DIC9305 Logique, informatique et sciences cognitives

Logique du premier-ordre

Roger Villemaire

Département d'informatique
UQAM

le 4 avril 2024

UQÀM
Université du Québec à Montréal

Plan

Plan

Définition

- La *logique du premier-ordre* étend le calcul propositionnel en considérant en plus des connecteurs booléens :
  - des *variables* $x, y, z, u, v, x_1, y_1, z_1, u_1, v_1, \ldots,$
  - des *constantes* $a, b, c, a_1, b_1, c_1, \ldots,$
  - des *symboles de relations* $R(x), S(x, y) \ldots,$
  - ainsi que les *quantificateurs*
    - *universel* $\forall x$ (“pour tout x”),
    - *existentiel* $\exists x$ (“il existe un x”).

Exemple de modélisation

La *distance sociale* peut être représentée par la relation $R(x, y)$: "x connaît y".

- $R(jean, marie)$, "*jean* connaît *marie*".
- $\forall x \exists y R(x, y)$, "on connaît tous quelqu'un".
- $\forall z(R(x, z) \rightarrow R(y, z))$, "$y$ connaît tous les gens que x connaît".
- $\forall y R(x, y)$, "x connaît tout le monde".

Plan

Sémantique

- Une *structure* $\mathcal{M}$ est un ensemble M, l'*univers*, muni de relations $R^{\mathcal{M}}$, une pour chaque relation R du langage. On *interprète* le sens d'un formule du premier-ordre sur $\mathcal{M}$, où $\bar{m}$ est un *tuplet* d'éléments de M de la bonne longueur, de la façon suivante :
  - $\mathcal{M} \models P(\bar{m})$, pour une relation P, si $P^{\mathcal{M}}(\bar{m})$.
  - $\mathcal{M} \models \neg\varphi$, si $\mathcal{M} \not\models \varphi$.
  - $\mathcal{M} \models \varphi \wedge \psi$, si $\mathcal{M} \models \varphi$ et $\mathcal{M} \models \psi$.
  - $\mathcal{M} \models \varphi \vee \psi$, si $\mathcal{M} \models \varphi$ ou $\mathcal{M} \models \psi$.
  - $\mathcal{M} \models \varphi \rightarrow \psi$, si $\mathcal{M} \models \psi$ lorsque $\mathcal{M} \models \varphi$.
  - $\mathcal{M} \models \exists x\varphi(x)$, s'il existe un $m \in M$, tel que $\mathcal{M} \models \varphi(m)$.
  - $\mathcal{M} \models \forall x\varphi(x)$, si pour tout $m \in M$, on a que $\mathcal{M} \models \varphi(m)$.

Universelle

- La logique du premier-ordre est souvent considérée comme étant *la logique universelle*.
- Ainsi, normalement, les autres logiques classiques peuvent *s'interpréter* en logique du premier-ordre.
- Ceci permet d'étendre les techniques et méthodes développées pour la logique du premier-ordre à toutes ces logiques
- et aussi de considérer celles-ci comme des fragments de la logique du premier-ordre.

Plan

Interprétation en logique du premier ordre : concepts

- Un concept C correspond à une relation unaire $C(x)$,
  - $\top$ correspond à $x = x$,
  - $\bot$ correspond à $x \neq x$,
  - $\sqcap$ correspond à la *conjonction* $\wedge$,
  - $\sqcup$ correspond à la *disjonction* $\vee$,
  - $\neg$ correspond à la *négation* $\neg$,

Interprétation en logique du premier ordre : rôles

- Un rôle r correspond à une relation binaire $r(x, y)$,
  - $\forall r.C$ correspond à $\forall y.(r(x, y) \rightarrow C(y))$,
  - $\exists r.C$ correspond à $\exists y.r(x, y) \wedge C(y)$,
  - $\geq n.\ r$ correspond à “il existe au moins n, y différents”,
  - $\leq n.\ r$ correspond à “il n’existe pas plus de n, y différents”,
  - $r \mid C$ correspond à $r(x, y) \wedge C(y)$.

Structures pour la logique de description

- Ceci correspond complètement à la sémantique que l'on avait vue pour la logique de description.
  - Une $\mathcal{L}$-structure $\mathcal{U}$ est formée
    - d'un ensemble U (d'éléments),
    - pour tout concept C de $\mathcal{L}$ d'une interprétation $C^{\mathcal{U}}$ qui est un sous-ensemble de U,
    - et pour tout rôle r de $\mathcal{L}$ d'une interprétation $r^{\mathcal{U}}$ qui est un sous-ensemble de couples (ordonnés) de U.

Exemples

- $\mathit{Etudiant} \sqcap \mathit{Prof}$ correspond à
  - $\mathit{Etudiant}(x) \wedge \mathit{Prof}(x)$,
- $\mathit{Etudiant} \sqcap \mathit{Prof} \sqsubseteq \bot$ correspond à
  - $\forall x(\mathit{Etudiant}(x) \wedge \mathit{Prof}(x) \rightarrow x \neq x)$,
- $\exists \mathit{suit}.\mathit{Cours}$ correspond à
  - $\exists y.\mathit{suit}(x, y) \wedge \mathit{Cours}(y)$,
- $\exists \mathit{suit}.\mathit{Cours} \sqsubseteq \mathit{Etudiant}$ correspond à
  - $\forall x(\exists y.\mathit{suit}(x, y) \wedge \mathit{Cours}(y) \rightarrow \mathit{Etudiant}(x))$,
- $\mathit{Etudiant} \sqsubseteq \forall \mathit{suit}.\mathit{Cours}$ correspond à
  - $\forall x(\mathit{Etudiant}(x) \rightarrow \forall y.(\mathit{suit}(x, y) \rightarrow \mathit{Cours}(y)))$.

Comparaison

- La logique de description restreint donc la logique du premier ordre en
  - ne permettant que des relations unaires et binaires,
  - et que des quantificateurs *gardés* $\forall y.(r(x,y) \rightarrow C(y))$ et $\exists y.r(x,y) \wedge C(y)$.
- Il s'agit donc d'un *fragment* de la logique du premier ordre.

Plan

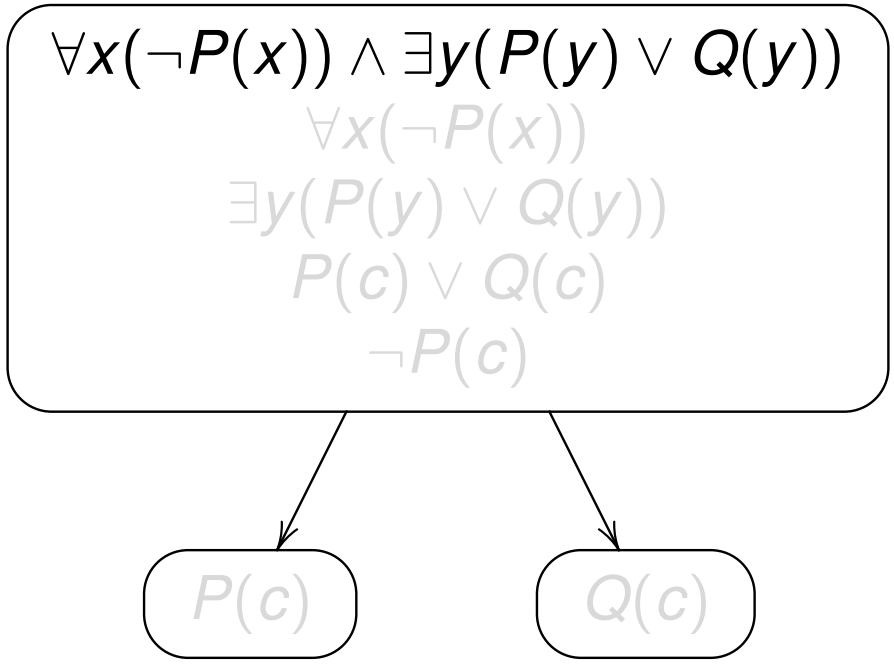
La forme normale négative

- Une formule est en *forme normale négative* (FNN) si la négation n'est utilisée que sur des symboles de relations.
- On peut transformer une formule en FNN en "poussant" les négations vers l'intérieur en utilisant les règles de de Morgan :
  - $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$.
  - $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$.
- et la dualité entre $\exists$ et $\forall$:
  - $\neg(\exists x\varphi) \equiv \forall x\neg\varphi$.
  - $\neg(\forall x\varphi) \equiv \exists x\neg\varphi$.

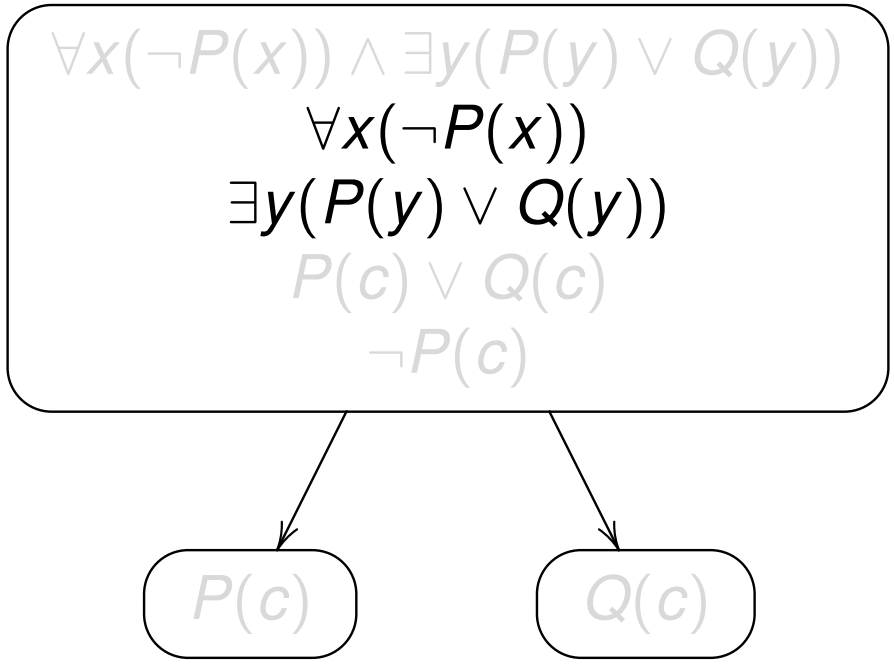
La méthode des tableaux

- On construit un arbre en débutant avec une racine contenant une formule (en FNN) à satisfaire et pour tout noeud $\mathcal{N}$ on applique :
  - Si $\varphi \wedge \psi \in \mathcal{N}$ alors on ajoute φ et ψ à $\mathcal{N}$.
  - Si $\varphi \vee \psi \in \mathcal{N}$ alors on crée deux descendants $\mathcal{N}_1$ et $\mathcal{N}_2$ de $\mathcal{N}$, contenant respectivement φ et ψ,
  - Si $\exists x \varphi(x)$ alors on ajoute $\varphi(c)$ pour c une constante n'apparaissant pas sur cette branche,
  - Si $\forall x \varphi(x)$ alors on ajoute $\varphi(c)$ pour toutes les constantes c présentes sur cette branche (il doit y avoir au moins une constante sur la branche, sinon en rajouter une).

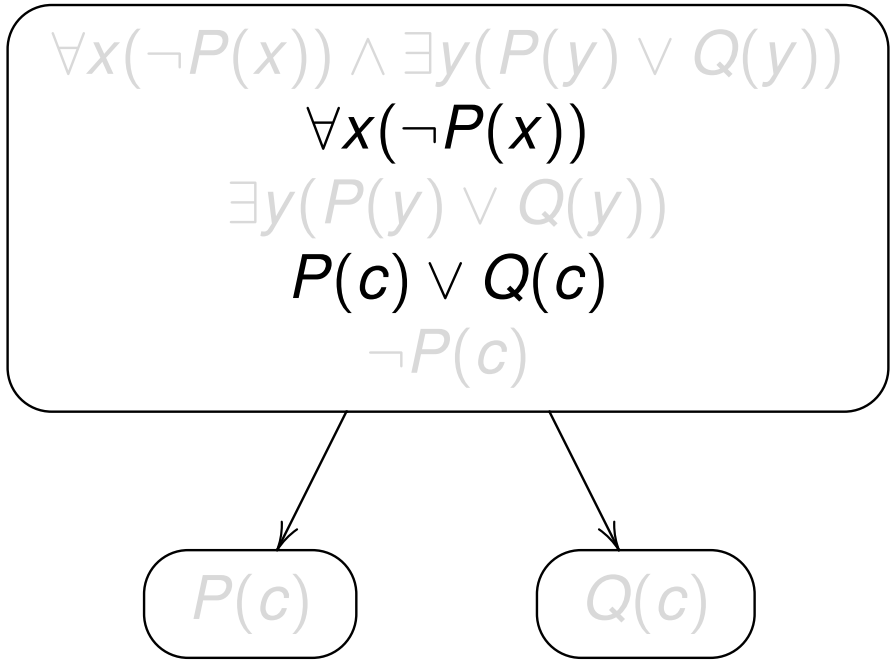
Exemple


$\forall x(\neg P(x)) \wedge \exists y(P(y) \vee Q(y))$
$\forall x(\neg P(x))$
$\exists y(P(y) \vee Q(y))$
$P(c) \vee Q(c)$
$\neg P(c)$
$P(c)$
$Q(c)$

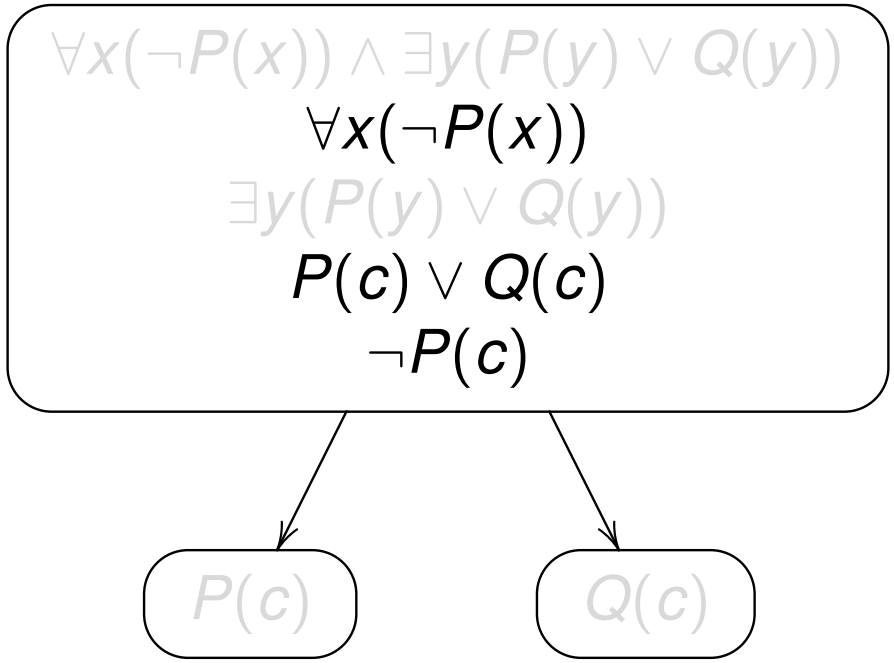
Exemple


$\forall x(\neg P(x)) \wedge \exists y(P(y) \vee Q(y))$
$\forall x(\neg P(x))$
$\exists y(P(y) \vee Q(y))$
$P(c) \vee Q(c)$
$\neg P(c)$
$P(c)$
$Q(c)$

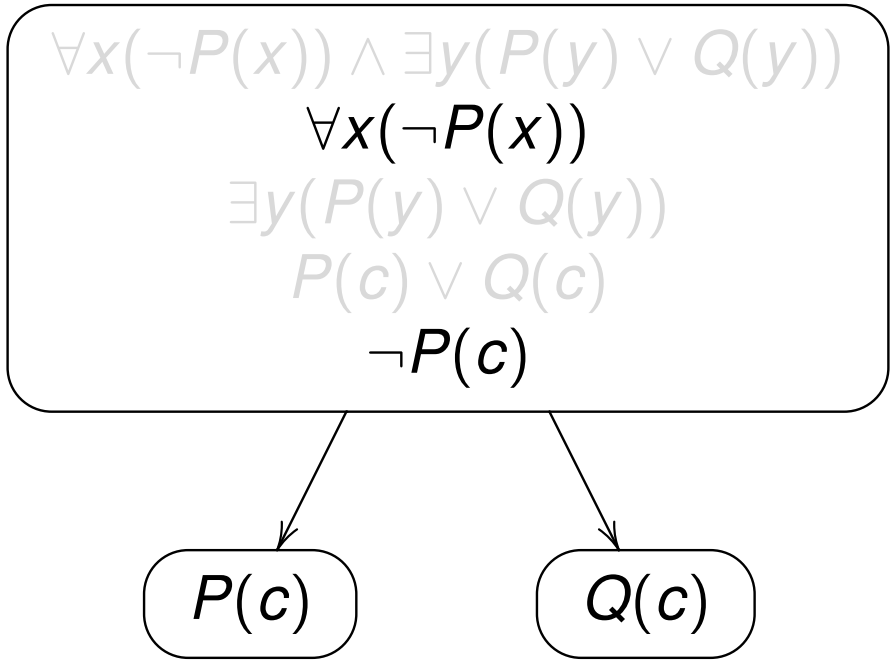
Exemple


$\forall x(\neg P(x)) \wedge \exists y(P(y) \vee Q(y))$
$\forall x(\neg P(x))$
$\exists y(P(y) \vee Q(y))$
$P(c) \vee Q(c)$
$\neg P(c)$
$P(c)$
$Q(c)$

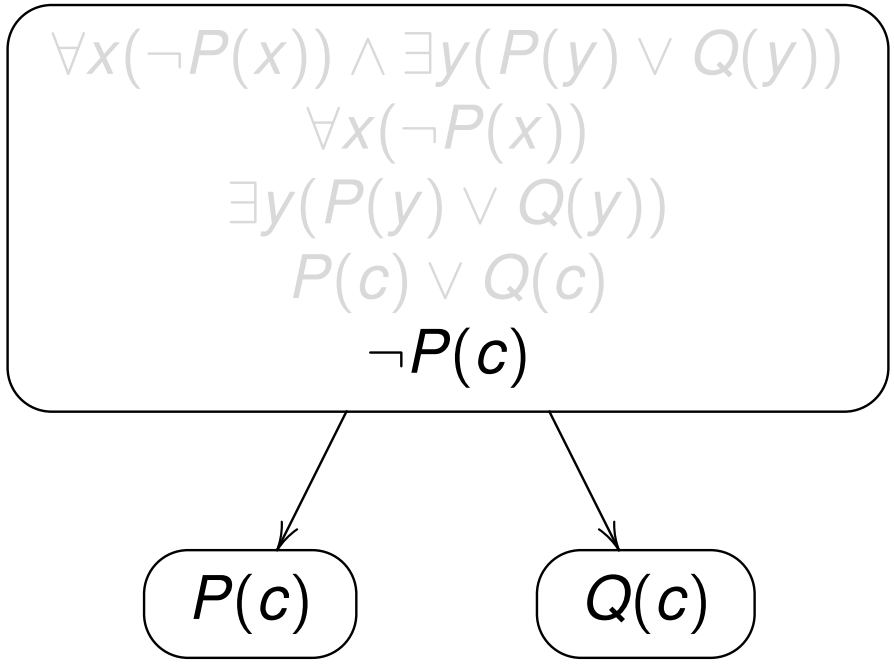
Exemple


$\forall x(\neg P(x)) \wedge \exists y(P(y) \vee Q(y))$
$\forall x(\neg P(x))$
$\exists y(P(y) \vee Q(y))$
$P(c) \vee Q(c)$
$\neg P(c)$
$P(c)$
$Q(c)$

Exemple


$\forall x(\neg P(x)) \wedge \exists y(P(y) \vee Q(y))$
$\forall x(\neg P(x))$
$\exists y(P(y) \vee Q(y))$
$P(c) \vee Q(c)$
$\neg P(c)$
$P(c)$
$Q(c)$

Exemple

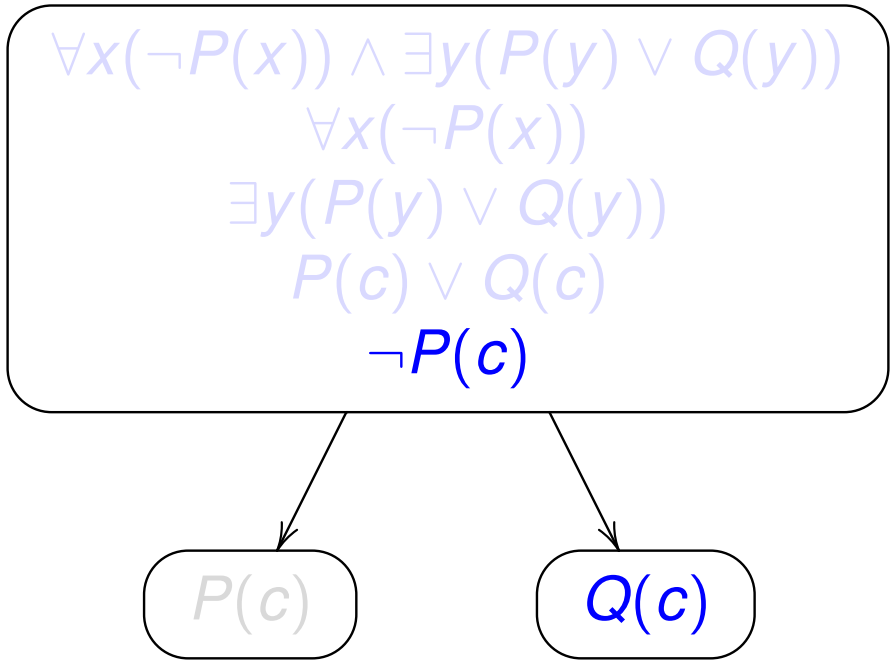

$\forall x(\neg P(x)) \wedge \exists y(P(y) \vee Q(y))$
$\forall x(\neg P(x))$
$\exists y(P(y) \vee Q(y))$
$P(c) \vee Q(c)$
$\neg P(c)$
$P(c)$
$Q(c)$

Définitions

- On dit qu'un tableau est *complet* si toutes les formules ont été traitées par les règles ci-dessus.
- On dit qu'une branche d'un tableau est *close* si elle contient P et $\neg P$, pour un certain P.
- On dit qu'un tableau est *clos* si toutes ses branches sont closes.

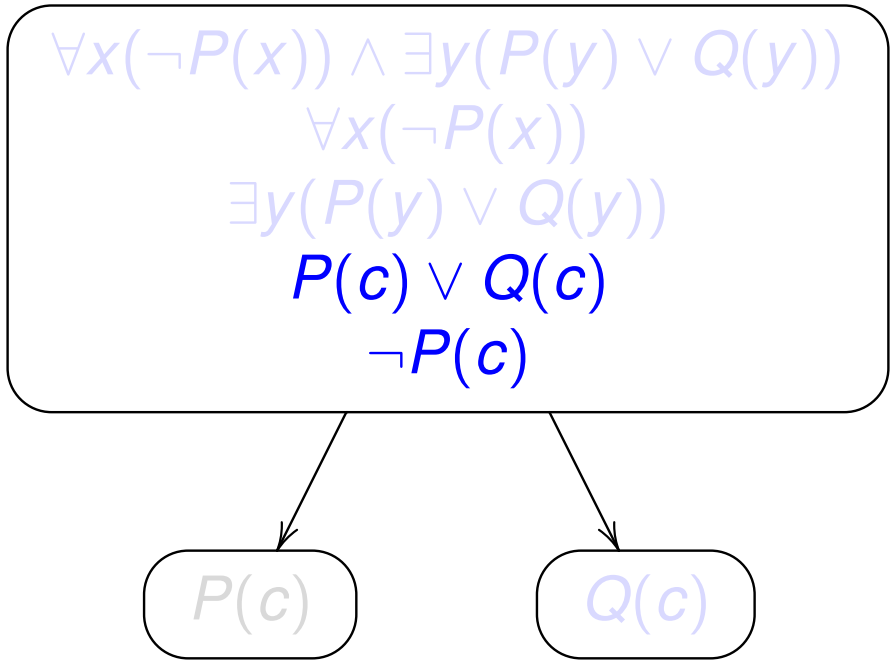
Modèle

Une branche ouverte d'un tableau permet de définir une structure satisfaisant la formule de départ.



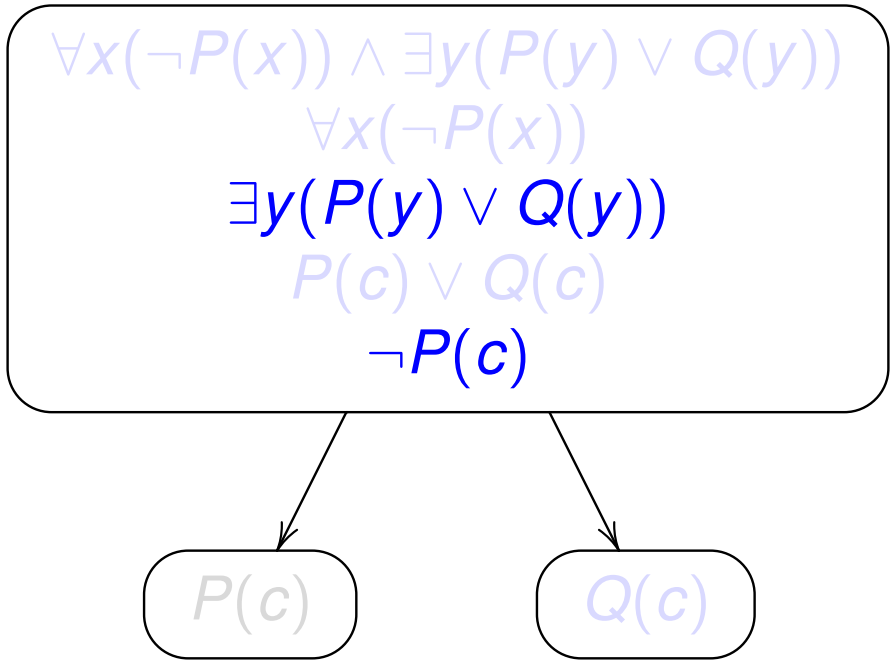
Modèle

Une branche ouverte d'un tableau permet de définir une structure satisfaisant la formule de départ.



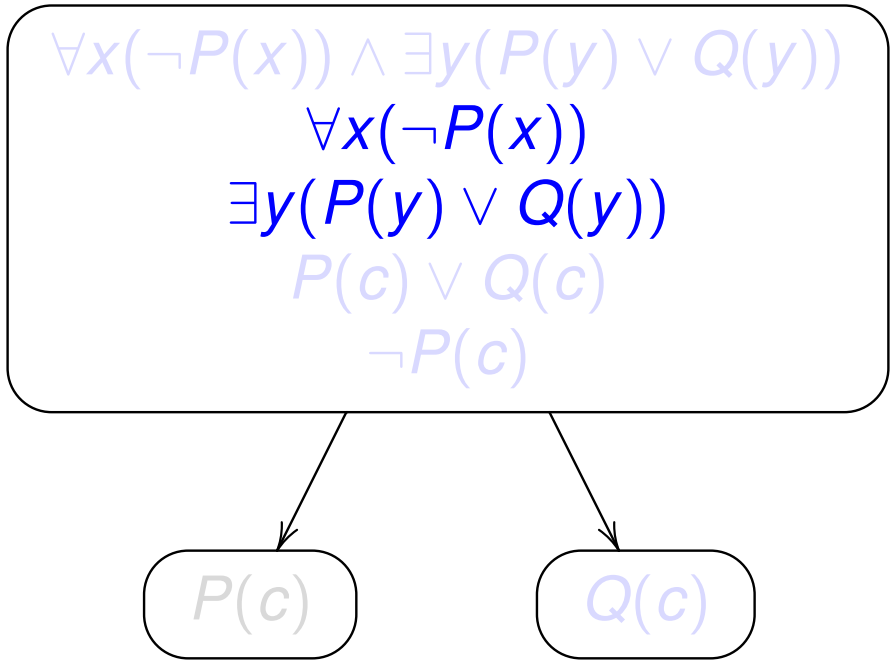
Modèle

Une branche ouverte d'un tableau permet de définir une structure satisfaisant la formule de départ.



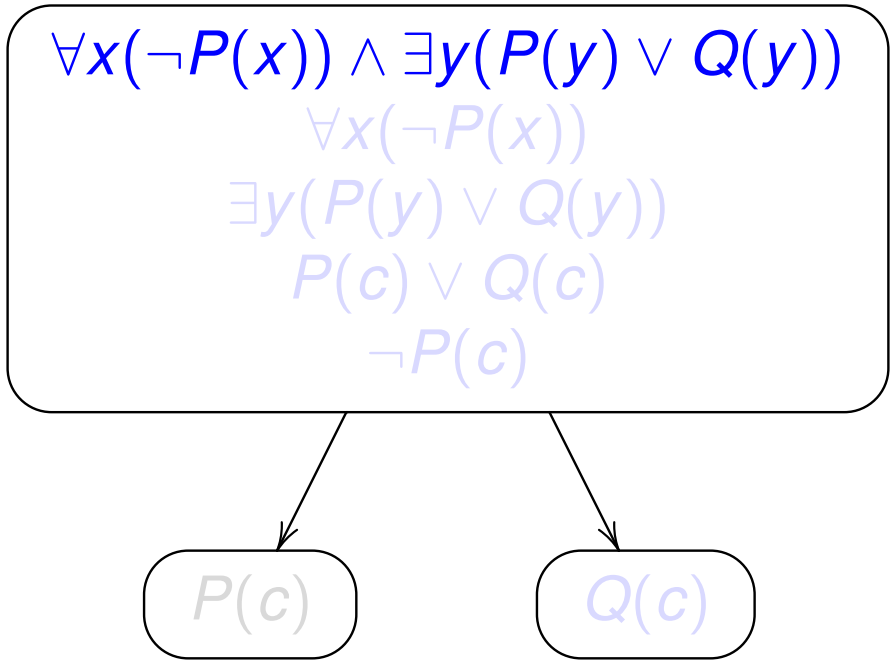
Modèle

Une branche ouverte d'un tableau permet de définir une structure satisfaisant la formule de départ.



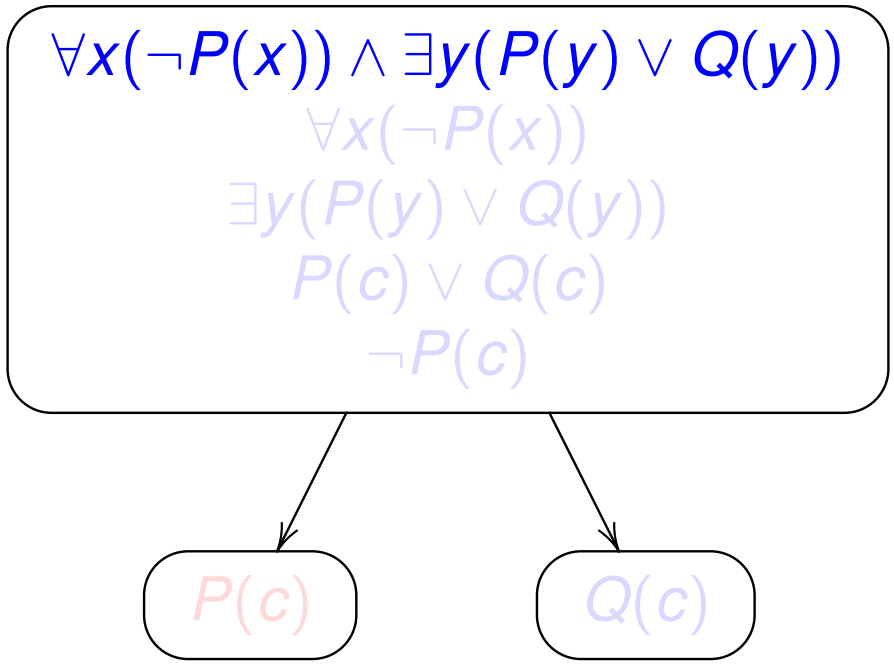
Modèle

Une branche ouverte d'un tableau permet de définir une structure satisfaisant la formule de départ.



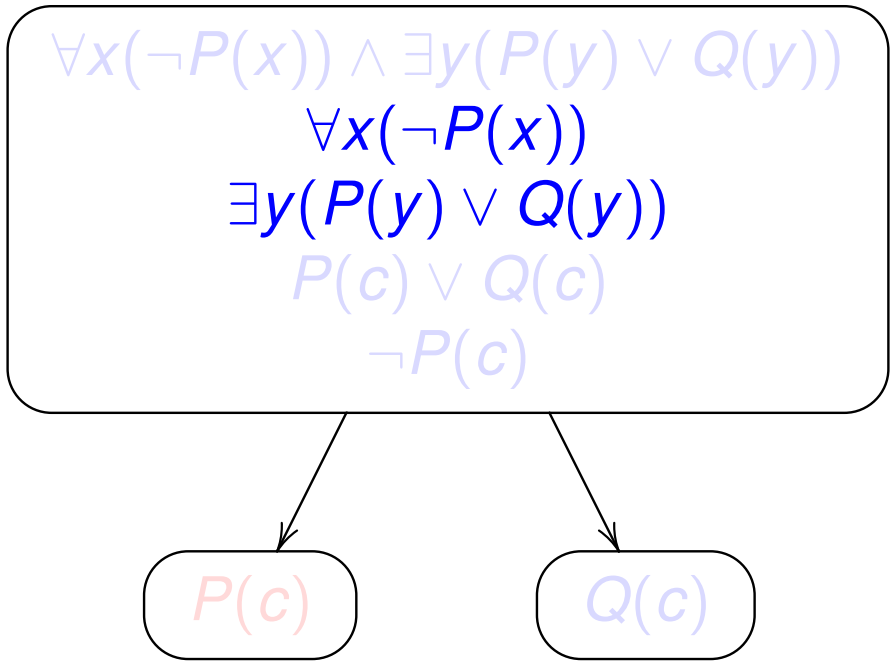
Branche

Une structure satisfaisant une formule (un *modèle*) doit satisfaire toutes les formules d'au moins une branche d'un tableau (pour cette formule).
Considérons $M = \{c\}$ avec $\neg P(c)$ et $Q(c)$.



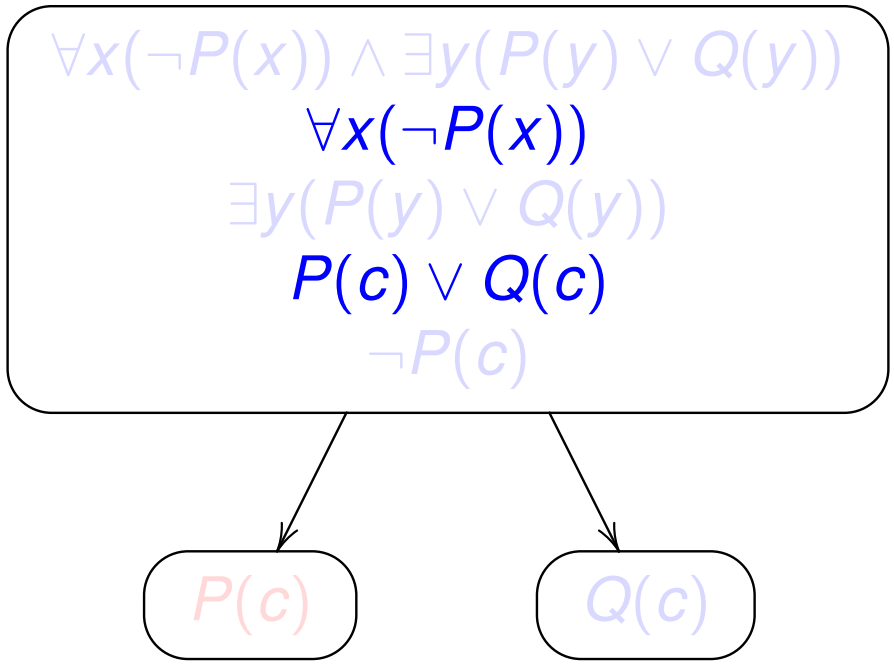
Branche

Une structure satisfaisant une formule (un *modèle*) doit satisfaire toutes les formules d'au moins une branche d'un tableau (pour cette formule).
Considérons $M = \{c\}$ avec $\neg P(c)$ et $Q(c)$.



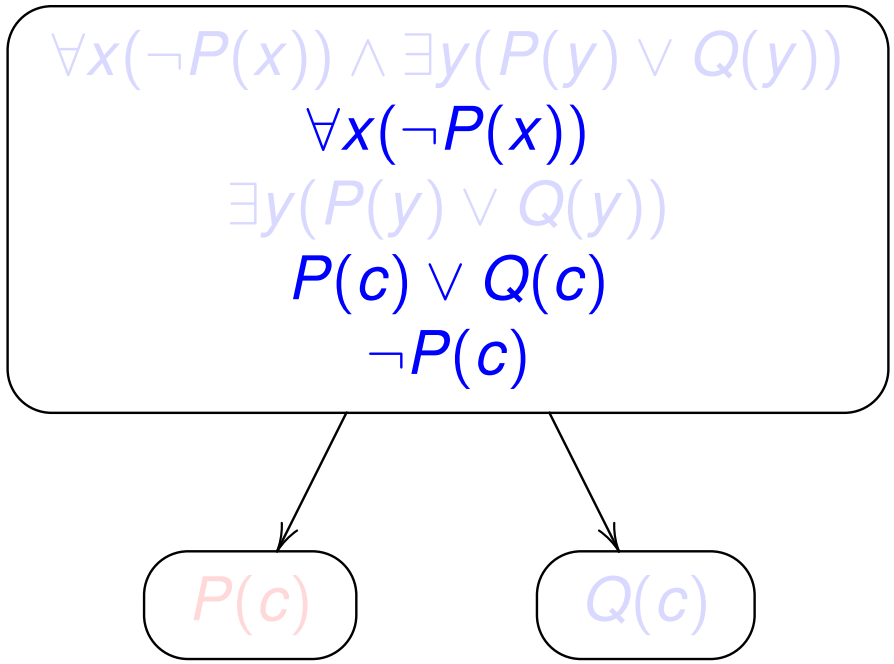
Branche

Une structure satisfaisant une formule (un *modèle*) doit satisfaire toutes les formules d'au moins une branche d'un tableau (pour cette formule).
Considérons $M = \{c\}$ avec $\neg P(c)$ et $Q(c)$.



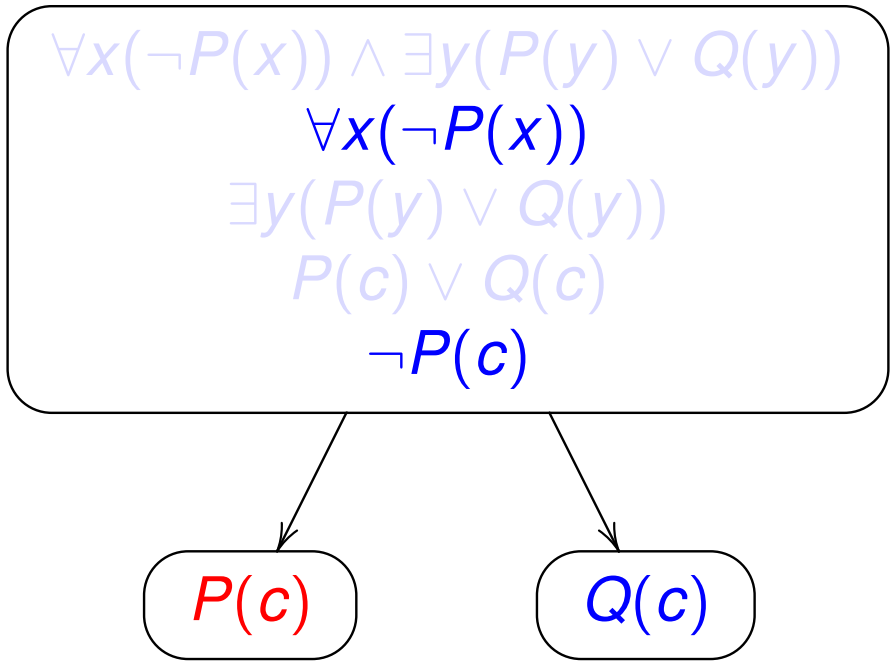
Branche

Une structure satisfaisant une formule (un *modèle*) doit satisfaire toutes les formules d'au moins une branche d'un tableau (pour cette formule).
Considérons $M = \{c\}$ avec $\neg P(c)$ et $Q(c)$.



Branche

Une structure satisfaisant une formule (un *modèle*) doit satisfaire toutes les formules d'au moins une branche d'un tableau (pour cette formule).
Considérons $M = \{c\}$ avec $\neg P(c)$ et $Q(c)$.



Réfutation et Démonstration

- On dit qu'un formule φ est *réfutable* si tout tableau complet pour φ est clos.
- On dit qu'un formule φ est *démontrable*, noté $\vdash \varphi$ si $\neg\varphi$ est réfutable.

Complétude

Théorème

φ est insatisfaisable si et seulement si φ est réfutable.

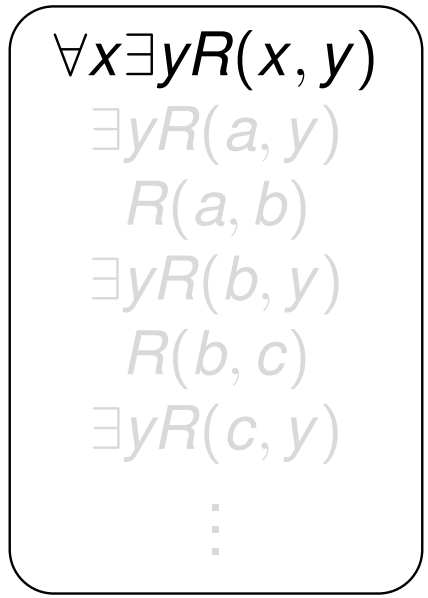
Théorème

$\models \varphi$ si et seulement si $\vdash \varphi$.

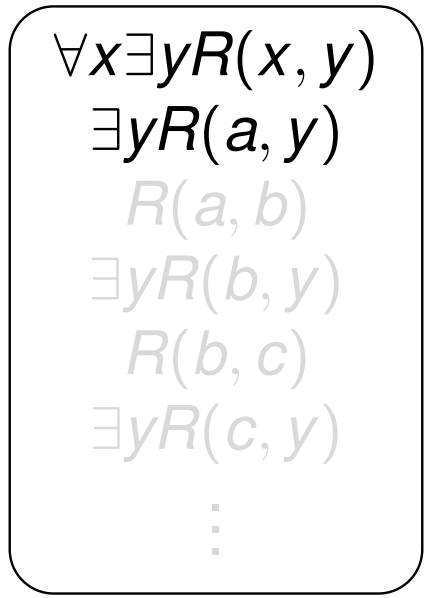
Démonstration et contre-exemple

- Si une formule est démontrable, toutes les branches du tableau pour sa négation seront closes. On aura donc un arbre fini, la réfutation de la négation de la formule, qui est une évidence finie du fait que la formule est démontrable et donc valide.
- Si une formule n'est pas démontrable, une branche au moins du tableau pour sa négation est ouverte. Cette branche est une évidence que la formule n'est pas démontrable et donc n'est pas une tautologie. Mais cette branche peut être infinie !

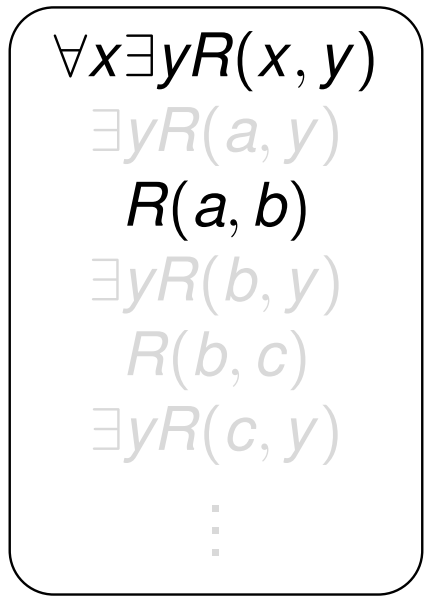
Exemple de branche infinie



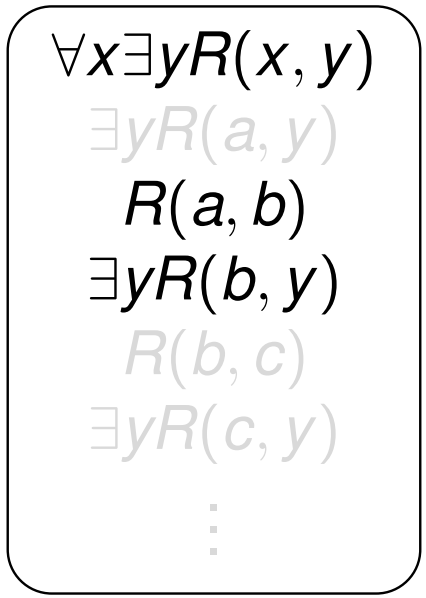
Exemple de branche infinie



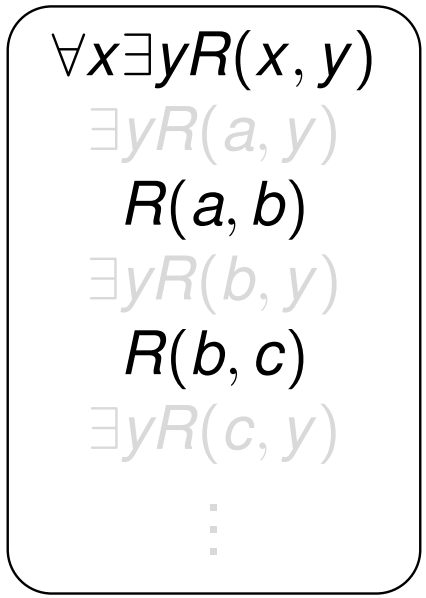
Exemple de branche infinie



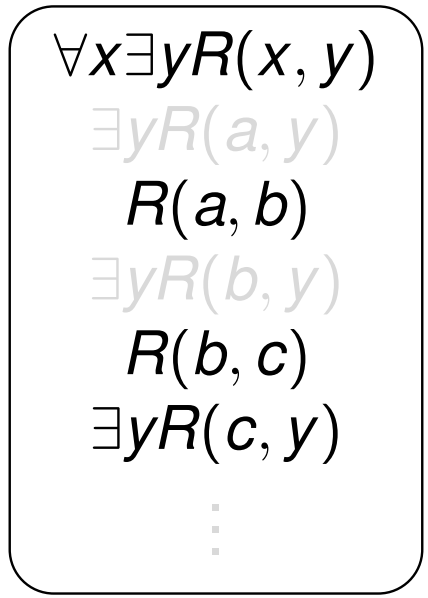
Exemple de branche infinie



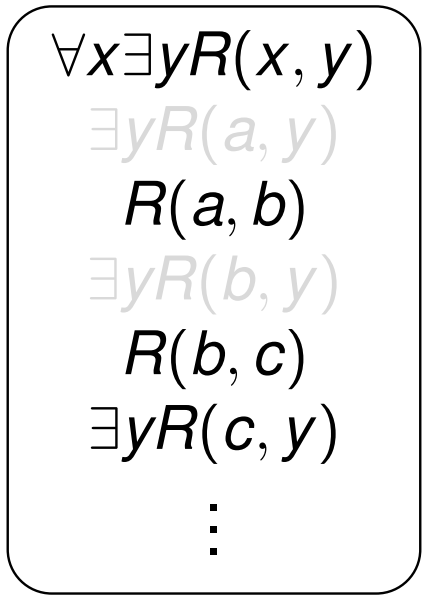
Exemple de branche infinie



Exemple de branche infinie



Exemple de branche infinie



Structures infinies

- Bien que dans certains cas, par exemple pour la plupart des logiques de description, il soit possible d'éviter de telles branches infinies, par exemple en réutilisant les constantes, ceci n'est pas le cas pour la logique du premier ordre.
- Par exemple, une formule sans modèle fini (par exemple, l'ordre total sans extrémités) ne pourra pas donner lieu à une brache finie.
- On peut néanmoins encoder certains types de braches infinies par une description finie, mais ceci, non plus, n'est pas toujours possible.

Incomplétude

- En fait, Gödel a montré en 1930 que la logique du premier-ordre est *indécidable* : il n'y a pas d'algorithme permettant de déterminer si une formule est démontrable ou non et ceci est le cas dès qu'on y formalise les propriétés de bases des entiers naturels avec l'addition et la multiplication.

Conclusion

- La logique du premier-ordre permet d'exprimer des propriétés d'univers d'individus reliés par des relations d'arités quelconques, ce qui est très général.
- La méthode des tableaux est un système déductif complet pour la logique du premier-ordre.
  - Néanmoins un tableau peut être infini, donc il s'agit d'une méthode de *semi-décision*.
- Il reste que pour certains fragments de la logique du premier ordre, tout particulièrement les logiques de description et modales, la méthode des tableaux ne donne lieu qu'à des branches finies.
  - Ceci donne donc un algorithme de décision qui peut être implémenté par des outils informatiques.