Vérification de formules de logique temporelle à temps arborescent

Alban Linard

mai 2008

Menu du jour

1. Logique CTL
   - Opérateurs
   - Quantificateurs
2. Vérification sur le graphe d'accessibilité
3. Vérification avec des BDDs
   - Traduction des opérateurs
   - Traduction de la récursion

Computational Tree Logic

A partir d'un état, on s'intéresse à tous les états qui peuvent suivre

- Indéterminisme
- Temps arborescent

Opérateurs

p : propriété atomique, p est vérifiée

$f \wedge g$, $f \vee g$, $\neg f$, ... : opérateurs logiques

$G\ f$: tout au long du chemin, f est vérifiée

$F\ f$: au moins une fois dans le chemin, f est vérifiée

$X\ f$: l'état successeur vérifie f

$f\ U\ g$: f est vérifiée jusqu'à ce que g soit vérifiée (g finira par être vérifiée)

$f\ W\ g$: f est vérifiée jusqu'à ce que g soit vérifiée (g ne sera peut-être jamais vérifiée)

Remarques :

- Les opérateurs manipulent un chemin
- Où est le temps arborescent ?

Propriété atomique

Sémantique : $c \models f \iff e_0 \models f$

Explication : un chemin vérifie la propriété f ssi son premier état vérifie f

Globally

Sémantique : $c \models Gf \iff \forall i, c_i \models f$

Explication : la propriété f est vérifiée tout au long du chemin c ssi tout sous-chemin de c vérifie f

Finally

Sémantique : $c \models Ff \iff \exists i, c_i \models f$

Explication : la propriété f est vérifiée au moins une fois le long du chemin c ssi l'un des sous-chemins de c vérifie f

Next

Sémantique : $c \models Xf \iff c_1 \models f$

Explication : la propriété f est vérifiée par l'état successeur le long de c

Until

Sémantique : $c \models fUg \iff \exists i, c_i \models g \wedge \forall j < i, c_j \models f$

Explication : à partir d'une certaine étape du chemin c, tous les sous-chemins vérifient g, et f est vérifiée par tous les sous-chemins le précédant

Weak until

Sémantique : $c \models fWg \iff fUg \vee Gf$

Explication : si g est vérifiée à partir d'une certaine étape du chemin, alors f a été vérifiée tout au long du chemin précédant

Quantificateurs

Pour passer à une logique arborescente, chaque opérateur doit être quantitifé par :

$\forall f$: pour tous les chemins, f est vérifiée

$\exists f$: il existe un chemin pour lequel f est vérifiée

On passe d'une vue chemin par chemin (LTL) à une vue arborescente.

Forall

Sémantique : $e \models Af \iff \forall c \in C(e) \models f$

Explication : dans tout chemin partant de l'état e, f est vérifiée

Exists

Sémantique : $e \models Ef \iff \exists c \in C(e) \models f$

Explication : dans au moins un chemin partant de l'état e, f est vérifiée

Exemples

Donner les formules CTL des propriétés :

P1 :

P2 :

P3 :

Vérification sur le graphe d'accessibilité

Algorithme très simple :

- étiquetage des propositions atomiques sur les états du graphe les vérifiant
- étiquetage du graphe avec les sous-formules CTL utilisant les propositions atomiques étiquetées
- itération des étiquetages jusqu'à obtention de la formule CTL complète
- si les états initiaux sont étiquetés par la formule, celle-ci est vérifiée

Propagation arrière

- Des états vérifiant les propriétés atomiques, on remonte vers les états initiaux
- Il faut donc calculer avant tout le graphe d'accessibilité

Propagation arrière

- Des états vérifiant les propriétés atomiques, on remonte vers les états initiaux
- Il faut donc calculer avant tout le graphe d'accessibilité
- Autre possibilité, calculer à partir de l'ensemble des états respectant les propriétés atomiques et regarder si l'ensemble des états initiaux sont étiquetés par la formule complète

f

$EX\ f$

$AX\ f$

$EF\ f$

$AF\ f$

$EG\ f$

$AG\ f$

$E(f\ U\ g)$

$A(f\ U\ g)$

Vérification avec des BDDs

On dispose de :

- la relation de transition :
  retourne les états successeurs d'un ensemble d'états
- la relation inverse :
  retourne les états prédecesseurs d'un ensemble d'états

En partant des états vérifiant les propriétés atomiques, on remonte vers les états initiaux vérifiant la formule CTL. Si ceux-ci sont les états initiaux du système, la formule est vérifiée.

Opérations disponibles pour les BDDs

$EX\ f$:

$AX\ f$:

$EF\ f$:

$AF\ f$:

$EG\ f$:

$AG\ f$:

$E(f\ U\ g)$:

$A(f\ U\ g)$:

Comment traduire en opérations $\wedge$, $\vee$, ... ?

L'ensemble des états de l'espace d'états s vérifiant f est ?

f

L'ensemble des états de l'espace d'états s vérifiant f est ?

- $s \wedge f$

$EX\ f$

Quels est l'ensemble des états dont au moins un successeur vérifie f ?

$EX\ f$

Quels est l'ensemble des états dont au moins un successeur vérifie f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est

$EX\ f$

Quels est l'ensemble des états dont au moins un successeur vérifie f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$

$EX\ f$

Quels est l'ensemble des états dont au moins un successeur vérifie f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$
- L'ensemble s des états prédecesseurs d'un ensemble s' d'états est

$EX\ f$

Quels est l'ensemble des états dont au moins un successeur vérifie f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$
- L'ensemble s des états prédecesseurs d'un ensemble s' d'états est $t^{-1}(s')$

$EX\ f$

Quels est l'ensemble des états dont au moins un successeur vérifie f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$
- L'ensemble s des états prédecesseurs d'un ensemble s' d'états est $t^{-1}(s')$

EXf se traduit en $t^{-1}(f)$

$AX\ f$

Quel est l'ensemble des états dont tous les successeurs vérifient f ?

$AX\ f$

Quel est l'ensemble des états dont tous les successeurs vérifient f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$

$AX\ f$

Quel est l'ensemble des états dont tous les successeurs vérifient f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$
- Comment exprimer que tous les successeurs doivent vérifier f ?

$AX\ f$

Quel est l'ensemble des états dont tous les successeurs vérifient f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$
- Comment exprimer que tous les successeurs doivent vérifier f ?
  - tous les successeurs vérifient f : $t(s) \wedge f = t(s)$

$AX\ f$

Quel est l'ensemble des états dont tous les successeurs vérifient f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$
- Comment exprimer que tous les successeurs doivent vérifier f ?
  - tous les successeurs vérifient f : $t(s) \wedge f = t(s)$

$AX\ f$

Quel est l'ensemble des états dont tous les successeurs vérifient f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$
- Comment exprimer que tous les successeurs doivent vérifier f ?
  - tous les successeurs vérifient f : $t(s) \wedge f = t(s)$
- Comment exprimer que tous les prédecesseurs ont tous leurs successeurs vérifiant f ?

$AX\ f$

Quel est l'ensemble des états dont tous les successeurs vérifient f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$
- Comment exprimer que tous les successeurs doivent vérifier f ?
  - tous les successeurs vérifient f : $t(s) \wedge f = t(s)$
- Comment exprimer que tous les prédecesseurs ont tous leurs successeurs vérifiant f ?
  - $t(t^{-1}(f)) \wedge f$ pas très joli...

$AX\ f$

Quel est l'ensemble des états dont tous les successeurs vérifient f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$
- Comment exprimer que tous les successeurs doivent vérifier f ?
  - tous les successeurs vérifient f : $t(s) \wedge f = t(s)$
- Comment exprimer que tous les prédecesseurs ont tous leurs successeurs vérifiant f ?
  - $t(t^{-1}(f)) \wedge f$ pas très joli...
  - en faisant la négation de la formule :

$AX\ f$

Quel est l'ensemble des états dont tous les successeurs vérifient f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$
- Comment exprimer que tous les successeurs doivent vérifier f ?
  - tous les successeurs vérifient f : $t(s) \wedge f = t(s)$
- Comment exprimer que tous les prédecesseurs ont tous leurs successeurs vérifiant f ?
  - $t(t^{-1}(f)) \wedge f$ pas très joli...
  - en faisant la négation de la formule : $\neg EX \neg f$

$AX\ f$

Quel est l'ensemble des états dont tous les successeurs vérifient f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$
- Comment exprimer que tous les successeurs doivent vérifier f ?
  - tous les successeurs vérifient f : $t(s) \wedge f = t(s)$
- Comment exprimer que tous les prédecesseurs ont tous leurs successeurs vérifiant f ?
  - $t(t^{-1}(f)) \wedge f$ pas très joli...
  - en faisant la négation de la formule : $\neg EX \neg f$
  - on sait traduire EX !

$AX\ f$

Quel est l'ensemble des états dont tous les successeurs vérifient f ?

- L'ensemble s' des états successeurs d'un ensemble s d'états est $t(s)$
- Comment exprimer que tous les successeurs doivent vérifier f ?
  - tous les successeurs vérifient f : $t(s) \wedge f = t(s)$
- Comment exprimer que tous les prédecesseurs ont tous leurs successeurs vérifiant f ?
  - $t(t^{-1}(f)) \wedge f$ pas très joli...
  - en faisant la négation de la formule : $\neg EX \neg f$
  - on sait traduire EX !

$AX\ f$ se traduit en $\neg EX \neg f$

$EF\ f$

Quel est l'ensemble des états tels

$EF\ f$

Quel est l'ensemble des états tels qu'au moins un chemin partant de chacun contient un état vérifiant f ?

$EF\ f$

Quel est l'ensemble des états tels qu'au moins un chemin partant de chacun contient un état vérifiant f ?

- Cette formule touche un chemin et non un simple successeur

$EF\ f$

Quel est l'ensemble des états tels qu'au moins un chemin partant de chacun contient un état vérifiant f ?

- Cette formule touche un chemin et non un simple successeur
- Exprimable de façon récursive

$EF\ f$

Quel est l'ensemble des états tels qu'au moins un chemin partant de chacun contient un état vérifiant f ?

- Cette formule touche un chemin et non un simple successeur
- Exprimable de façon récursive
- Les états vérifiant f ou dont au moins un successeur vérifie $EF\ f$:

$EF\ f$

Quel est l'ensemble des états tels qu'au moins un chemin partant de chacun contient un état vérifiant f ?

- Cette formule touche un chemin et non un simple successeur
- Exprimable de façon récursive
- Les états vérifiant f ou dont au moins un successeur vérifie $EF\ f : f \vee EX\ EF\ f$

$EF\ f$

Quel est l'ensemble des états tels qu'au moins un chemin partant de chacun contient un état vérifiant f ?

- Cette formule touche un chemin et non un simple successeur
- Exprimable de façon récursive
- Les états vérifiant f ou dont au moins un successeur vérifie $EF\ f : f \vee EX\ EF\ f$
- On sait traduire EX (mais pas encore la récursion)

$EF\ f$

Quel est l'ensemble des états tels qu'au moins un chemin partant de chacun contient un état vérifiant f ?

- Cette formule touche un chemin et non un simple successeur
- Exprimable de façon récursive
- Les états vérifiant f ou dont au moins un successeur vérifie $EF\ f$: $f \vee EX\ EF\ f$
- On sait traduire EX (mais pas encore la récursion)

$EF\ f$ se traduit en $f \vee EX\ EF\ f$

$AF\ f$

Quel est l'ensemble des états tels

$AF\ f$

Quel est l'ensemble des états tels que tous les chemins partant de chacun contiennent un état vérifiant f ?

$AF\ f$

Quel est l'ensemble des états tels que tous les chemins partant de chacun contiennent un état vérifiant f ?

- De même, exprimable de façon récursive

$AF\ f$

Quel est l'ensemble des états tels que tous les chemins partant de chacun contiennent un état vérifiant f ?

- De même, exprimable de façon récursive
- Les états vérifiant f ou dont tous les successeurs vérifient $AF\ f$:

$AF\ f$

Quel est l'ensemble des états tels que tous les chemins partant de chacun contiennent un état vérifiant f ?

- De même, exprimable de façon récursive
- Les états vérifiant f ou dont tous les successeurs vérifient $AF\ f$: $f \vee AX\ AF\ f$

$AF\ f$

Quel est l'ensemble des états tels que tous les chemins partant de chacun contiennent un état vérifiant f ?

- De même, exprimable de façon récursive
- Les états vérifiant f ou dont tous les successeurs vérifient $AF\ f$: $f \vee AX\ AF\ f$
- On sait traduire AX (mais pas encore la récursion)

$AF\ f$

Quel est l'ensemble des états tels que tous les chemins partant de chacun contiennent un état vérifiant f ?

- De même, exprimable de façon récursive
- Les états vérifiant f ou dont tous les successeurs vérifient $AF\ f$: $f \vee AX\ AF\ f$
- On sait traduire AX (mais pas encore la récursion)

$AF\ f$ se traduit en $f \vee AX\ AF\ f$

$EG\ f$

Quel est l'ensemble des états tels

$EG\ f$

Quel est l'ensemble des états tels qu'un moins un chemin partant de chacun vérifie toujours f ?

$EG\ f$

Quel est l'ensemble des états tels qu'un moins un chemin partant de chacun vérifie toujours f ?

- Attention, la construction récursive ne fonctionne pas de la même manière

$EG\ f$

Quel est l'ensemble des états tels qu'un moins un chemin partant de chacun vérifie toujours f ?

- Attention, la construction récursive ne fonctionne pas de la même manière
- On enlève à chaque itération les états ne vérifiant pas f alors que le chemin partant d'eux vérifiait f

$EG\ f$

Quel est l'ensemble des états tels qu'un moins un chemin partant de chacun vérifie toujours f ?

- Attention, la construction récursive ne fonctionne pas de la même manière
- On enlève à chaque itération les états ne vérifiant pas f alors que le chemin partant d'eux vérifiait f
- $f \wedge EX\ EG\ f$

$EG\ f$

Quel est l'ensemble des états tels qu'un moins un chemin partant de chacun vérifie toujours f ?

- Attention, la construction récursive ne fonctionne pas de la même manière
- On enlève à chaque itération les états ne vérifiant pas f alors que le chemin partant d'eux vérifiait f
- $f \wedge EX\ EG\ f$
- On sait traduire EX (mais pas encore la récursion)

$EG\ f$

Quel est l'ensemble des états tels qu'un moins un chemin partant de chacun vérifie toujours f ?

- Attention, la construction récursive ne fonctionne pas de la même manière
- On enlève à chaque itération les états ne vérifiant pas f alors que le chemin partant d'eux vérifiait f
- $f \wedge EX\ EG\ f$
- On sait traduire EX (mais pas encore la récursion)

$EG\ f$ se traduit en $f \wedge EX\ EG\ f$

$AG\ f$

Quel est l'ensemble des états tels

$AG\ f$

Quel est l'ensemble des états tels que tous les chemins partant de chacun d'eux vérifie toujours f ?

$AG\ f$

Quel est l'ensemble des états tels que tous les chemins partant de chacun d'eux vérifie toujours f ?

- $f \wedge AX\ AG\ f$

$AG\ f$

Quel est l'ensemble des états tels que tous les chemins partant de chacun d'eux vérifie toujours f ?

- $f \wedge AX\ AG\ f$
- On sait traduire AX (mais pas encore la récursion)

$AG\ f$

Quel est l'ensemble des états tels que tous les chemins partant de chacun d'eux vérifie toujours f ?

- $f \wedge AX\ AG\ f$
- On sait traduire AX (mais pas encore la récursion)

$AG\ f$ se traduit en $f \wedge AX\ AG\ f$

$E(f\ U\ g)$

Quel est l'ensemble des états tels

$E(f\ U\ g)$

Quel est l'ensemble des états tels qu'au moins un chemin vérifie f jusqu'à vérifier g ?

$E(f\ U\ g)$

Quel est l'ensemble des états tels qu'au moins un chemin vérifie f jusqu'à vérifier g ?

- Toujours traduire de manière récursive...

$E(f\ U\ g)$

Quel est l'ensemble des états tels qu'au moins un chemin vérifie f jusqu'à vérifier g ?

- Toujours traduire de manière récursive...
- Ces états vérifient g ou vérifient f et sont à l'origine d'au moins un chemin vérifiant $E(f\ U\ g)$

$E(f\ U\ g)$

Quel est l'ensemble des états tels qu'au moins un chemin vérifie f jusqu'à vérifier g ?

- Toujours traduire de manière récursive...
- Ces états vérifient g ou vérifient f et sont à l'origine d'au moins un chemin vérifiant $E(f\ U\ g)$
- $g \vee (f \wedge EX\ E(f\ U\ g))$

$E(f\ U\ g)$

Quel est l'ensemble des états tels qu'au moins un chemin vérifie f jusqu'à vérifier g ?

- Toujours traduire de manière récursive...
- Ces états vérifient g ou vérifient f et sont à l'origine d'au moins un chemin vérifiant $E(f\ U\ g)$
- $g \vee (f \wedge EX\ E(f\ U\ g))$

$E(f\ U\ g)$ se traduit en $g \vee (f \wedge EX\ E(f\ U\ g))$

$A(f\ U\ g)$

Quel est l'ensemble des états tels

$A(f\ U\ g)$

Quel est l'ensemble des états tels que tous les chemins vérifient f jusqu'à vérifier g ?

$A(f\ U\ g)$

Quel est l'ensemble des états tels que tous les chemins vérifient f jusqu'à vérifier g ?

- Toujours traduire de manière récursive...

$A(f\ U\ g)$

Quel est l'ensemble des états tels que tous les chemins vérifient f jusqu'à vérifier g ?

- Toujours traduire de manière récursive...
- Ces états vérifient g ou vérifient f et tous les chemins en partant vérifiant $A(f\ U\ g)$

$A(f\ U\ g)$

Quel est l'ensemble des états tels que tous les chemins vérifient f jusqu'à vérifier g ?

- Toujours traduire de manière récursive...
- Ces états vérifient g ou vérifient f et tous les chemins en partant vérifiant $A(f\ U\ g)$
- $g \vee (f \wedge AX\ A(f\ U\ g))$

$A(f\ U\ g)$

Quel est l'ensemble des états tels que tous les chemins vérifient f jusqu'à vérifier g ?

- Toujours traduire de manière récursive...
- Ces états vérifient g ou vérifient f et tous les chemins en partant vérifiant $A(f\ U\ g)$
- $g \vee (f \wedge AX\ A(f\ U\ g))$

$A(f\ U\ g)$ se traduit en $g \vee (f \wedge AX\ A(f\ U\ g))$

Traduction de la récursion

Rappel :

- $EF\ f$ se traduit en $f \vee EX\ EF\ f$
- $AF\ f$ se traduit en $f \vee AX\ AF\ f$
- $EG\ f$ se traduit en $f \wedge EX\ EG\ f$
- $AG\ f$ se traduit en $f \wedge AX\ AG\ f$
- $E(f\ U\ g)$ se traduit en $g \vee (f \wedge EX\ E(f\ U\ g))$
- $A(f\ U\ g)$ se traduit en $g \vee (f \wedge AX\ A(f\ U\ g))$

- Différence entre $\vee$ et $\wedge$: construction incrémentale ou décrémentale
- Fin de la récursion : lorsqu'un point fixe est atteint

Point fixe incrémental

- τ fonction monotone $S \longrightarrow S$
- monotone : $s \subseteq s' \implies \tau(s) \subseteq \tau(s')$
- point fixe : $\mu y.\tau(y) = lim(\tau(\ldots(\tau(false))\ldots))$
- incrémental : $s \subseteq \tau(s)$
- *false* est l'ensemble vide, la formule 0
- A chaque itération, τ ajoute des éléments dans l'ensemble

Point fixe décrémental

- τ fonction monotone $S \rightarrow S$
- monotone : $s \subseteq s' \implies \tau(s) \subseteq \tau(s')$
- point fixe : $\nu y.\tau(y) = lim(\tau(\ldots(\tau(true))\ldots))$
- décrémental : $\tau(s) \subseteq s$
- *true* est la formule 1
- A chaque itération, τ supprime des éléments de l'ensemble

Traduction des formules CTL

Traduction des formules CTL

$EF\ f\ :\ \mu y.(f \vee EX\ y)$

Traduction des formules CTL

$$EF\ f : \mu y.(f \vee EX\ y)$$
$$AF\ f : \mu y.(f \vee AX\ y)$$

Traduction des formules CTL

$$EF\ f : \mu y.(f \vee EX\ y)$$
$$AF\ f : \mu y.(f \vee AX\ y)$$
$$EG\ f : \nu y.(f \wedge EX\ y)$$

Traduction des formules CTL

$$EF\ f : \mu y.(f \vee EX\ y)$$
$$AF\ f : \mu y.(f \vee AX\ y)$$
$$EG\ f : \nu y.(f \wedge EX\ y)$$
$$AG\ f : \nu y.(f \wedge AX\ y)$$

Traduction des formules CTL

$$EF\ f : \mu y.(f \vee EX\ y)$$
$$AF\ f : \mu y.(f \vee AX\ y)$$
$$EG\ f : \nu y.(f \wedge EX\ y)$$
$$AG\ f : \nu y.(f \wedge AX\ y)$$
$$E(f\ U\ g) : \mu y.(g \vee (f \wedge EX\ y))$$

Traduction des formules CTL

$$EF\ f : \mu y.(f \vee EX\ y)$$
$$AF\ f : \mu y.(f \vee AX\ y)$$
$$EG\ f : \nu y.(f \wedge EX\ y)$$
$$AG\ f : \nu y.(f \wedge AX\ y)$$
$$E(f\ U\ g) : \mu y.(g \vee (f \wedge EX\ y))$$
$$A(f\ U\ g) : \mu y.(g \vee (f \wedge AX\ y))$$

Conclusion

Pour faire de la vérification de formules CTL en BDD :

- représenter les transitions,
- calculer l'espace d'état,
- calculer le BDD représentant la formule CTL
- vérifier que le résultat obtenu est bien l'ensemble des états initiaux

Sources

- Présentation "Introduction to Model Checking" de Ken McMillan
  MOVEP 1998
- Cours de Vérification Formelle de E. Encrenaz-Tiphène et I. Vernier-Mounier