# TIPE 25/26 - Cycles et Boucles

## Méthode des tableaux : Optimisation pour des formules de la forme (?)

GIL Dorian

# Sommaire

# Présentation

On souhaite prouver une formule dans la logique propositionelle :

**Definition (Méthode des tableaux)**

Méthode par laquelle on prouve une assertion $B$ ayant pour hypothèse $(A_n)$ en montrant que $\{A_1, \ldots, A_n, \neg B\}$ est insatisfaisable (Cela revient à montrer qu'une implication est vraie car sa négation ne peut être vraie).

# Présentation

On souhaite prouver une formule dans la logique propositionelle :

**Definition (Méthode des tableaux)**

Méthode par laquelle on prouve une assertion $B$ ayant pour hypothèse $(A_n)$ en montrant que $\{A_1, \ldots, A_n, \neg B\}$ est insatisfaisable (Cela revient à montrer qu'une implication est vraie car sa négation ne peut être vraie).

- On place $\neg\phi$ et ses hypothèses dans la racine.
- On applique des règles $(R_x)$ à chaque formule en bout d'arbre qui sont developpables
- Si on trouve $a$ et $\neg a$ dans l'arbre (un *cycle*), alors $\phi$ est vrai

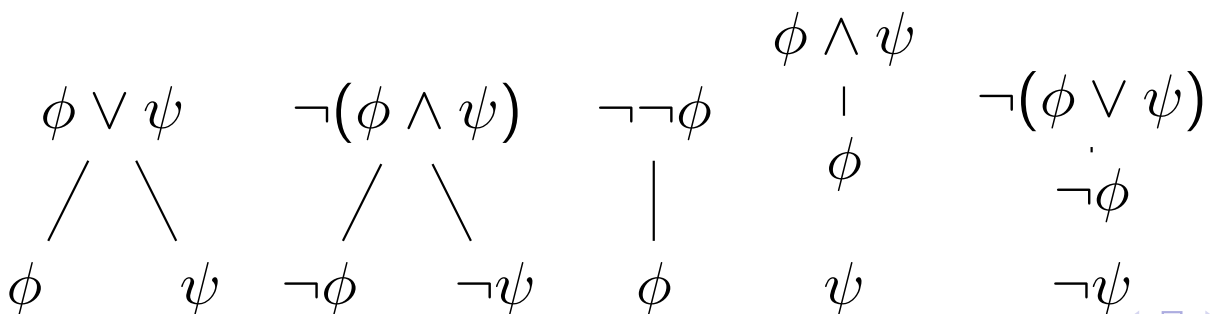
# Présentation

On souhaite prouver une formule dans la logique propositionelle :

**Definition (Méthode des tableaux)**

Méthode par laquelle on prouve une assertion $B$ ayant pour hypothèse $(A_n)$ en montrant que $\{A_1, \ldots, A_n, \neg B\}$ est insatisfaisable (Cela revient à montrer qu'une implication est vraie car sa négation ne peut être vraie).

- On place $\neg\phi$ et ses hypothèses dans la racine.
- On applique des règles $(R_x)$ à chaque formule en bout d'arbre qui sont developpables
- Si on trouve $a$ et $\neg a$ dans l'arbre (un *cycle*), alors $\phi$ est vrai



# Exemple

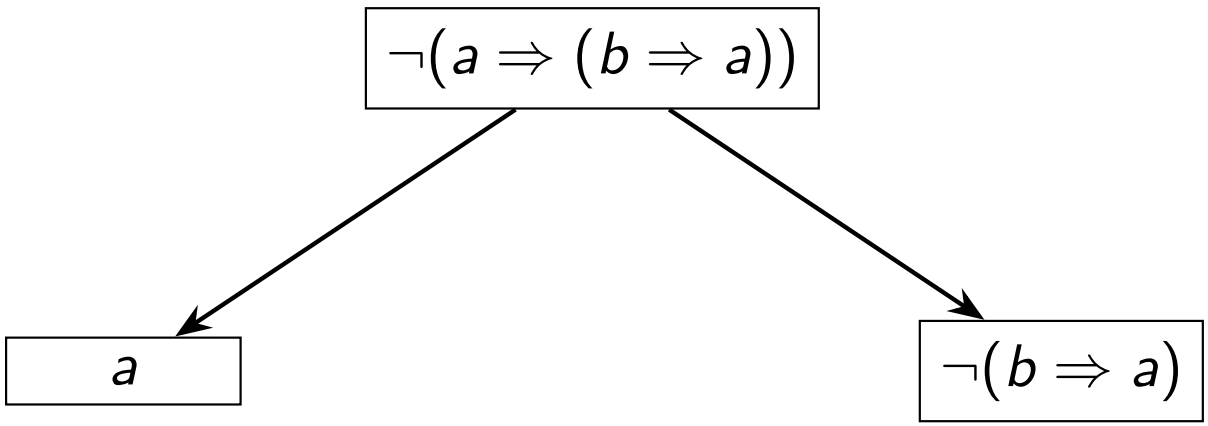
**Formule:** $a \Rightarrow (b \Rightarrow a)$

$$\boxed{\neg(a \Rightarrow (b \Rightarrow a))}$$

On trouve un cycle $a \leftrightarrow \neg a$. Cette négation de la formule est insatisfaisable, donc la formule a été prouvé.

# Exemple

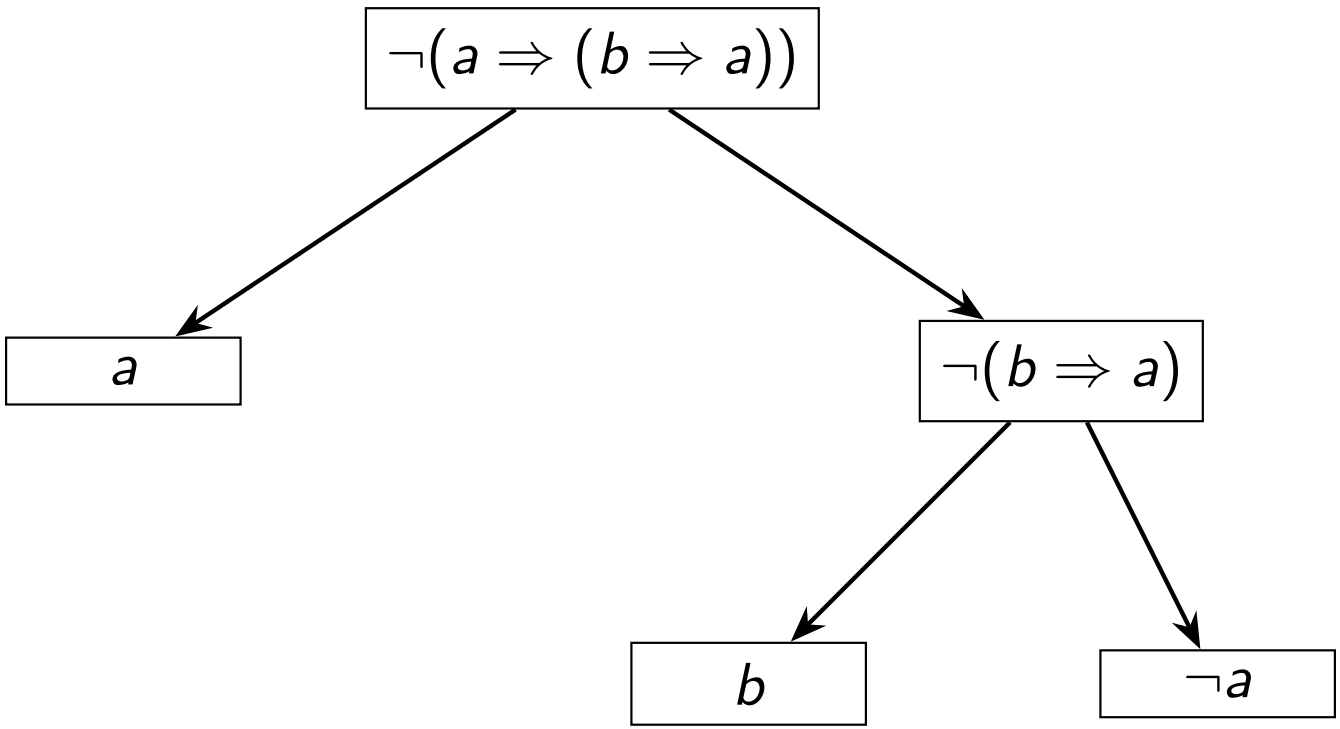
**Formule:** $a \Rightarrow (b \Rightarrow a)$




On trouve un cycle $a \leftrightarrow \neg a$. Cette négation de la formule est insatisfaisable, donc la formule a été prouvé.

# Exemple

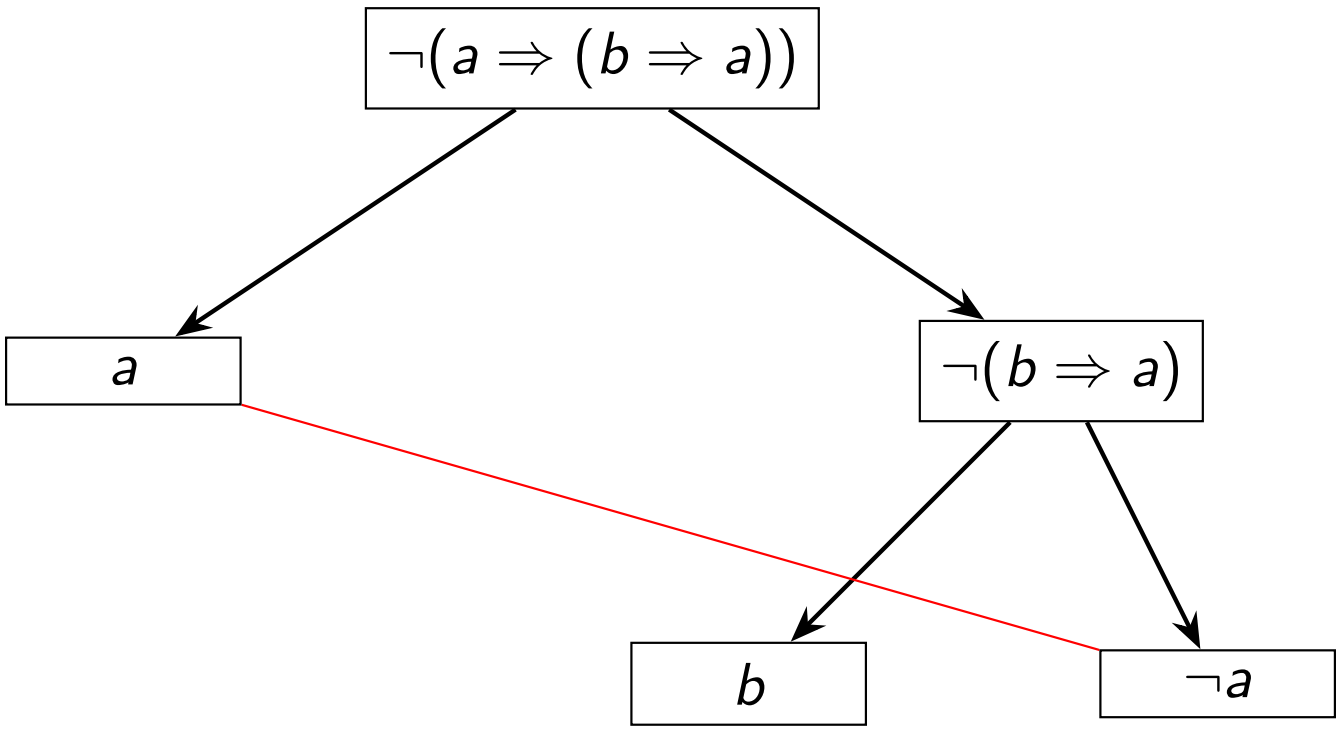
**Formule:** $a \Rightarrow (b \Rightarrow a)$




On trouve un cycle $a \leftrightarrow \neg a$. Cette négation de la formule est insatisfaisable, donc la formule a été prouvé.

# Exemple

**Formule:** $a \Rightarrow (b \Rightarrow a)$




On trouve un cycle $a \leftrightarrow \neg a$. Cette négation de la formule est insatisfaisable, donc la formule a été prouvé.

# Implémentation

Le code

# Ce qui est à faire

Mon but sur le long terme

- Implémenter les tableaux en logique propositionelle

Sur le court terme :

# Ce qui est à faire

Mon but sur le long terme

- Implémenter les tableaux en logique propositionelle
- Trouver et prouver des optimisations pour les formules (?)

Sur le court terme :

# Ce qui est à faire

Mon but sur le long terme

- Implémenter les tableaux en logique propositionelle
- Trouver et prouver des optimisations pour les formules (?)
- Implémenter et commenter les résultats de l'optimisation

Sur le court terme :

# Ce qui est à faire

Mon but sur le long terme

- Implémenter les tableaux en logique propositionelle
- Trouver et prouver des optimisations pour les formules (?)
- Implémenter et commenter les résultats de l'optimisation
- Faire de même cette méthode en logique du première ordre OU continuer à trouver des optimisations dans la logique propositionelle

Sur le court terme :

# Ce qui est à faire

Mon but sur le long terme

- Implémenter les tableaux en logique propositionelle
- Trouver et prouver des optimisations pour les formules (?)
- Implémenter et commenter les résultats de l'optimisation
- Faire de même cette méthode en logique du première ordre OU continuer à trouver des optimisations dans la logique propositionelle

Sur le court terme :

- One

# Ce qui est à faire

Mon but sur le long terme

- Implémenter les tableaux en logique propositionelle
- Trouver et prouver des optimisations pour les formules (?)
- Implémenter et commenter les résultats de l'optimisation
- Faire de même cette méthode en logique du première ordre OU continuer à trouver des optimisations dans la logique propositionelle

Sur le court terme :

- One
- Two