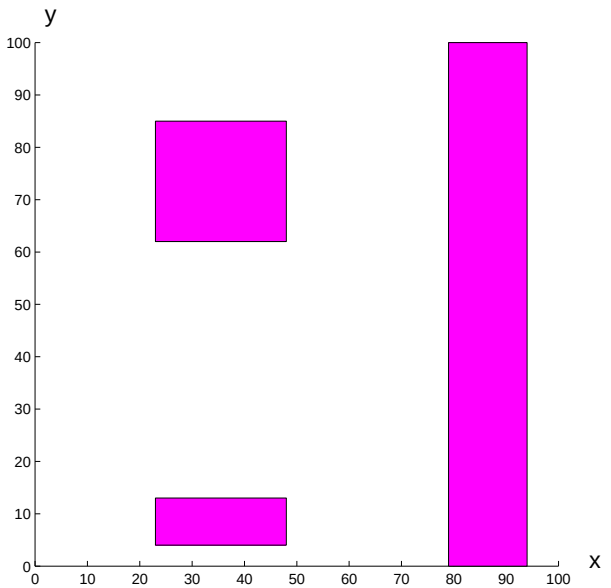
Nous avons décrit la structure des *automates à intervalles*.
Hypothèse : domaine d'une variable est une union *finie* d'intervalles fermés de $\mathbb{R}$.



Automates à intervalles

Nous avons décrit la structure des *automates à intervalles*.
Hypothèse : domaine d'une variable est une union *finie* d'intervalles fermés de $\mathbb{R}$.

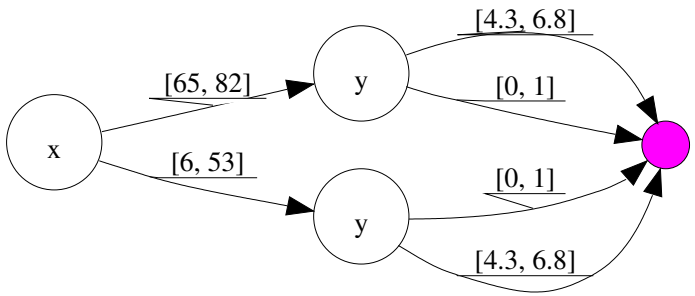




Chaque automate à intervalles (IA) représente une fonction booléenne sur les variables considérées, ou, de manière équivalente, un ensemble de solutions.

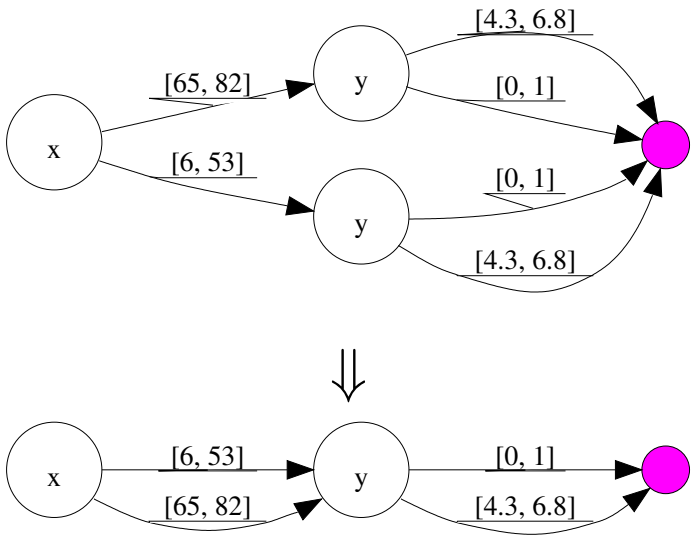
Réduction d'un IA : fusion des nœuds isomorphes

Nous avons décrit des opérations permettant de réduire la taille d'un IA sans changer sa sémantique.



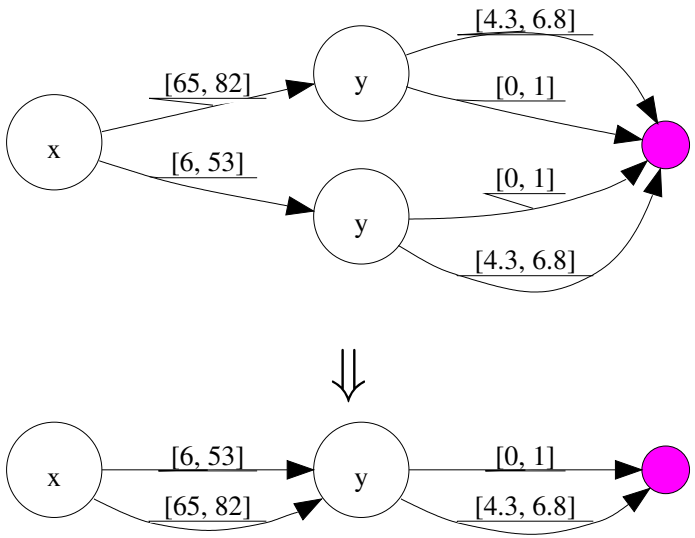
Réduction d'un IA : fusion des nœuds isomorphes

Nous avons décrit des opérations permettant de réduire la taille d'un IA sans changer sa sémantique.



Réduction d'un IA : fusion des nœuds isomorphes

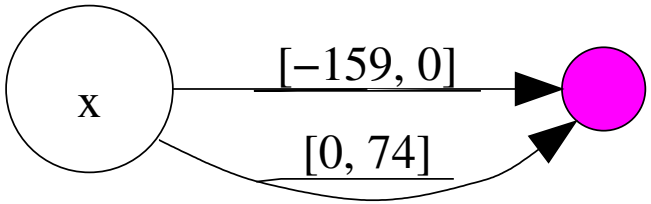
Nous avons décrit des opérations permettant de réduire la taille d'un IA sans changer sa sémantique.



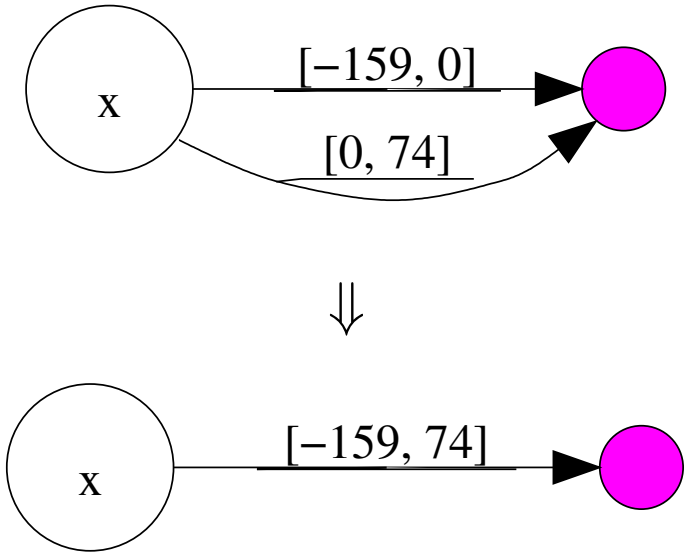
Nœuds isomorphes

Deux nœuds N_1, N_2 d'un IA ϕ sont *isomorphes* ssi $\mathrm{Var}(N_1) = \mathrm{Var}(N_2)$ et il existe une bijection σ de $\mathrm{Out}(N_1)$ dans $\mathrm{Out}(N_2)$, telle que $\forall E \in \mathrm{Out}(N_1), \mathrm{Itv}(E) = \mathrm{Itv}(\sigma(E))$ et $\mathrm{Dest}(E) = \mathrm{Dest}(\sigma(E))$.

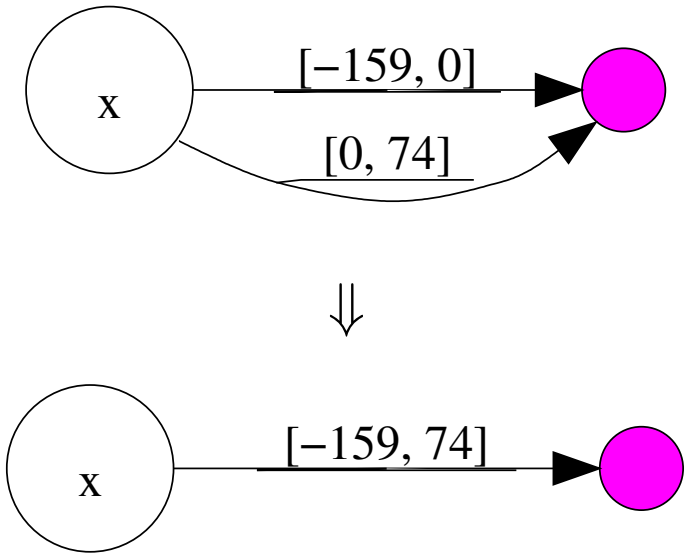
Réduction d'un IA : fusion des arcs contigus



Réduction d'un IA : fusion des arcs contigus



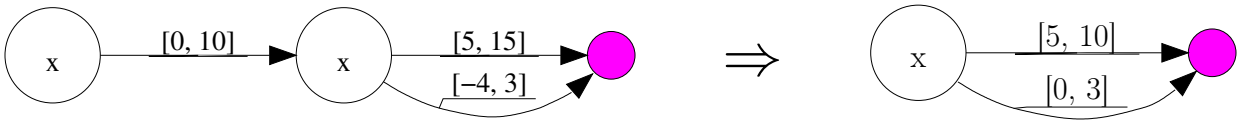
Réduction d'un IA : fusion des arcs contigus



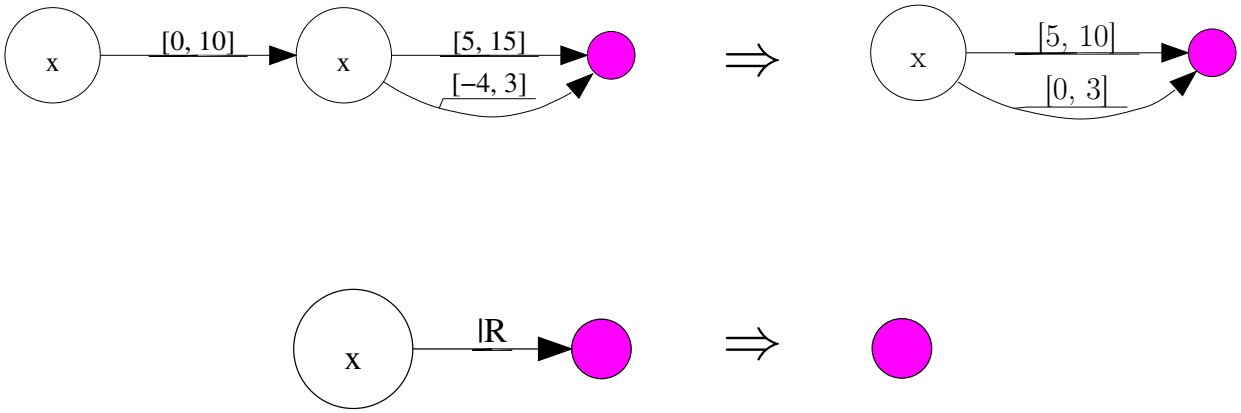
Arcs contigus

Deux arcs E_1, E_2 d'un IA ϕ sont *contigus* ssi $\mathrm{Src}(E_1) = \mathrm{Src}(E_2)$, $\mathrm{Dest}(E_1) = \mathrm{Dest}(E_2)$ et il existe un intervalle $I \in \mathbb{R}$ tel que $\mathrm{Itv}(E_1) \cup \mathrm{Itv}(E_2) = I \cap \mathrm{Dom}(\mathrm{Var}(E_1))$.

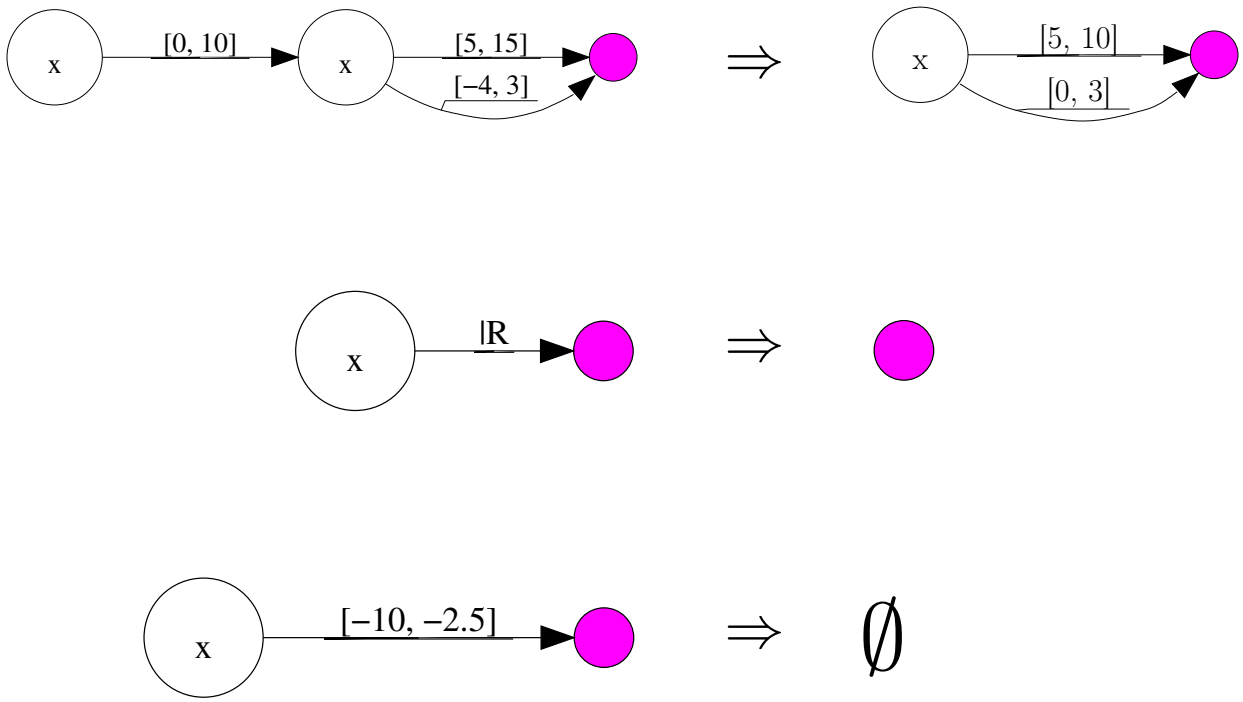
Réduction d'un IA



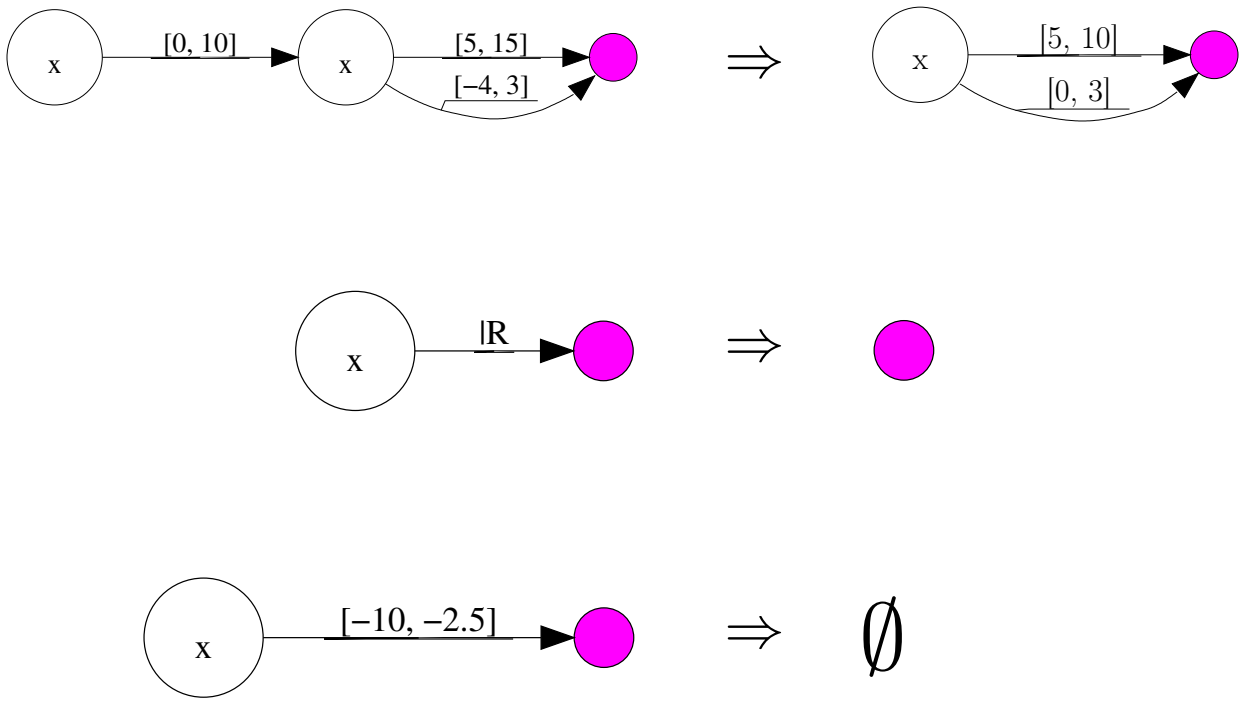
Réduction d'un IA



Réduction d'un IA



Réduction d'un IA

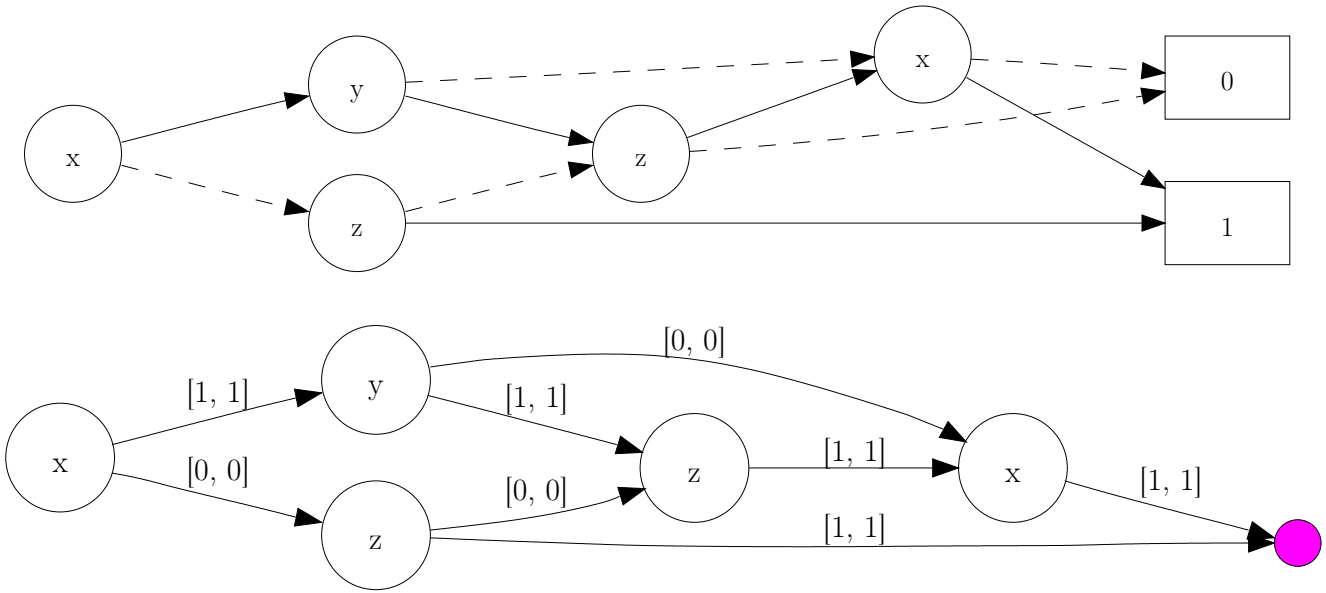


Proposition : réduction d'un IA

Il existe un algorithme polynomial qui transforme n'importe quel IA en un IA réduit équivalent.

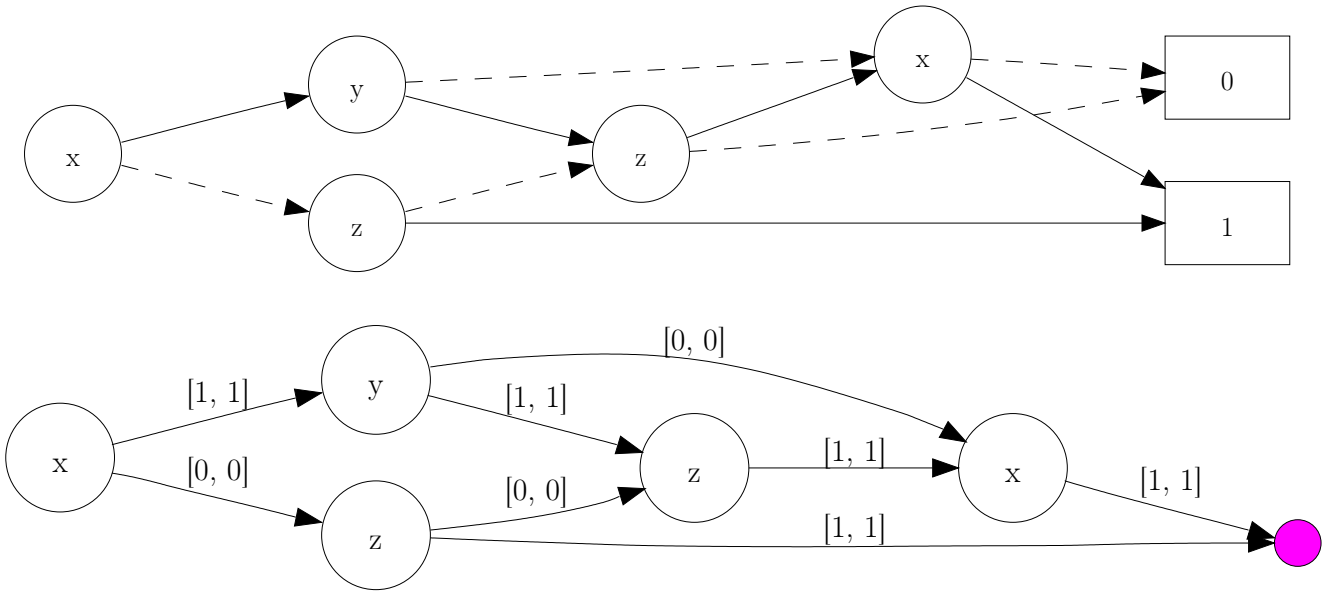
Relation avec BDD

Les IAs sont une généralisation des BDDs aux variables continues : les BDDs sont des IAs particuliers (variables Booléennes, nœuds déterministes).



Relation avec BDD

Les IAs sont une généralisation des BDDs aux variables continues : les BDDs sont des IAs particuliers (variables Booléennes, nœuds déterministes).



Comme dans un BDD, on peut répéter plusieurs fois la même variable sur un chemin, et l'ordre n'a pas d'importance.

Plan

1 Introduction au problème

2 Automates à intervalles

3 Exploitation d'une politique

4 Automates à intervalles convergents

5 Construction de FIAs

6 Résultats

Utilisation d'une politique

Pour utiliser en ligne des politiques de décision compilées en IA, deux opérations nécessaires :

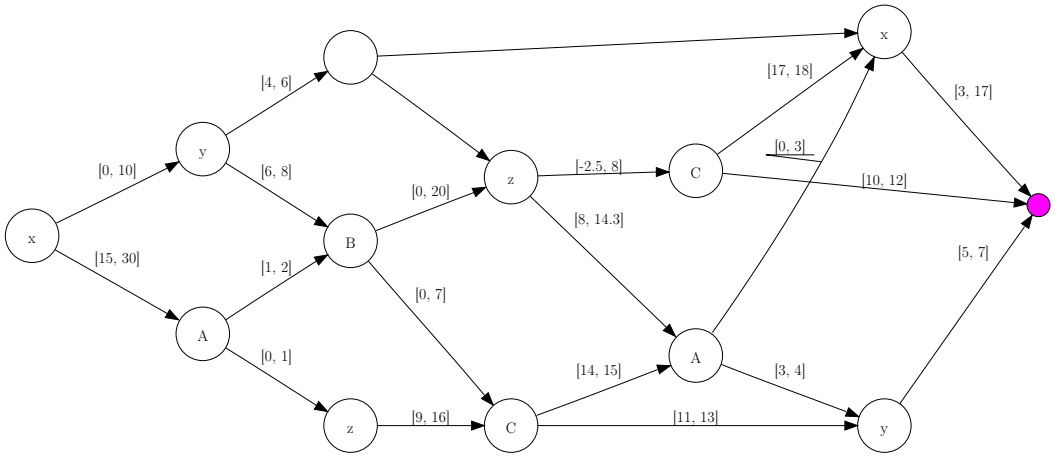
- le conditionnement (assigne les variables d'état conformément à ce qui est observé) ;
- l'extraction de modèle (produit une décision parmi toutes celles possibles).

Utilisation d'une politique

Pour utiliser en ligne des politiques de décision compilées en IA, deux opérations nécessaires :

- le conditionnement (assigne les variables d'état conformément à ce qui est observé) ;
- l'extraction de modèle (produit une décision parmi toutes celles possibles).

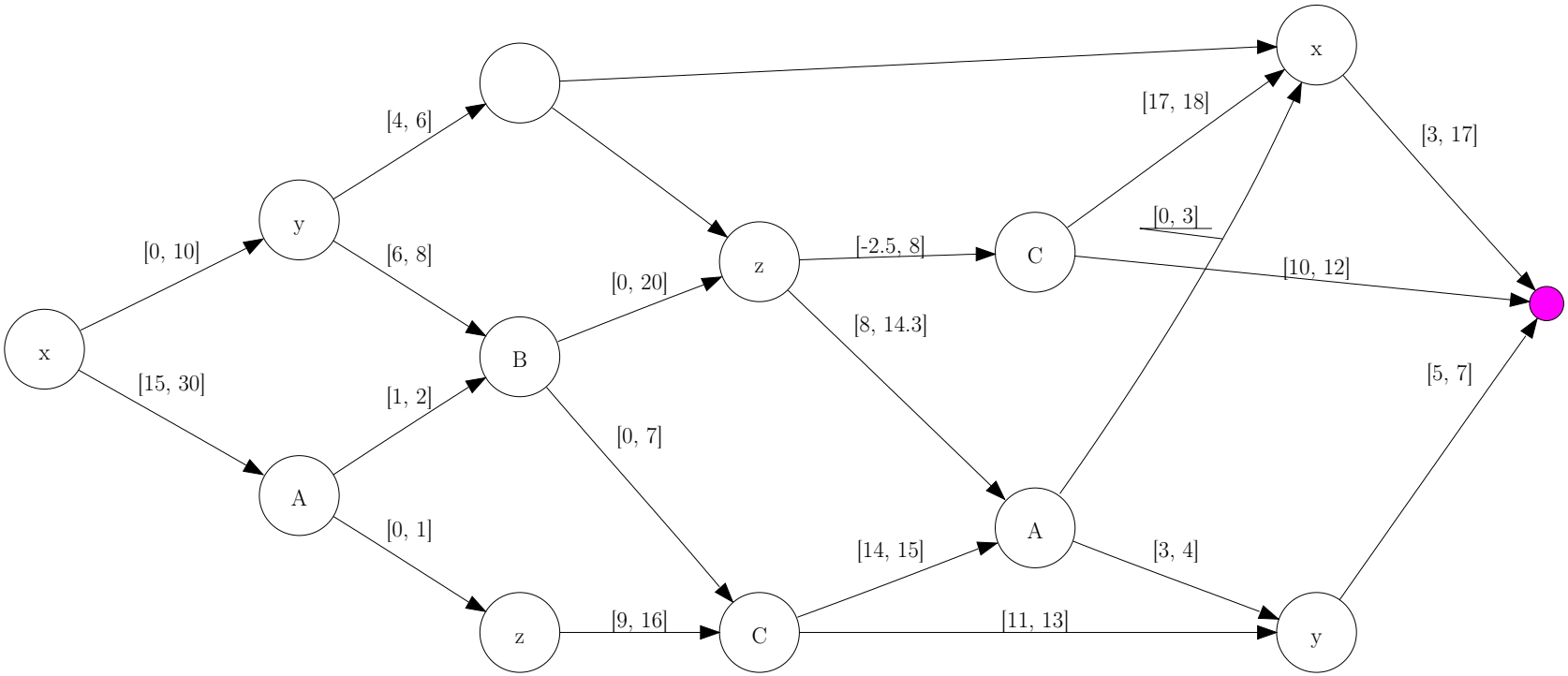
Illustration sur un exemple simple, avec trois variables d'état et trois variables de décision :



Conditionnement

On observe l'état $x = 17, y = 6, z = 8$.

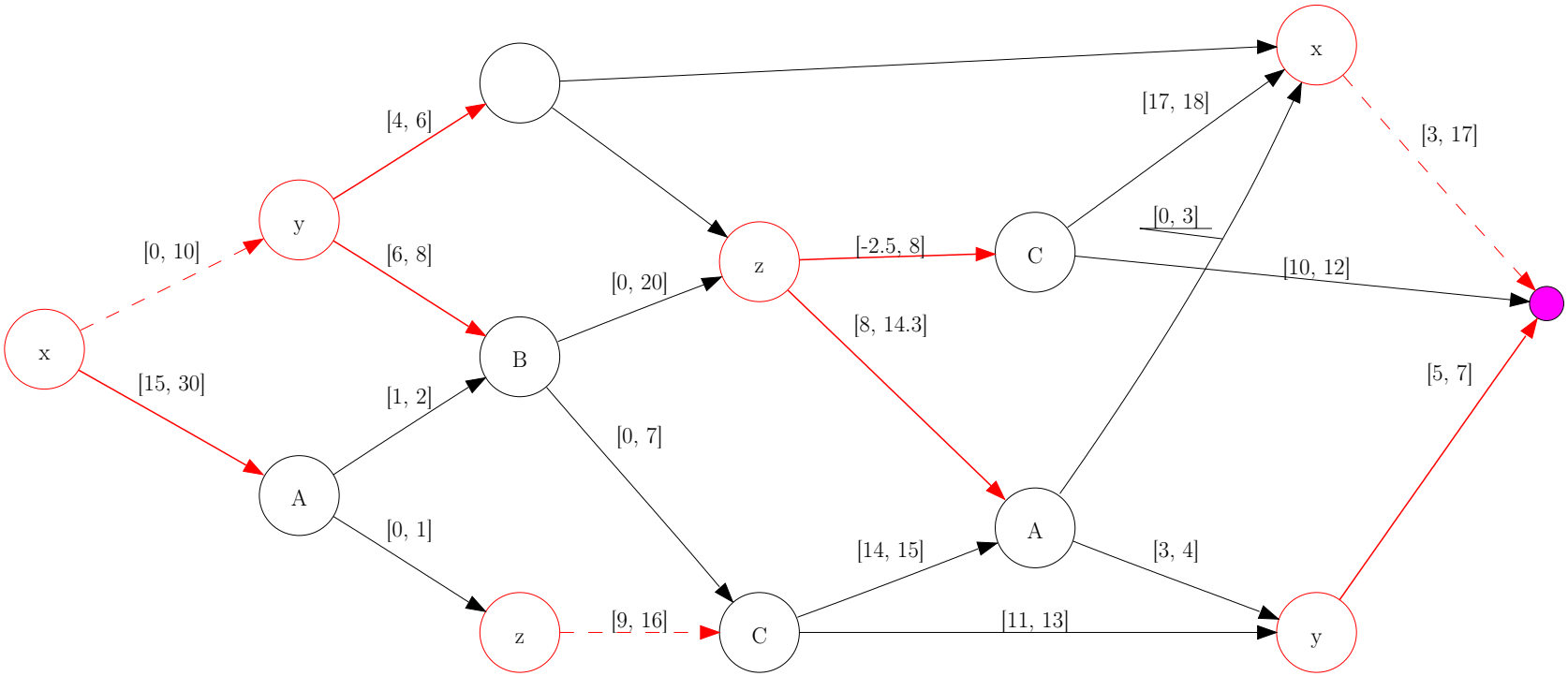
$\longrightarrow$ Conditionnement de la politique :



Conditionnement

On observe l'état $x = 17, y = 6, z = 8$.

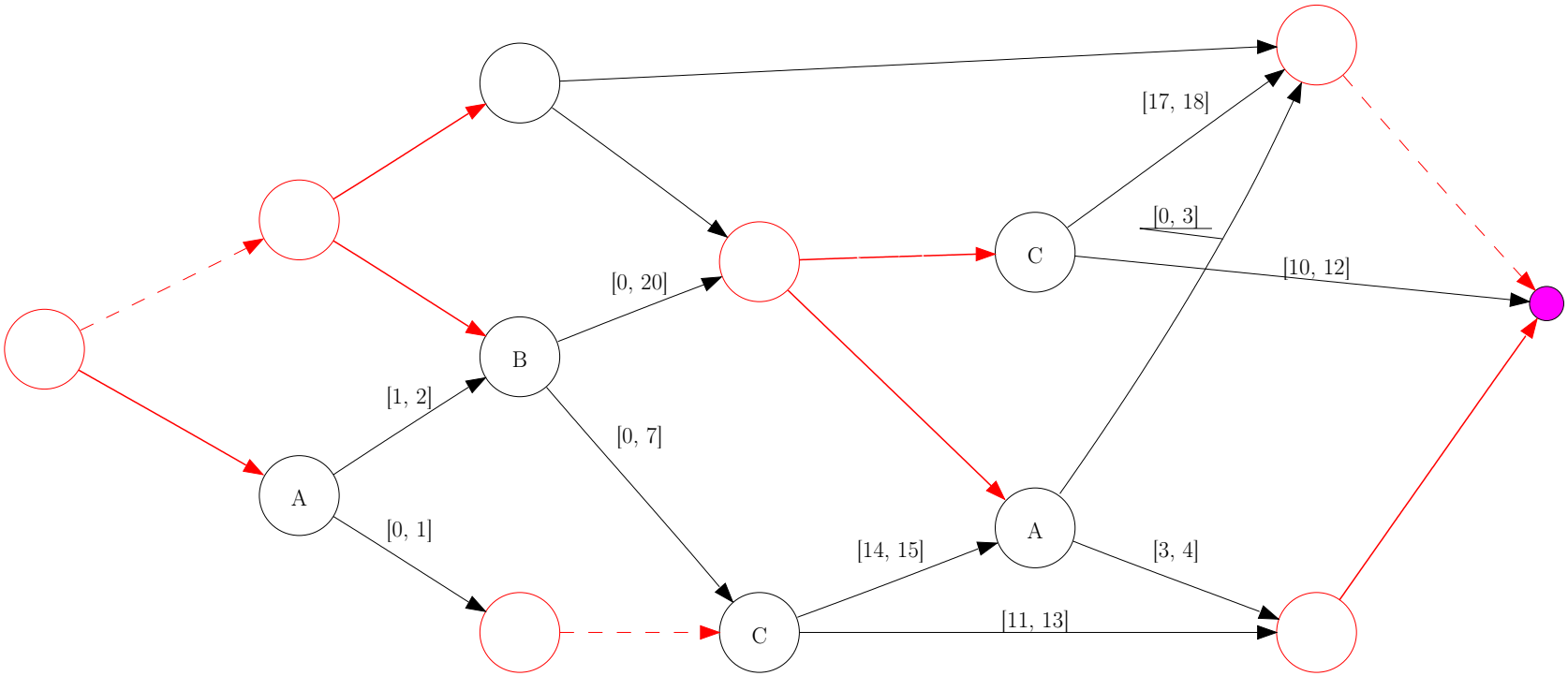
$\longrightarrow$ Conditionnement de la politique :



Conditionnement

On observe l'état $x = 17, y = 6, z = 8$.

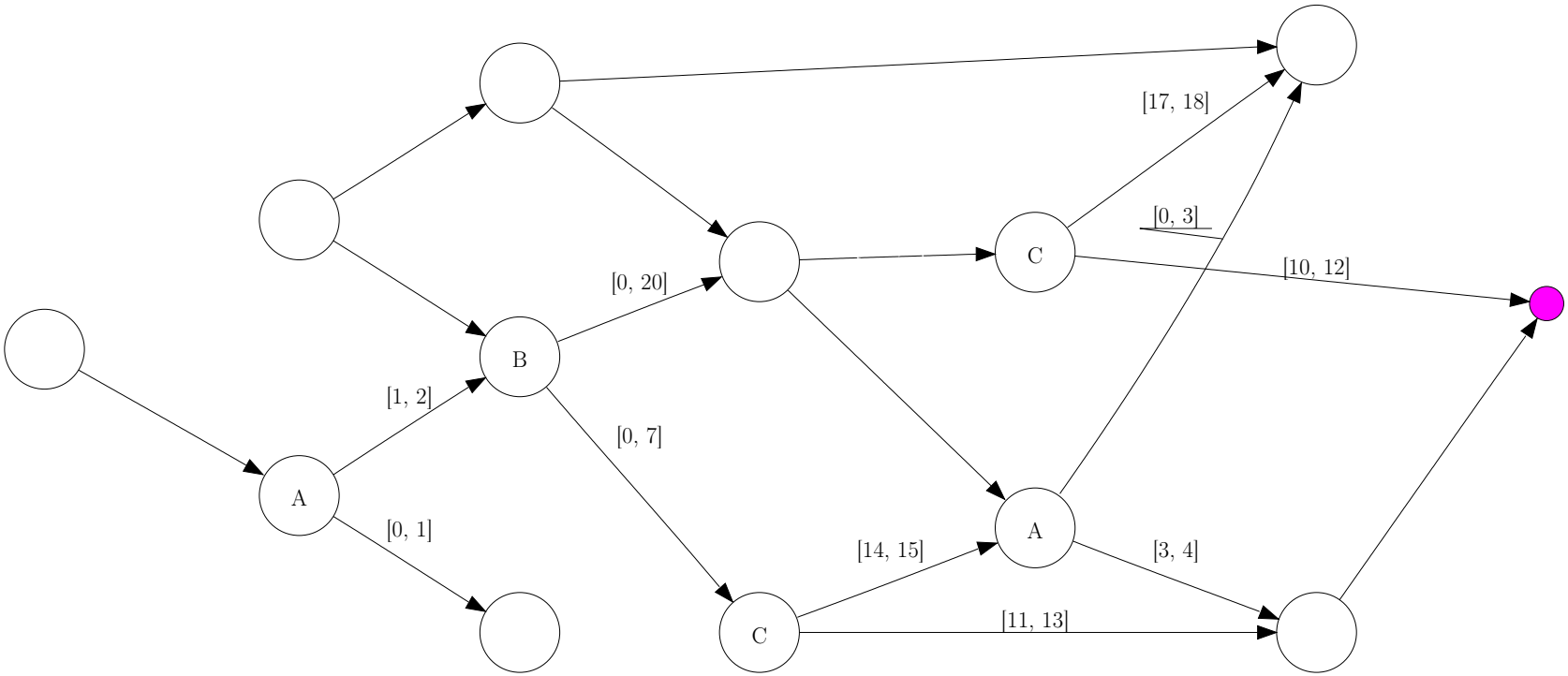
$\longrightarrow$ Conditionnement de la politique :



Conditionnement

On observe l'état $x = 17, y = 6, z = 8$.

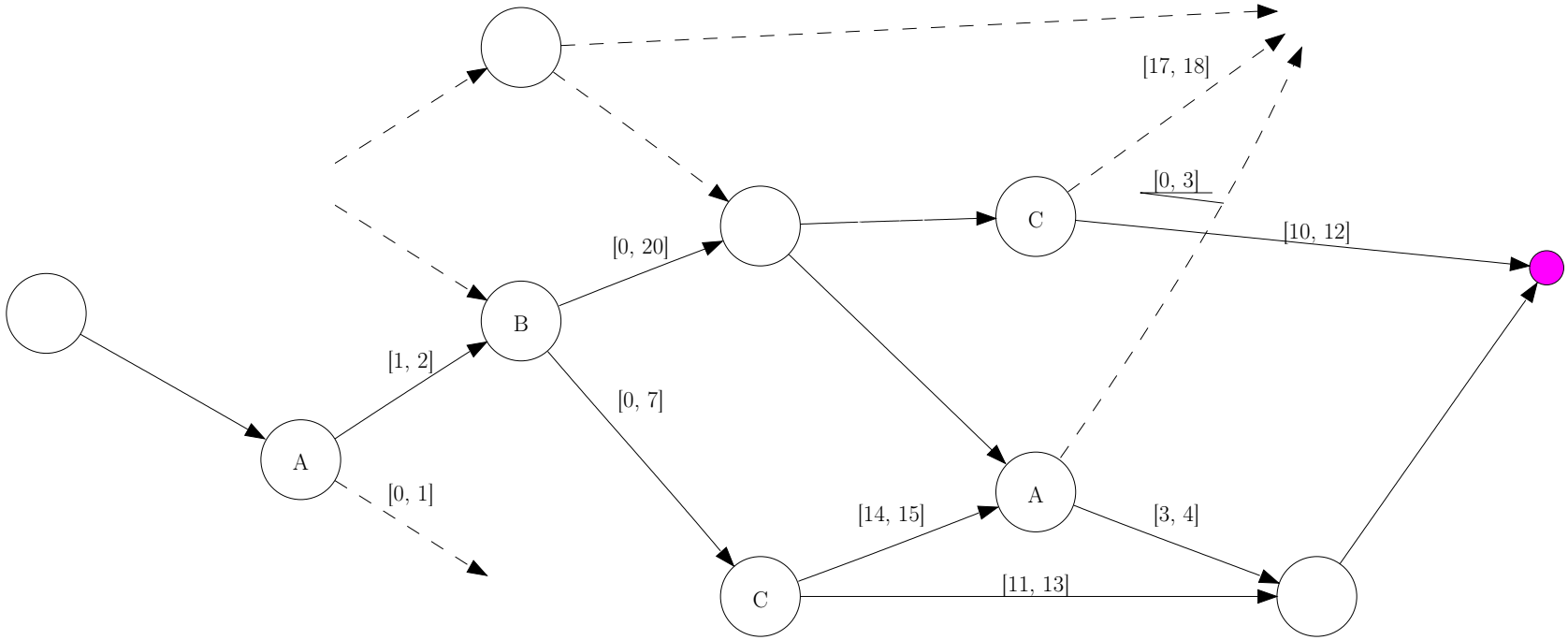
$\longrightarrow$ Conditionnement de la politique :



Conditionnement

On observe l'état $x = 17, y = 6, z = 8$.

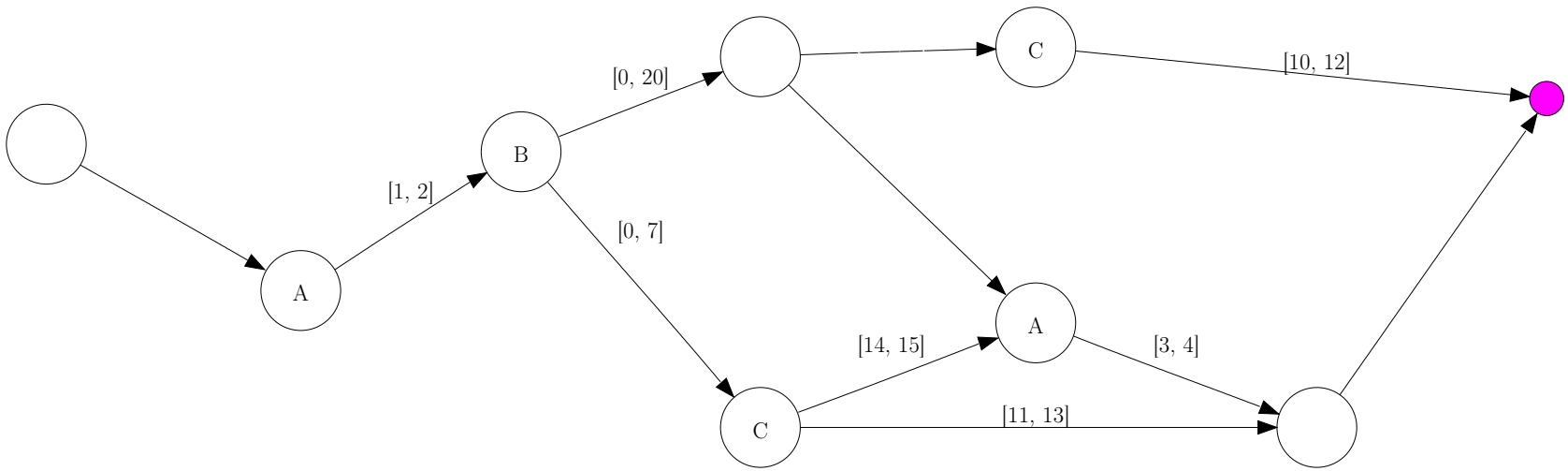
$\longrightarrow$ Conditionnement de la politique :



Conditionnement

On observe l'état $x = 17, y = 6, z = 8$.

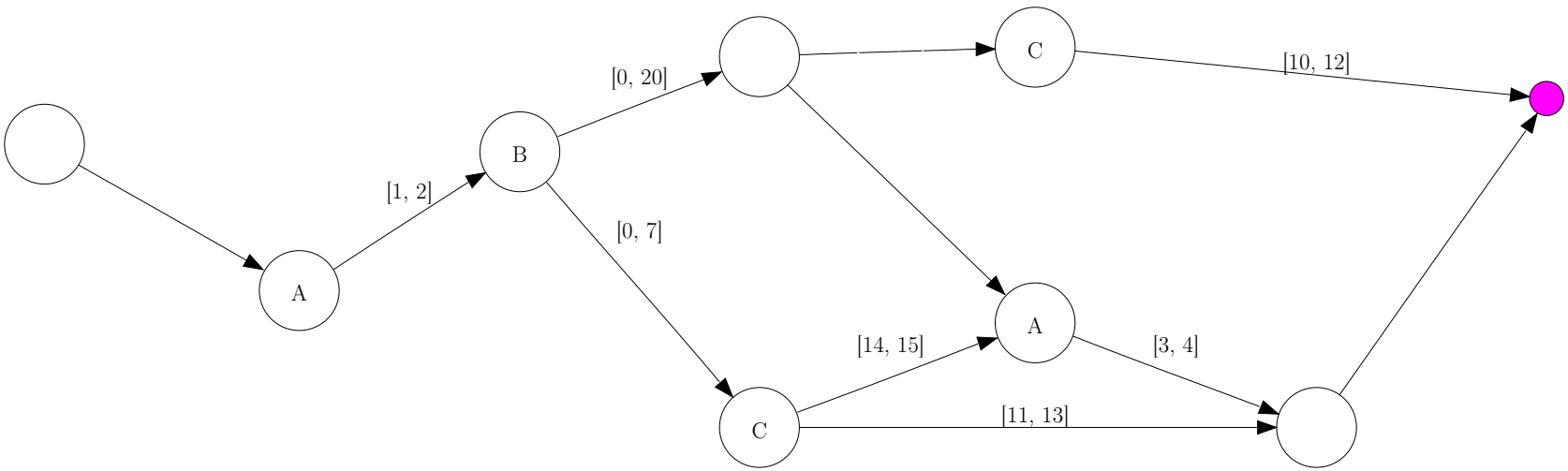
$\longrightarrow$ Conditionnement de la politique :



Extraction de modèle

On obtient un ensemble de décisions possibles.

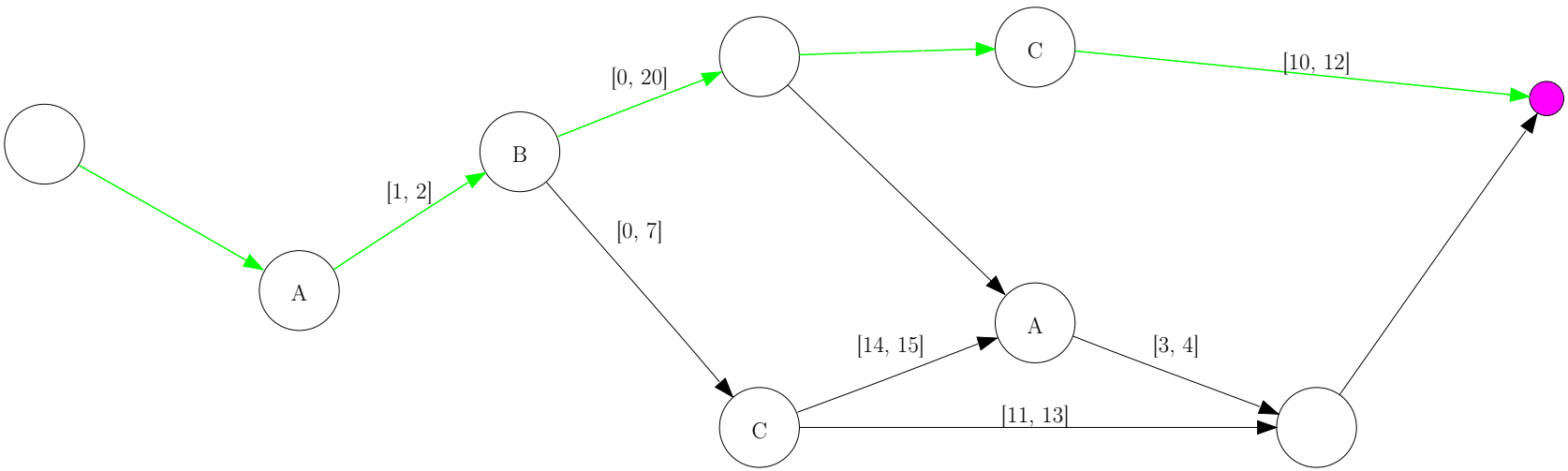
On veut en choisir une :



Extraction de modèle

On obtient un ensemble de décisions possibles.

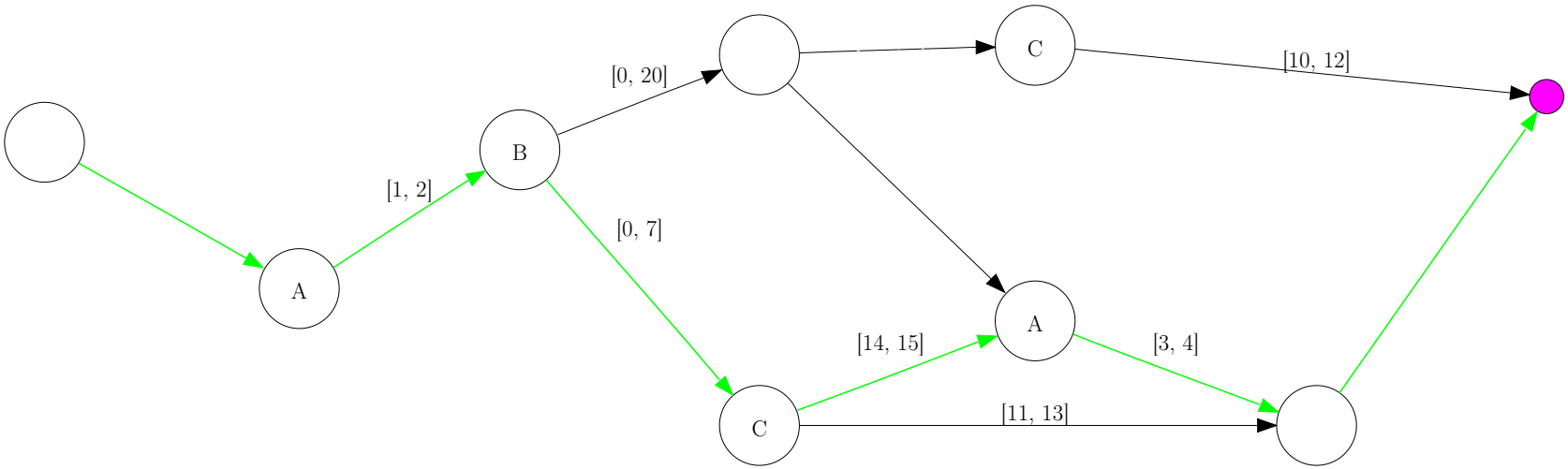
On veut en choisir une :



Extraction de modèle

On obtient un ensemble de décisions possibles.

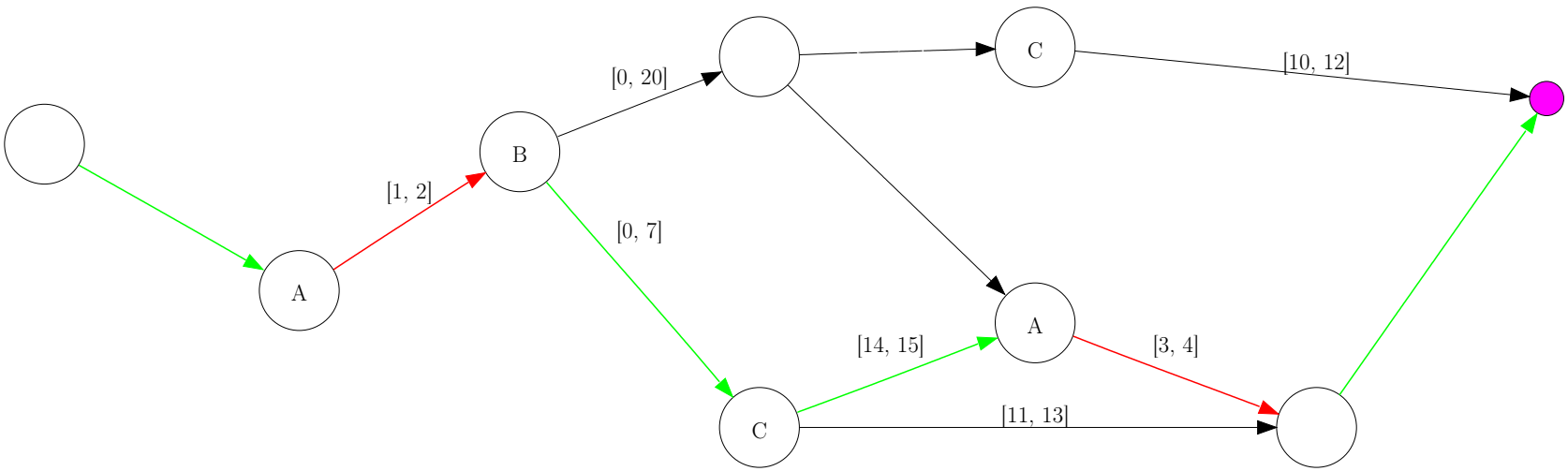
On veut en choisir une :



Extraction de modèle

On obtient un ensemble de décisions possibles.

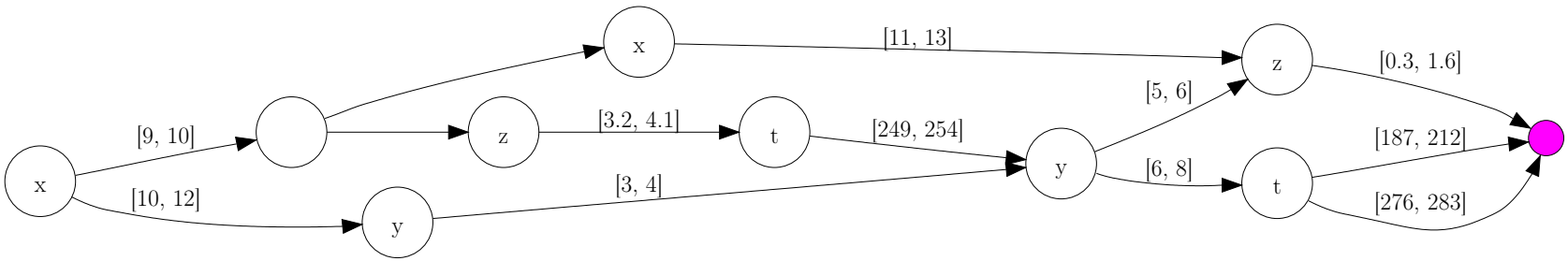
On veut en choisir une :



Extraction de modèle

Le problème est que les chemins d'un IA peuvent être inconsistants.

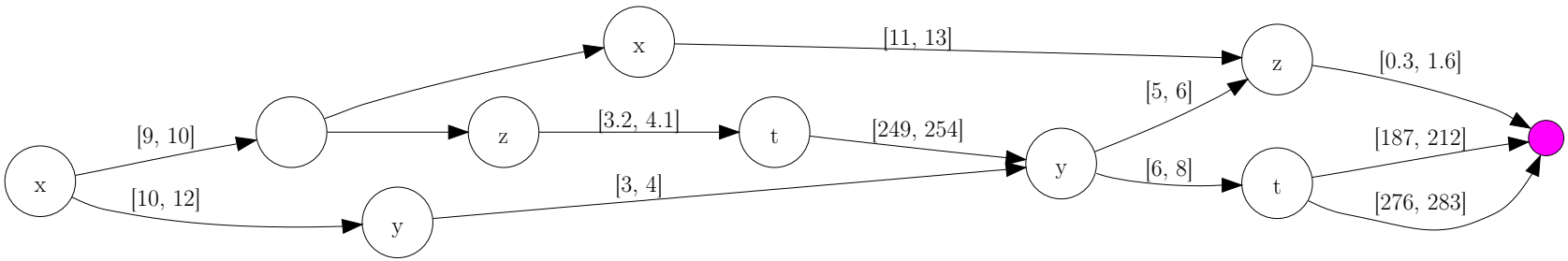
Un IA peut même n'avoir aucun chemin consistant :



Extraction de modèle

Le problème est que les chemins d'un IA peuvent être inconsistants.

Un IA peut même n'avoir aucun chemin consistant :



Comme pour les BDDs, cela rend l'extraction de modèle difficile.

$\longrightarrow$ restriction sur les IAs, pour faciliter cette opération.

Plan

1 Introduction au problème

2 Automates à intervalles

3 Exploitation d'une politique

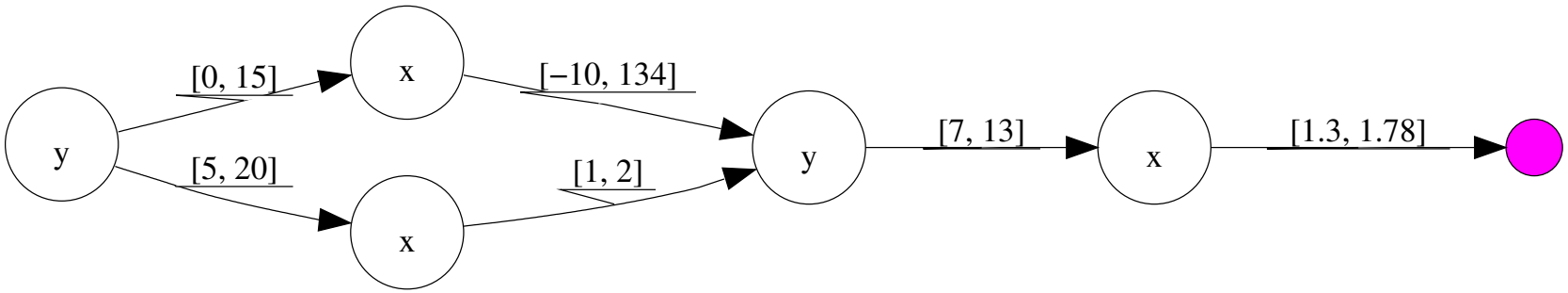
4 Automates à intervalles convergents

5 Construction de FIAs

6 Résultats

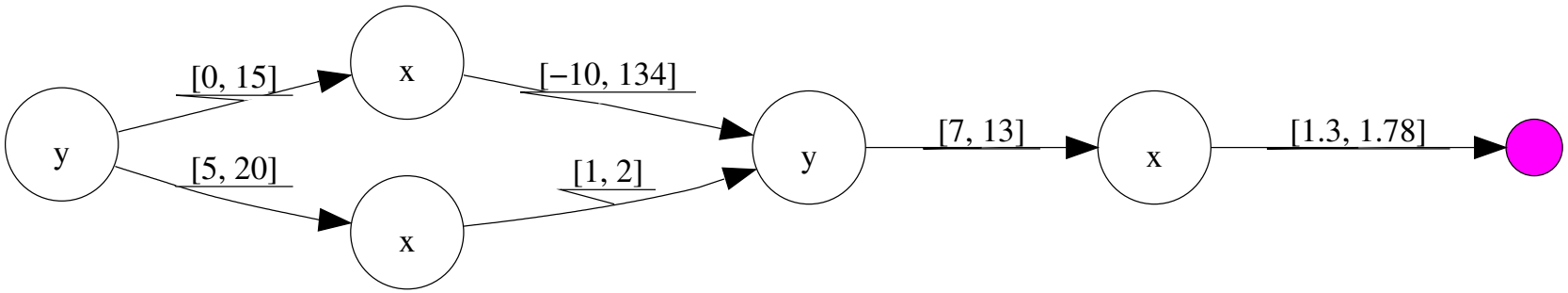
Automates à intervalles convergents (FIAs)

Idée : contraindre les intervalles à ne pouvoir que diminuer le long d'un chemin ; on appelle cette propriété la *convergence*.



Automates à intervalles convergents (FIAs)

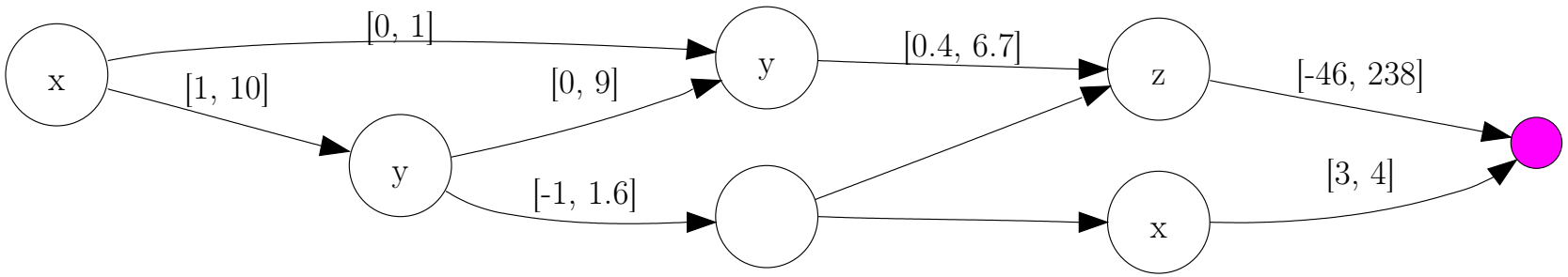
Idée : contraindre les intervalles à ne pouvoir que diminuer le long d'un chemin ; on appelle cette propriété la *convergence*.



Les FIAs (*focusing IAs*) permettent d'effectuer les deux opérations désirées en temps polynomial.

Extraction de modèle sur les FIAs

Chaque chemin d'un FIA réduit correspond à au moins un modèle :



Relation avec FBDD

Cette restriction rend l'extraction de modèle facile, exactement comme la restriction « read-once » sur les BDDs, qui donne les FBDDs.

Les FBDDs (et OBDDs) sont des FIAs particuliers.

Relation avec FBDD

Cette restriction rend l'extraction de modèle facile, exactement comme la restriction « read-once » sur les BDDs, qui donne les FBDDs.

Les FBDDs (et OBDDs) sont des FIAs particuliers.

Outre les deux opérations précédemment citées, nous avons vérifié la faisabilité sur les IAs et FIAs des opérations classiquement étudiées dans le contexte de la compilation de connaissances.

Opérations supportées par IAs et FIAs

Tableaux des opérations supportées :

Requêtes	IA	FIA
CO(consistance)	○	√
VA(validité)	○	○
EQ(équivalence)	○	○
MC(model checking)	√	√
MX(extr. de modèle)	○	√
CX(extr. de contexte)	○	√
ME(énum. de modèles)	○	√

Transformations	IA	FIA
CD(conditionnement)	√	√
∧**tC**(conj. avec un terme)	√	√
FO(proj. existentielle)	○	√
SFO(proj. exist. simple)	√	√
EN(proj. universelle)	○	○
SEN(proj. univ. simple)	√	○
∧**C**(conjonction)	√	○
∧**BC**(conj. binaire)	√	○
∨**C**(disjonction)	√	√

√ signifie « satisfait », ○ « ne satisfait pas, excepté si P = NP ».

Plan

1 Introduction au problème

2 Automates à intervalles

3 Exploitation d'une politique

4 Automates à intervalles convergents

5 Construction de FIAs

6 Résultats

Construction de FIAs

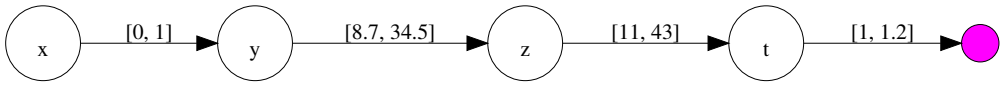
- Il est immédiat de construire le FIA représentant une « boîte »

$$[0, 1] \times [8.7, 34.5] \times [11, 43] \times [1, 1.2]$$

Construction de FIAs

- Il est immédiat de construire le FIA représentant une « boîte »

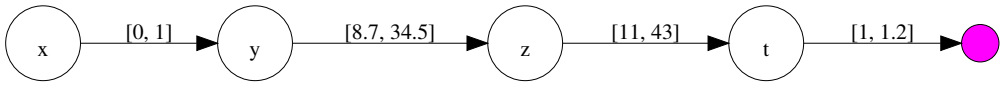
$$[0, 1] \times [8.7, 34.5] \times [11, 43] \times [1, 1.2]$$



Construction de FIAs

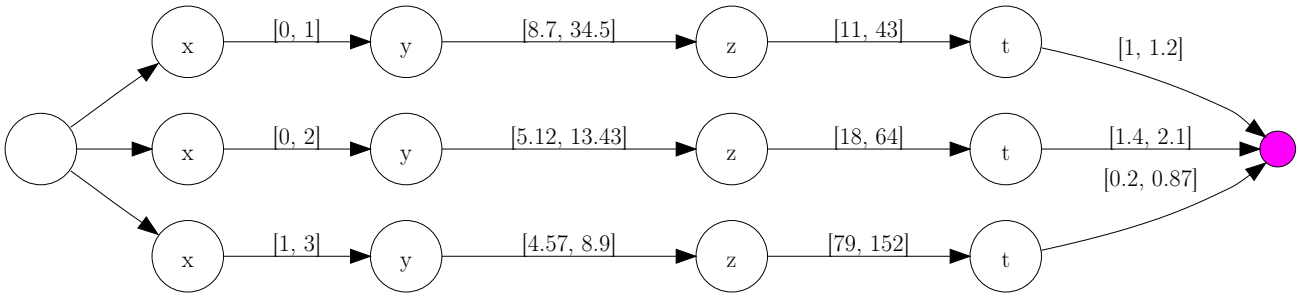
- Il est immédiat de construire le FIA représentant une « boîte »

$$[0,1] \times [8.7, 34.5] \times [11, 43] \times [1, 1.2]$$



- La disjonction ($\vee\mathbf{C}$) est facile sur les FIAs

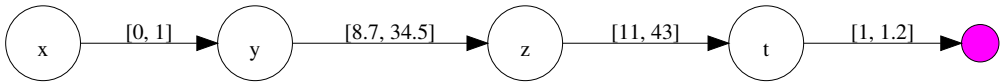
$\longrightarrow$ On peut facilement obtenir des FIAs à partir d'union de boîtes.



Construction de FIAs

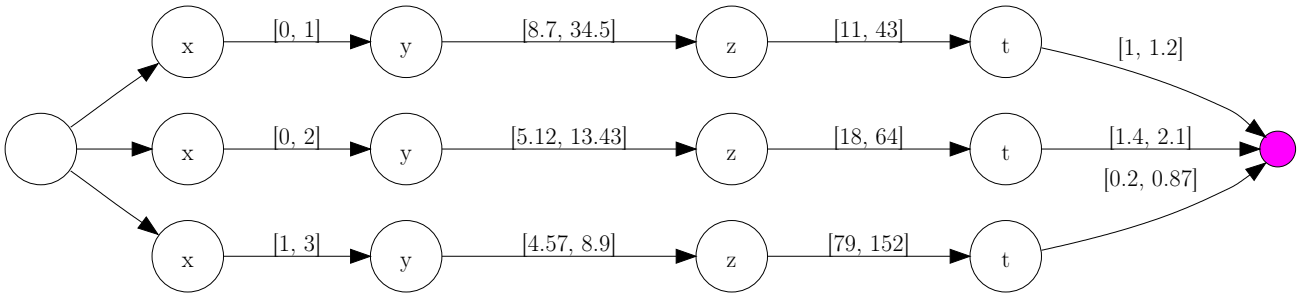
- Il est immédiat de construire le FIA représentant une « boîte »

$$[0,1] \times [8.7, 34.5] \times [11, 43] \times [1, 1.2]$$



- La disjonction ($\vee\mathbf{C}$) est facile sur les FIAs

$\longrightarrow$ On peut facilement obtenir des FIAs à partir d'union de boîtes.



Cette méthode s'applique notamment à la compilation de politiques fournies par un algorithme sous forme de liste de boîtes (et en particulier de tuples) admissibles.

Construction de FIAs

Autre méthode : utiliser la trace d'un solveur de CSP basé sur les intervalles, à la manière de [Huang and Darwiche, 2005].

Nous avons utilisé cette méthode avec le solveur RealPaver [Granvilliers and Benhamou, 2006].

Principe

Le solveur découpe les domaines jusqu'à trouver une boîte entièrement contenue dans l'ensemble-solution, ou de taille inférieure à un seuil donné. Notre algorithme crée à ce moment-là de nouveaux nœuds, et les fusionne avec le FIA courant.

Plan

1 Introduction au problème

2 Automates à intervalles

3 Exploitation d'une politique

4 Automates à intervalles convergents

5 Construction de FIAs

6 Résultats

Problèmes

`obsmem` : régit les connexions entre l'appareil d'observation et la mémoire de masse d'un satellite.

`robot` : un robot explorant un environnement.

`ring` : *benchmark* standard de planification non-déterministe.

`drone` : un drone doit remplir différents objectifs sur un certain nombre de zones en temps limité. (deux versions : discrète et hybride)

Résultats

problem	durée réd. (ms)	taille (arcs)	% entrée	% OBDD	**CD**(ms)	**MX**(ms)
obsmem2	1102	100	74	66	1	5
obsmem3	2168	197	75	69	4	11
obsmem4	4729	342	75	70	4	11
obsmem5	5657	546	76	70	7	19
obsmem6	9433	820	76	76	11	35
porobot	4035	56	97	36	0	1
forobot	52767	60	99	31	0	3
ring7	92	13	75	71	0	1
ring8	185	13	78	75	0	1
ring9	92	13	80	75	0	1
ring10	82	13	81	75	0	2
drone10	46732	453	95	47	11	23
drone20	947174	763	97	44	30	61
drone30	2850715	944	98	43	21	48
drone40	5721059	944	98	45	15	29
drone10	104373	16820	35	×	7143	110
drone20	418885	38076	35	×	16970	193
drone30	1850326	53917	36	×	23597	612

Conclusion

Nous avons décrit un langage-cible de compilation permettant théoriquement de représenter et d'exploiter en ligne des politiques de décision à variables discrètes et continues.

Gains en taille significatifs par rapport aux OBDDs.

Durée des opérations intéressante, bien qu'à améliorer.

Conclusion

Nous avons décrit un langage-cible de compilation permettant théoriquement de représenter et d'exploiter en ligne des politiques de décision à variables discrètes et continues.

Gains en taille significatifs par rapport aux OBDDs.

Durée des opérations intéressante, bien qu'à améliorer.

Pour la suite : Amélioration des algorithmes, et comparaison avec d'autres structures, non-Booléennes (automates de Vempaty [Vempaty, 1992]).

Perspectives

En fonction de la pertinence des résultats obtenus :

- Approfondir méthodes de compilation des FIAs (heuristiques...)
- Intégration de probabilités dans les IAs, compilation partielle [Venturini and Provan, 2008]
- Formalismes différents (compilation modulo une théorie *via* un *SMT solver*, R^**-trees*...)

Bibliographie

Granvilliers, L. and Benhamou, F. (2006).

Algorithm 852 : RealPaver : an Interval Solver Using Constraint Satisfaction Techniques.

ACM Trans. Math. Softw., 32(1) :138–156.

Huang, J. and Darwiche, A. (2005).

DPLL with a Trace : From SAT to Knowledge Compilation.

In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 156–162.

Vempaty, N. R. (1992).

Solving Constraint Satisfaction Problems Using Finite State Automata.

In *Association for the Avancement of Artificial Intelligence Conference (AAAI)*, pages 453–458.

Venturini, A. and Provan, G. (2008).

Incremental Algorithms for Approximate Compilation.

In *Association for the Avancement of Artificial Intelligence Conference (AAAI)*, pages 1495–1499.