Introduction

La table de vérité

Pour prouver qu'une formule est satisfaisable ou non, qu'une formule est valide ou non, qu'un ensemble est consistent ou non,. . . etc.

Introduction

La table de vérité

Pour prouver qu'une formule est satisfaisable ou non, qu'une formule est valide ou non, qu'un ensemble est consistent ou non,... etc.

Inconvénients

Bien qu'il soit possible d'énumérer toutes les valuations pour établir une tables de vérité, la méthode est coûteuse en espace et en temps :

- n variables dans les formules $\Longrightarrow \quad 2^n$ lignes.

Introduction

La table de vérité

Pour prouver qu'une formule est satisfaisable ou non, qu'une formule est valide ou non, qu'un ensemble est consistent ou non,... etc.

Inconvénients

Bien qu'il soit possible d'énumérer toutes les valuations pour établir une tables de vérité, la méthode est coûteuse en espace et en temps :

- n variables dans les formules $\Longrightarrow \quad 2^n$ lignes.
- formules complexes $\Longrightarrow$ un grand nombre de colonnes.

Introduction

Solution !

Une preuve courte est préférable à une longue liste de valuations.

Introduction

Solution !

Une preuve courte est préférable à une longue liste de valuations.

? Comment prouver

- La satisfisabilité d'une formule
- La validité d'une formule
- La compatibilité d'un ensemble de formules
- Une formule est conséquence d'un ensemble de formules

Introduction

Solution !

Une preuve courte est préférable à une longue liste de valuations.

? Comment prouver

- La satisfisabilité d'une formule
- La validité d'une formule
- La compatibilité d'un ensemble de formules
- Une formule est conséquence d'un ensemble de formules

sans utiliser la table de vérité

Tableaux sémantiques

Tableaux sémantiques

Algorithme pour établir la satisfaisabilité/validité de formules de la logique propositionnelle.

Tableaux sémantiques

Tableaux sémantiques

Algorithme pour établir la satisfaisabilité/validité de formules de la logique propositionnelle.

Le principe est très simple :

Pour prouver la satisfaisabilité d'une formule de la logique propositionnelle, on cherche systématiquement un modèle pour cette formule :
On suppose que φ est Vrai est on cherche un modèle v

Tableaux sémantiques

Rappel :

Une formule propositionnelle φ est un **littéral** *si et seulement si*

- elle est une proposition ou la négation d'une proposition.

Deux formules φ et $\neg\varphi$ sont des formules *complémentaires*.

Tableaux sémantiques

Exemple 1 : $\varphi = a \wedge (\neg b \vee \neg a)$

Tableaux sémantiques

Exemple 1 : $\varphi = a \wedge (\neg b \vee \neg a)$

$$\{a \wedge (\neg b \vee \neg a)\}$$

Tableaux sémantiques

Exemple 1 : $\varphi = a \wedge (\neg b \vee \neg a)$

$$\{a \wedge (\neg b \vee \neg a)\}$$
$$\downarrow$$
$$\{a \text{ , } \neg b \vee \neg a\}$$

Tableaux sémantiques

Exemple 1 : $\varphi = a \wedge (\neg b \vee \neg a)$

$$\{a \wedge (\neg b \vee \neg a)\}$$

$$\downarrow$$

$$\{a \ , \ \neg b \vee \neg a\}$$

$$\swarrow$$

$$\{a \ , \neg b\}$$

Tableaux sémantiques

Exemple 1 : $\varphi = a \wedge (\neg b \vee \neg a)$

$$\{a \wedge (\neg b \vee \neg a)\}$$

$$\{a\ ,\ \neg b \vee \neg a\}$$

$$\{a\ , \neg b\} \qquad \{a\ ,\ \neg a\}$$

Tableaux sémantiques

Exemple 1 : $\varphi = a \wedge (\neg b \vee \neg a)$

$$\{a \wedge (\neg b \vee \neg a)\}$$

$$\downarrow$$

$$\{a \;,\; \neg b \vee \neg a\}$$

$$\swarrow \qquad \searrow$$

$$\{a \;, \neg b\} \qquad\qquad \{a \;,\; \neg a\}$$

$$\boxed{v(a) = 1 \;,\; v(b) = 0}$$

Tableaux sémantiques

Exemple 1 : $\varphi = a \wedge (\neg b \vee \neg a)$

$$\{a \wedge (\neg b \vee \neg a)\}$$

$$\downarrow$$

$$\{a \;,\; \neg b \vee \neg a\}$$

$$\swarrow \qquad\qquad \searrow$$

$$\{a \;, \neg b\} \qquad\qquad \{a \;,\; \neg a\}$$

$$\boxed{v(a) = 1 \;,\; v(b) = 0}$$

Donc, $a \wedge (\neg b \vee \neg a)$ est satisfaisable

Tableaux sémantiques

Preuve de satisfaisabilité :

Le problème de preuve de satisfaisabilité de φ a été réduit à un problème de **satisfaisabilité d'un ensemble de littéraux**.

Tableaux sémantiques

Preuve de satisfaisabilité :

Le problème de preuve de satisfaisabilité de φ a été réduit à un problème de **satisfaisabilité d'un ensemble de littéraux**.

Définiton :

Un ensemble de littéraux est satisfaisable *ssi* il ne contient pas deux littéraux complémentaires.
Par exemple :

$$\{p\ ,\ \neg q\} \qquad\qquad \{p\ ,\ \neg p\}$$

Tableaux sémantiques

Preuve de satisfaisabilité :

Le problème de preuve de satisfaisabilité de φ a été réduit à un problème de **satisfaisabilité d'un ensemble de littéraux**.

Définiton :

Un ensemble de littéraux est satisfaisable *ssi* il ne contient pas deux littéraux complémentaires.
Par exemple :

$$\{p\ ,\ \neg q\} \qquad\qquad \{p\ ,\ \neg p\}$$

Tableaux sémantiques

Exemple 2 : $\varphi = (a \vee b) \wedge (\neg a \wedge \neg b)$

Tableaux sémantiques

Exemple 2 : $\varphi = (a \vee b) \wedge (\neg a \wedge \neg b)$

$$\{(a \vee b) \wedge (\neg a \wedge \neg b)\}$$

Tableaux sémantiques

Exemple 2 : $\varphi = (a \vee b) \wedge (\neg a \wedge \neg b)$

$$\{(a \vee b) \wedge (\neg a \wedge \neg b)\}$$
$$\downarrow$$
$$\{a \vee b \text{ , } \neg a \wedge \neg b\}$$

Tableaux sémantiques

Exemple 2 : $\varphi = (a \vee b) \wedge (\neg a \wedge \neg b)$

$$\{(a \vee b) \wedge (\neg a \wedge \neg b)\}$$
$$\downarrow$$
$$\{a \vee b \;,\; \neg a \wedge \neg b\}$$
$$\downarrow$$
$$\{a \vee b \;, \neg a \;,\; \neg b\}$$

Tableaux sémantiques

Exemple 2 : $\varphi = (a \vee b) \wedge (\neg a \wedge \neg b)$

$$\{(a \vee b) \wedge (\neg a \wedge \neg b)\}$$

$$\downarrow$$

$$\{a \vee b\ ,\ \neg a \wedge \neg b\}$$

$$\downarrow$$

$$\{a \vee b\ , \neg a\ ,\ \neg b\}$$

$$\swarrow$$

$$\{a\ ,\ \neg a\ ,\ \neg b\}$$

Tableaux sémantiques

Exemple 2 : $\varphi = (a \vee b) \wedge (\neg a \wedge \neg b)$

$$\{(a \vee b) \wedge (\neg a \wedge \neg b)\}$$

$$\downarrow$$

$$\{a \vee b \ , \ \neg a \wedge \neg b\}$$

$$\downarrow$$

$$\{a \vee b \ , \neg a \ , \ \neg b\}$$

$$\swarrow \qquad \searrow$$

$$\{a \ , \ \neg a \ , \ \neg b\} \qquad \{b \ , \ \neg a \ , \ \neg b\}$$

Tableaux sémantiques

Exemple 2 : $\varphi = (a \vee b) \wedge (\neg a \wedge \neg b)$

$$\{(a \vee b) \wedge (\neg a \wedge \neg b)\}$$

$$\downarrow$$

$$\{a \vee b \ , \ \neg a \wedge \neg b\}$$

$$\downarrow$$

$$\{a \vee b \ , \neg a \ , \ \neg b\}$$

$$\swarrow \qquad \searrow$$

$$\{a \ , \ \neg a \ , \ \neg b\} \qquad \{b \ , \ \neg a \ , \ \neg b\}$$

Donc, $\varphi = (a \vee b) \wedge (\neg a \wedge \neg b)$ est non satisfaisable

Tableaux sémantiques

Exemple 3 : $\varphi = (a \vee (b \wedge c))$

Tableaux sémantiques

Exemple 3 : $\varphi = (a \vee (b \wedge c))$

$$\{(a \vee (b \wedge c))\}$$

$$\swarrow$$

$$\{a\}$$

Tableaux sémantiques

Exemple 3 : $\varphi = (a \vee (b \wedge c))$

$$\{(a \vee (b \wedge c))\}$$

$$\{a\} \qquad\qquad \{b \wedge c\}$$

Tableaux sémantiques

Exemple 3 : $\varphi = (a \vee (b \wedge c))$

$$\begin{tikzcd}
 & \{(a \vee (b \wedge c))\} \arrow[dl] \arrow[dr] & \\
\{a\} & & \{b \wedge c\} \arrow[d] \\
 & & \{b, c\}
\end{tikzcd}$$

Tableaux sémantiques

Exemple 3 : $\varphi = (a \vee (b \wedge c))$

$$\{(a \vee (b \wedge c))\}$$

$$\{a\} \qquad \{b \wedge c\}$$

$$\{b, c\}$$

	a	b	c
v_1 :	1	0	0
v_2 :	1	0	1
v_3 :	1	1	0
v_4 :	1	1	1

Tableaux sémantiques

Exemple 3 : $\varphi = (a \vee (b \wedge c))$

$$\{(a \vee (b \wedge c))\}$$

$$\{a\} \qquad\qquad \{b \wedge c\}$$

$$\{b, c\}$$

	a	b	c
v_1 :	1	0	0
v_2 :	1	0	1
v_3 :	1	1	0
v_4 :	1	1	1

	a	b	c
v_1 :	0	1	1
v_2 :	1	1	1

Tableaux sémantiques

Exemple 4 : $\varphi = ((a \wedge b) \Rightarrow c)$

Tableaux sémantiques

Exemple 4 : $\varphi = ((a \wedge b) \Rightarrow c)$

$$\{\varphi = ((a \wedge b) \Rightarrow c)\}$$

$$\{\neg(a \wedge b)\}$$

Tableaux sémantiques

Exemple 4 : $\varphi = ((a \wedge b) \Rightarrow c)$

$$\begin{tikzcd}[column sep=small]
& \{\varphi = ((a \wedge b) \Rightarrow c)\} \arrow[dl] \arrow[dr] & \\
\{\neg(a \wedge b)\} & & \{c\}
\end{tikzcd}$$

Tableaux sémantiques

Exemple 4 : $\varphi = ((a \wedge b) \Rightarrow c)$

$$\begin{array}{ccc} & \{\varphi = ((a \wedge b) \Rightarrow c)\} & \\ \swarrow & & \searrow \\ \{\neg(a \wedge b)\} & & \{c\} \\ \swarrow & & \\ \{\neg a\} & & \end{array}$$

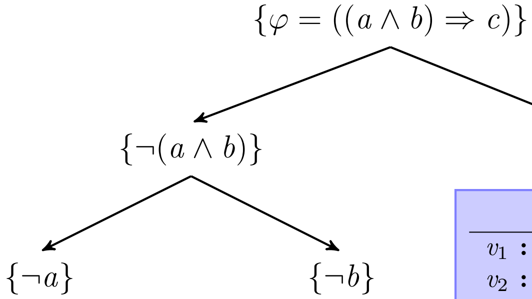
Tableaux sémantiques

Exemple 4 : $\varphi = ((a \wedge b) \Rightarrow c)$

$$\begin{tikzcd}[column sep=small]
 & & \{\varphi = ((a \wedge b) \Rightarrow c)\} \arrow[dl] \arrow[dr] & \\
 & \{\neg(a \wedge b)\} \arrow[dl] \arrow[dr] & & \{c\} \\
\{\neg a\} & & \{\neg b\} &
\end{tikzcd}$$

Tableaux sémantiques

Exemple 4 : $\varphi = ((a \wedge b) \Rightarrow c)$



	a	b	c
v_1 :	0	0	1
v_2 :	0	1	1
v_3 :	1	0	1
v_4 :	1	1	1

Tableaux sémantiques

Exemple 4 : $\varphi = ((a \wedge b) \Rightarrow c)$

$$\{\varphi = ((a \wedge b) \Rightarrow c)\}$$

$$\{\neg(a \wedge b)\} \qquad \{c\}$$

$$\{\neg a\} \qquad \{\neg b\}$$

	a	**b**	**c**
v_1 :	**0**	**0**	**0**
v_2 :	**0**	**0**	**1**
v_3 :	**0**	**1**	**0**
v_4 :	**0**	**1**	**1**

	a	**b**	**c**
v_1 :	**0**	**0**	**1**
v_2 :	**0**	**1**	**1**
v_3 :	**1**	**0**	**1**
v_4 :	**1**	**1**	**1**

Tableaux sémantiques

Exemple 4 : $\varphi = ((a \wedge b) \Rightarrow c)$

$$\begin{tikzcd}
 & & \{\varphi = ((a \wedge b) \Rightarrow c)\} \arrow[dl] \arrow[dr] & \\
 & \{\neg(a \wedge b)\} \arrow[dl] \arrow[dr] & & \{c\} \\
\{\neg a\} & & \{\neg b\} &
\end{tikzcd}$$

	a	b	c
v_1 :	0	0	0
v_2 :	0	0	1
v_3 :	0	1	0
v_4 :	0	1	1

	a	b	c
v_1 :	0	0	0
v_2 :	0	0	1
v_3 :	1	0	0
v_4 :	1	0	1

	a	b	c
v_1 :	0	0	1
v_2 :	0	1	1
v_3 :	1	0	1
v_4 :	1	1	1

Tableaux sémantiques

α-règles

Quand on sélectionne une *conjonction*,on a une seule branche. On appelle ce type de règles, les *α-règles*

β-règles

Quand on sélectionne une *disjonction*,on a une deux branche. On appelle ce type de règles, les *β-règles*

Tableaux sémantiques

$\{\alpha\}$	$\{\alpha_1 \ , \ \alpha_2\}$
$\{\neg\neg\varphi\}$	$\{\varphi\}$

Tableaux sémantiques

$\{\alpha\}$	$\{\alpha_1 \;, \alpha_2\}$
$\{\neg\neg\varphi\}$	$\{\varphi\}$
$\{\varphi_1 \wedge \varphi_2\}$	$\{\varphi_1 \;, \varphi_2\}$

Tableaux sémantiques

$\{\alpha\}$	$\{\alpha_1 , \alpha_2\}$
$\{\neg\neg\varphi\}$	$\{\varphi\}$
$\{\varphi_1 \wedge \varphi_2\}$	$\{\varphi_1 , \varphi_2\}$
$\{\neg(\varphi_1 \vee \varphi_2)\}$	$\{\neg\varphi_1 , \neg\varphi_2\}$

Tableaux sémantiques

$\{\alpha\}$	$\{\alpha_1 \;,\; \alpha_2\}$
$\{\neg\neg\varphi\}$	$\{\varphi\}$
$\{\varphi_1 \wedge \varphi_2\}$	$\{\varphi_1 \;,\; \varphi_2\}$
$\{\neg(\varphi_1 \vee \varphi_2)\}$	$\{\neg\varphi_1 \;,\; \neg\varphi_2\}$
$\{\neg(\varphi_1 \Rightarrow \varphi_2)\}$	$\{\varphi_1 \;,\; \neg\varphi_2\}$

Tableaux sémantiques

$\{\alpha\}$	$\{\alpha_1 \ , \ \alpha_2\}$
$\{\neg\neg\varphi\}$	$\{\varphi\}$
$\{\varphi_1 \wedge \varphi_2\}$	$\{\varphi_1 \ , \ \varphi_2\}$
$\{\neg(\varphi_1 \vee \varphi_2)\}$	$\{\neg\varphi_1 \ , \ \neg\varphi_2\}$
$\{\neg(\varphi_1 \Rightarrow \varphi_2)\}$	$\{\varphi_1 \ , \ \neg\varphi_2\}$
$\{\varphi_1 \Leftrightarrow \varphi_2\}$	$\{\varphi_1 \Rightarrow \varphi_2 \ , \ \varphi_2 \Rightarrow \varphi_1\}$

Tableaux sémantiques

$\{\alpha\}$	$\{\alpha_1 \ , \ \alpha_2\}$
$\{\neg\neg\varphi\}$	$\{\varphi\}$
$\{\varphi_1 \wedge \varphi_2\}$	$\{\varphi_1 \ , \ \varphi_2\}$
$\{\neg(\varphi_1 \vee \varphi_2)\}$	$\{\neg\varphi_1 \ , \ \neg\varphi_2\}$
$\{\neg(\varphi_1 \Rightarrow \varphi_2)\}$	$\{\varphi_1 \ , \ \neg\varphi_2\}$
$\{\varphi_1 \Leftrightarrow \varphi_2\}$	$\{\varphi_1 \Rightarrow \varphi_2 \ , \ \varphi_2 \Rightarrow \varphi_1\}$

TABLE : α-règles

Remarque :

Toutes les formules α peuvent être considérées comme équivalentes à des conjonctions.
Par exemple : $\neg(\varphi_1 \vee \varphi_2)$ est équivalente à $\neg\varphi_1 \wedge \neg\varphi_2$

Tableaux sémantiques

$\{\beta\}$	$\{\beta_1\}$, $\{\beta_2\}$
$\{\varphi_1 \vee \varphi_2\}$	$\{\varphi_1\}$, $\{\varphi_2\}$

Tableaux sémantiques

$\{\beta\}$	$\{\beta_1\}$, $\{\beta_2\}$
$\{\varphi_1 \vee \varphi_2\}$	$\{\varphi_1\}$, $\{\varphi_2\}$
$\{\neg(\varphi_1 \wedge \varphi_2)\}$	$\{\neg\varphi_1\}$, $\{\neg\varphi_2\}$

Tableaux sémantiques

$\{\beta\}$	$\{\beta_1\}$, $\{\beta_2\}$
$\{\varphi_1 \vee \varphi_2\}$	$\{\varphi_1\}$, $\{\varphi_2\}$
$\{\neg(\varphi_1 \wedge \varphi_2)\}$	$\{\neg\varphi_1\}$, $\{\neg\varphi_2\}$
$\{\varphi_1 \Rightarrow \varphi_2\}$	$\{\neg\varphi_1\}$, $\{\varphi_2\}$

Tableaux sémantiques

$\{\beta\}$	$\{\beta_1\}$, $\{\beta_2\}$
$\{\varphi_1 \vee \varphi_2\}$	$\{\varphi_1\}$, $\{\varphi_2\}$
$\{\neg(\varphi_1 \wedge \varphi_2)\}$	$\{\neg\varphi_1\}$, $\{\neg\varphi_2\}$
$\{\varphi_1 \Rightarrow \varphi_2\}$	$\{\neg\varphi_1\}$, $\{\varphi_2\}$
$\{\neg(\varphi_1 \Leftrightarrow \varphi_2)\}$	$\{\neg(\varphi_1 \Rightarrow \varphi_2)\}$, $\{\neg(\varphi_2 \Rightarrow \varphi_1)\}$

Tableaux sémantiques

$\{\beta\}$	$\{\beta_1\}\ ,\ \{\beta_2\}$
$\{\varphi_1 \vee \varphi_2\}$	$\{\varphi_1\}\ ,\ \{\varphi_2\}$
$\{\neg(\varphi_1 \wedge \varphi_2)\}$	$\{\neg\varphi_1\}\ ,\ \{\neg\varphi_2\}$
$\{\varphi_1 \Rightarrow \varphi_2\}$	$\{\neg\varphi_1\}\ ,\ \{\varphi_2\}$
$\{\neg(\varphi_1 \Leftrightarrow \varphi_2)\}$	$\{\neg(\varphi_1 \Rightarrow \varphi_2)\}\ ,\ \{\neg(\varphi_2 \Rightarrow \varphi_1)\}$

TABLE : β-règles

Remarque :

Toutes les formules β peuvent être considérées comme équivalentes à des disjonctions.

Par exemple : $\neg(\varphi_1 \wedge \varphi_2)$ est équivalente à $\neg\varphi_1 \vee \neg\varphi_2$

Algorithme de construction

Quelques notations :

- Un arbre représentant le tableau sémantique d'une formule φ sera noté par T_{φ}.
- l, m, n représentent des nœuds d'un arbre.
- $Lab(l)$ dénote l'étiquette du nœud l , c'est un ensemble de formules.
- $Status(l)$ est le statut du nœud l :
  - — pour tout nœud qui n'est pas une feuille, et
  - $Status(l) \in \{ouvert\ ,\ fermé\}$ si l est une feuille.
- R dénote un ensemble de nœuds.

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
  Choisir $l \in R$;
  si *Lab(l) est un ensemble de littéraux* **alors**
    si *Lab(l) contient une paire de littéraux complémentaires* **alors**
      Status(l) $\leftarrow$ *fermé*
    sinon
      Status(l) $\leftarrow$ *ouvert*
    $R \leftarrow R \setminus \{l\}$;
  sinon
    Choisir $\psi \in Lab(l)$;
    si *ψ est une α-formule* **alors**
      Créer un fils du noeud l, soit m;
      $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
    si *ψ est une β-formule* **alors**
      Créer deux fils du noeud l, soit m et n;
      $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
      $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
    Choisir $l \in R$;
    si *$Lab(l)$ est un ensemble de littéraux* **alors**
        si *$Lab(l)$ contient une paire de littéraux complémentaires* **alors**
            $Status(l) \leftarrow$ *fermé*
        sinon
            $Status(l) \leftarrow$ *ouvert*
        $R \leftarrow R \setminus \{l\}$;
    sinon
        Choisir $\psi \in Lab(l)$;
        si *ψ est une α-formule* **alors**
            Créer un fils du noeud l, soit m;
            $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
        si *ψ est une β-formule* **alors**
            Créer deux fils du noeud l, soit m et n;
            $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
            $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
  Choisir $l \in R$;
  si *$Lab(l)$ est un ensemble de littéraux* **alors**
    si *$Lab(l)$ contient une paire de littéraux complémentaires* **alors**
      $Status(l) \leftarrow fermé$
    sinon
      $Status(l) \leftarrow ouvert$
    $R \leftarrow R \setminus \{l\}$;
  sinon
    Choisir $\psi \in Lab(l)$;
    si *ψ est une α-formule* **alors**
      Créer un fils du noeud l, soit m;
      $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
    si *ψ est une β-formule* **alors**
      Créer deux fils du noeud l, soit m et n;
      $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
      $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
    Choisir $l \in R$;
    si *$Lab(l)$ est un ensemble de littéraux* **alors**
        si *$Lab(l)$ contient une paire de littéraux complémentaires* **alors**
            $Status(l) \leftarrow$ *fermé*
        sinon
            $Status(l) \leftarrow$ *ouvert*
        $R \leftarrow R \setminus \{l\}$;
    sinon
        Choisir $\psi \in Lab(l)$;
        si *ψ est une α-formule* **alors**
            Créer un fils du noeud l, soit m;
            $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
        si *ψ est une β-formule* **alors**
            Créer deux fils du noeud l, soit m et n;
            $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
            $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
    Choisir $l \in R$;
    si *Lab(l) est un ensemble de littéraux* **alors**
        si *Lab(l) contient une paire de littéraux complémentaires* **alors**
            $Status(l) \leftarrow$ *fermé*
        sinon
            $Status(l) \leftarrow$ *ouvert*
        $R \leftarrow R \setminus \{l\}$;
    sinon
        Choisir $\psi \in Lab(l)$;
        si *ψ est une α-formule* **alors**
            Créer un fils du noeud l, soit m;
            $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
        si *ψ est une β-formule* **alors**
            Créer deux fils du noeud l, soit m et n;
            $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
            $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
    Choisir $l \in R$;
    si *$Lab(l)$ est un ensemble de littéraux* **alors**
        si *$Lab(l)$ contient une paire de littéraux complémentaires* **alors**
            $Status(l) \leftarrow fermé$
        sinon
            $Status(l) \leftarrow ouvert$
        $R \leftarrow R \setminus \{l\}$;
    sinon
        Choisir $\psi \in Lab(l)$;
        si *ψ est une α-formule* **alors**
            Créer un fils du noeud l, soit m;
            $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
        si *ψ est une β-formule* **alors**
            Créer deux fils du noeud l, soit m et n;
            $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
            $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
    Choisir $l \in R$;
    si *$Lab(l)$ est un ensemble de littéraux* **alors**
        si *$Lab(l)$ contient une paire de littéraux complémentaires* **alors**
            $Status(l) \leftarrow fermé$
        sinon
            $Status(l) \leftarrow ouvert$
        $R \leftarrow R \setminus \{l\}$;
    sinon
        Choisir $\psi \in Lab(l)$;
        si *ψ est une α-formule* **alors**
            Créer un fils du noeud l, soit m;
            $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
        si *ψ est une β-formule* **alors**
            Créer deux fils du noeud l, soit m et n;
            $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
            $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
 Choisir $l \in R$;
 si *$Lab(l)$ est un ensemble de littéraux* **alors**
  si *$Lab(l)$ contient une paire de littéraux complémentaires* **alors**
   $Status(l) \leftarrow fermé$
  sinon
   $Status(l) \leftarrow ouvert$
  $R \leftarrow R \setminus \{l\}$;
 sinon
  Choisir $\psi \in Lab(l)$;
  si *ψ est une α-formule* **alors**
   Créer un fils du noeud l, soit m;
   $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
   $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
  si *ψ est une β-formule* **alors**
   Créer deux fils du noeud l, soit m et n;
   $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
   $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
   $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
  Choisir $l \in R$;
  si *$Lab(l)$ est un ensemble de littéraux* **alors**
    si *$Lab(l)$ contient une paire de littéraux complémentaires* **alors**
      $Status(l) \leftarrow$ *fermé*
    sinon
      $Status(l) \leftarrow$ *ouvert*
    $R \leftarrow R \setminus \{l\}$;
  sinon
    Choisir $\psi \in Lab(l)$;
    si *ψ est une α-formule* **alors**
      Créer un fils du noeud l, soit m;
      $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
    si *ψ est une β-formule* **alors**
      Créer deux fils du noeud l, soit m et n;
      $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
      $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
    Choisir $l \in R$;
    si *$Lab(l)$ est un ensemble de littéraux* **alors**
        si *$Lab(l)$ contient une paire de littéraux complémentaires* **alors**
            $Status(l) \leftarrow fermé$
        sinon
            $Status(l) \leftarrow ouvert$
        $R \leftarrow R \setminus \{l\}$;
    sinon
        Choisir $\psi \in Lab(l)$;
        si *ψ est une α-formule* **alors**
            Créer un fils du noeud l, soit m;
            $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
        si *ψ est une β-formule* **alors**
            Créer deux fils du noeud l, soit m et n;
            $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
            $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
    Choisir $l \in R$;
    si *Lab(l) est un ensemble de littéraux* **alors**
        si *Lab(l) contient une paire de littéraux complémentaires* **alors**
            $Status(l) \leftarrow$ *fermé*
        sinon
            $Status(l) \leftarrow$ *ouvert*
        $R \leftarrow R \setminus \{l\}$;
    sinon
        Choisir $\psi \in Lab(l)$;
        si *ψ est une α-formule* **alors**
            Créer un fils du noeud l, soit m;
            $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
        si *ψ est une β-formule* **alors**
            Créer deux fils du noeud l, soit m et n;
            $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
            $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
  Choisir $l \in R$;
  si $Lab(l)$ *est un ensemble de littéraux* **alors**
    si $Lab(l)$ *contient une paire de littéraux complémentaires* **alors**
      $Status(l) \leftarrow fermé$
    sinon
      $Status(l) \leftarrow ouvert$
    $R \leftarrow R \setminus \{l\}$;
  sinon
    Choisir $\psi \in Lab(l)$;
    si ψ *est une* α*-formule* **alors**
      Créer un fils du noeud l, soit m;
      $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
    si ψ *est une* β*-formule* **alors**
      Créer deux fils du noeud l, soit m et n;
      $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
      $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
    Choisir $l \in R$;
    si *Lab(l) est un ensemble de littéraux* **alors**
        si *Lab(l) contient une paire de littéraux complémentaires* **alors**
            $Status(l) \leftarrow fermé$
        sinon
            $Status(l) \leftarrow ouvert$
        $R \leftarrow R \setminus \{l\}$;
    sinon
        Choisir $\psi \in Lab(l)$;
        si *ψ est une α-formule* **alors**
            Créer un fils du noeud l, soit m;
            $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
        si *ψ est une β-formule* **alors**
            Créer deux fils du noeud l, soit m et n;
            $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
            $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
  Choisir $l \in R$;
  si *$Lab(l)$ est un ensemble de littéraux* **alors**
    si *$Lab(l)$ contient une paire de littéraux complémentaires* **alors**
      $Status(l) \leftarrow fermé$
    sinon
      $Status(l) \leftarrow ouvert$
    $R \leftarrow R \setminus \{l\}$;
  sinon
    Choisir $\psi \in Lab(l)$;
    si *ψ est une α-formule* **alors**
      Créer un fils du noeud l, soit m;
      $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
    si *ψ est une β-formule* **alors**
      Créer deux fils du noeud l, soit m et n;
      $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
      $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
    Choisir $l \in R$;
    si *$Lab(l)$ est un ensemble de littéraux* **alors**
        si *$Lab(l)$ contient une paire de littéraux complémentaires* **alors**
            $Status(l) \leftarrow fermé$
        sinon
            $Status(l) \leftarrow ouvert$
        $R \leftarrow R \setminus \{l\}$;
    sinon
        Choisir $\psi \in Lab(l)$;
        si *ψ est une α-formule* **alors**
            Créer un fils du noeud l, soit m;
            $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
        si *ψ est une β-formule* **alors**
            Créer deux fils du noeud l, soit m et n;
            $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
            $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
    Choisir $l \in R$;
    si *$Lab(l)$ est un ensemble de littéraux* **alors**
        si *$Lab(l)$ contient une paire de littéraux complémentaires* **alors**
            $Status(l) \leftarrow fermé$
        sinon
            $Status(l) \leftarrow ouvert$
        $R \leftarrow R \setminus \{l\}$;
    sinon
        Choisir $\psi \in Lab(l)$;
        si *ψ est une α-formule* **alors**
            Créer un fils du noeud l, soit m;
            $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
        si *ψ est une β-formule* **alors**
            Créer deux fils du noeud l, soit m et n;
            $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
            $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
            $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
  Choisir $l \in R$;
  si *Lab(l) est un ensemble de littéraux* **alors**
    si *Lab(l) contient une paire de littéraux complémentaires* **alors**
      $Status(l) \leftarrow fermé$
    sinon
      $Status(l) \leftarrow ouvert$
    $R \leftarrow R \setminus \{l\}$;
  sinon
    Choisir $\psi \in Lab(l)$;
    si *ψ est une α-formule* **alors**
      Créer un fils du noeud l, soit m;
      $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
    si *ψ est une β-formule* **alors**
      Créer deux fils du noeud l, soit m et n;
      $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
      $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

$R \leftarrow \{l\}$;
$Lab(l) \leftarrow \{\varphi\}$;
tant que $R \neq \emptyset$ **faire**
  Choisir $l \in R$;
  si *$Lab(l)$ est un ensemble de littéraux* **alors**
    si *$Lab(l)$ contient une paire de littéraux complémentaires* **alors**
      $Status(l) \leftarrow fermé$
    sinon
      $Status(l) \leftarrow ouvert$
    $R \leftarrow R \setminus \{l\}$;
  sinon
    Choisir $\psi \in Lab(l)$;
    si *ψ est une α-formule* **alors**
      Créer un fils du noeud l, soit m;
      $R \leftarrow (R \setminus \{l\}) \cup \{m\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\alpha_1, \alpha_2\}$;
    si *ψ est une β-formule* **alors**
      Créer deux fils du noeud l, soit m et n;
      $R \leftarrow (R \setminus \{l\}) \cup \{m, n\}$;
      $Lab(m) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_1\}$;
      $Lab(n) \leftarrow (Lab(l) \setminus \{\psi\}) \cup \{\beta_2\}$;

Algorithme de construction

Notations :

- Un tableau dont la construction est terminée est appelé un *tableau complet*.

Algorithme de construction

Notations :

- Un tableau dont la construction est terminée est appelé un *tableau complet*.
- Un tableau **fermé** est un tableau complet dont toutes les feuilles sont étiquettées avec fermé.

Algorithme de construction

Notations :

- Un tableau dont la construction est terminée est appelé un *tableau complet*.
- Un tableau **fermé** est un tableau complet dont toutes les feuilles sont étiquettées avec fermé.

Algorithme de construction

Notations :

- Un tableau dont la construction est terminée est appelé un *tableau complet*.
- Un tableau **fermé** est un tableau complet dont toutes les feuilles sont étiquettées avec fermé.C'est un tableau **ouvert** sinon.

Algorithme de construction

Définition

Si T_φ est le tableau complet de la formule φ :

- φ est non satisfaisable *ssi* T_φ est fermé.

Algorithme de construction

Définition

Si T_φ est le tableau complet de la formule φ :

- φ est non satisfaisable *ssi* T_φ est fermé.
- φ est satisfaisable *ssi*

Algorithme de construction

Définition

Si T_φ est le tableau complet de la formule φ :

- φ est non satisfaisable *ssi* T_φ est fermé.
- φ est satisfaisable *ssi*

Algorithme de construction

Définition

Si T_φ est le tableau complet de la formule φ :

- φ est non satisfaisable *ssi* T_φ est fermé.
- φ est satisfaisable *ssi* T_φ est ouvert.
- φ est valide ssi

? Question :

Si toutes les feuilles de T_φ sont ouvertes, est-ce que cela signifie de φ est valide ?

Algorithme de construction

Définition

Si T_φ est le tableau complet de la formule φ :

- φ est non satisfaisable *ssi* T_φ est fermé.
- φ est satisfaisable *ssi* T_φ est ouvert.
- φ est valide ssi

? Question :

Si toutes les feuilles de T_φ sont ouvertes, est-ce que cela signifie de φ est valide ?

Algorithme de construction

Définition

Si T_φ est le tableau complet de la formule φ :

- φ est non satisfaisable *ssi* T_φ est fermé.
- φ est satisfaisable *ssi* T_φ est ouvert.
- φ est valide ssi $T_{\neg\varphi}$ est fermé.

Algorithme de construction

Définition

Si T_φ est le tableau complet de la formule φ :

- φ est non satisfaisable *ssi* T_φ est fermé.
- φ est satisfaisable *ssi* T_φ est ouvert.
- φ est valide ssi $T_{\neg\varphi}$ est fermé.

? Question :

Si toutes les feuilles de T_φ sont ouvertes, est-ce que cela signifie de φ est valide ?

Preuve de la validité d'une formule

Rappel

- La formule φ est est valide *ssi* $\neg\varphi$ est non satisfaisable.

Preuve de la validité d'une formule

Rappel

- La formule φ est est valide *ssi* $\neg\varphi$ est non satisfaisable.
- Prouver que φ est valide revient à prouver que $\neg\varphi$ est non satisfaisable.

Preuve de la validité d'une formule

Rappel

- La formule φ est est valide *ssi* $\neg\varphi$ est non satisfaisable.
- Prouver que φ est valide revient à prouver que $\neg\varphi$ est non satisfaisable.

Preuve de la validité d'une formule

Rappel

- La formule φ est est valide *ssi* $\neg\varphi$ est non satisfaisable.
- Prouver que φ est valide revient à prouver que $\neg\varphi$ est non satisfaisable.

Validité

1. $\psi = \neg\varphi$

Preuve de la validité d'une formule

Rappel

- La formule φ est est valide *ssi* $\neg\varphi$ est non satisfaisable.
- Prouver que φ est valide revient à prouver que $\neg\varphi$ est non satisfaisable.

Validité

1. $\psi = \neg\varphi$
2. Algorithme des tableaux sémantiques précédent pour ψ

Preuve de la validité d'une formule

Rappel

- La formule φ est est valide *ssi* $\neg\varphi$ est non satisfaisable.
- Prouver que φ est valide revient à prouver que $\neg\varphi$ est non satisfaisable.

Validité

1. $\psi = \neg\varphi$
2. Algorithme des tableaux sémantiques précédent pour ψ
3. Si ψ est non satisfaisable alors φ est valide, sinon φ est non valide.

Preuve de la validité d'une formule

Exemple : $\varphi = (a \vee \neg a)$ est valide ?

Preuve de la validité d'une formule

Exemple : $\varphi = (a \vee \neg a)$ est valide ?
On pose $\psi = \neg\varphi$

Preuve de la validité d'une formule

Exemple : $\varphi = (a \vee \neg a)$ est valide ?
On pose $\psi = \neg\varphi$

$$\{\neg(a \vee \neg a)\}$$

Preuve de la validité d'une formule

Exemple : $\varphi = (a \vee \neg a)$ est valide ?

On pose $\psi = \neg\varphi$

$$\{\neg(a \vee \neg a)\}$$
$$\downarrow$$
$$\{\neg a\ ,\ \neg\neg a\}$$

Preuve de la validité d'une formule

Exemple : $\varphi = (a \vee \neg a)$ est valide ?
On pose $\psi = \neg\varphi$

$$\{\neg(a \vee \neg a)\}$$
$$\downarrow$$
$$\{\neg a \ , \ \neg\neg a\}$$
$$\downarrow$$
$$\{\neg a \ , \ \neg\neg a\}$$

Preuve de la validité d'une formule

Exemple : $\varphi = (a \vee \neg a)$ est valide ?
On pose $\psi = \neg\varphi$

$$\begin{array}{c}
\{\neg(a \vee \neg a)\} \\
\downarrow \\
\{\neg a \ , \ \neg\neg a\} \\
\downarrow \\
\{\neg a \ , \ \neg\neg a\} \\
\downarrow \\
\{\neg a \ , \ a\}
\end{array}$$

Preuve de la validité d'une formule

Exemple : $\varphi = (a \vee \neg a)$ est valide ?
On pose $\psi = \neg\varphi$

$$\{\neg(a \vee \neg a)\}$$
$$\downarrow$$
$$\{\neg a \ , \ \neg\neg a\}$$
$$\downarrow$$
$$\{\neg a \ , \ \neg\neg a\}$$
$$\downarrow$$
$$\{\neg a \ , \ a\} \quad \text{fermé}$$

ψ est **non satisfaisable**. Donc, φ est valide.

Preuve de la validité d'une formule

Exemple : $\varphi = (a \wedge \neg b)$ est valide ?

Preuve de la validité d'une formule

Exemple : $\varphi = (a \wedge \neg b)$ est valide ?
On pose $\psi = \neg\varphi$

Preuve de la validité d'une formule

Exemple : $\varphi = (a \wedge \neg b)$ est valide ?
On pose $\psi = \neg\varphi$

$$\{\neg(a \wedge \neg b)\}$$

Preuve de la validité d'une formule

Exemple : $\varphi = (a \wedge \neg b)$ est valide ?
On pose $\psi = \neg\varphi$

$$
\begin{array}{ccc}
 & & \{\neg(a \wedge \neg b)\} \\
 & \swarrow & \\
\{\neg a\} & &
\end{array}
$$

Preuve de la validité d'une formule

Exemple : $\varphi = (a \wedge \neg b)$ est valide ?
On pose $\psi = \neg\varphi$

$$\begin{tikzcd}
& \{\neg(a \wedge \neg b)\} \arrow[dl] \arrow[dr] & \\
\{\neg a\} & & \{\neg\neg b\}
\end{tikzcd}$$

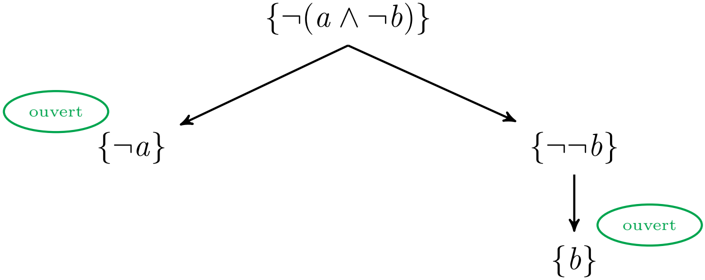
Preuve de la validité d'une formule

Exemple : $\varphi = (a \wedge \neg b)$ est valide ?
On pose $\psi = \neg\varphi$

$$\begin{tikzcd}
 & \{\neg(a \wedge \neg b)\} \arrow[dl] \arrow[dr] & \\
\{\neg a\} & & \{\neg\neg b\} \arrow[d] \\
 & & \{b\}
\end{tikzcd}$$

Preuve de la validité d'une formule

Exemple : $\varphi = (a \land \neg b)$ est valide ?
On pose $\psi = \neg\varphi$



ψ est **satisfaisable**. Donc, φ est non valide.

Preuve de la compatiblité d'un ensemble de formules

Rappel

Un ensemble de formule Γ est **satisfaisable** s'il admet au moins un modèle :

- S'il existe une valuation v telle que $v \models \varphi$ pour toute $\varphi \in \Gamma$

Preuve de la compatiblité d'un ensemble de formules

Rappel

Un ensemble de formule Γ est **satisfaisable** s'il admet au moins un modèle :

- S'il existe une valuation v telle que $v \models \varphi$ pour toute $\varphi \in \Gamma$
- Si $\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$ est satisfaisable.

Preuve de la compatiblité d'un ensemble de formules

Rappel

Un ensemble de formule Γ est **satisfaisable** s'il admet au moins un modèle :

- S'il existe une valuation v telle que $v \models \varphi$ pour toute $\varphi \in \Gamma$
- Si $\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$ est satisfaisable.

Preuve de la compatiblité d'un ensemble de formules

Rappel

Un ensemble de formule Γ est **satisfaisable** s'il admet au moins un modèle :

- S'il existe une valuation v telle que $v \models \varphi$ pour toute $\varphi \in \Gamma$
- Si $\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$ est satisfaisable.

? $\Sigma = \{\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n\}$ **est satisfaisable**

1. $\psi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$

Preuve de la compatiblité d'un ensemble de formules

Rappel

Un ensemble de formule Γ est **satisfaisable** s'il admet au moins un modèle :

- S'il existe une valuation v telle que $v \models \varphi$ pour toute $\varphi \in \Gamma$
- Si $\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$ est satisfaisable.

? $\Sigma = \{\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n\}$ **est satisfaisable**

1. $\psi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$
2. Algorithme des tableaux sémantiques précédent pour ψ

Preuve de la compatiblité d'un ensemble de formules

Rappel

Un ensemble de formule Γ est **satisfaisable** s'il admet au moins un modèle :

- S'il existe une valuation v telle que $v \models \varphi$ pour toute $\varphi \in \Gamma$
- Si $\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$ est satisfaisable.

? $\Sigma = \{\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n\}$ **est satisfaisable**

1. $\psi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$
2. Algorithme des tableaux sémantiques précédent pour ψ
3. Si ψ est satisfaisable alors Σ est compbatible, sinon Σ est contradictoire.