Chapitre 2
Définition de la logique propositionnelle
Cours 3
Sémantique de la logique propositionnelle

Affectations

Valeurs de vérités : 0 et 1.

Definition

Une *affectation* est une fonction

$$v : V \to \{0, 1\}$$

Le support d'une affectation v est défini comme

$$supp(v) = \{x \in V \mid v(x) = 1\}$$

Affectations

Valeurs de vérités : 0 et 1.

Definition

Une *affectation* est une fonction

$$v : V \to \{0, 1\}$$

Le support d'une affectation v est défini comme

$$supp(v) = \{x \in V \mid v(x) = 1\}$$

Il existe des définitions différentes

- Pour 0 : False, ff, ...
- Pour 1 : True, tt, ...
- Les affectations sont parfois définies comme des fonctions partielles.

Il existe des définitions différentes

- Pour 0 : False, ff, ...
- Pour 1 : True, tt, ...
- Les affectations sont parfois définies comme des fonctions partielles.

Notation pour les affectations

- Nous écrivons

$$[x_1 \mapsto 1, x_2 \mapsto 1, x_3 \mapsto 1, \ldots, x_n \mapsto 1]$$

  pour l'affectation qui aux variables $x_1, \ldots, x_n$ associe la valeur 1, et qui associe à toute autre variable la valeur 0.
- On ne peut pas écrire toutes les affectations de cette façon.

Notation pour les affectations

- Nous écrivons

$$[x_1 \mapsto 1, x_2 \mapsto 1, x_3 \mapsto 1, \ldots, x_n \mapsto 1]$$

  pour l'affectation qui aux variables $x_1, \ldots, x_n$ associe la valeur 1, et qui associe à toute autre variable la valeur 0.
- On ne peut pas écrire toutes les affectations de cette façon.

Exemple affectation

$$[x_1 \mapsto 1, x_2 \mapsto 0, x_3 \mapsto 1]$$

est l'affectation qui associe à x_1 et x_3 la valeur 1, et à toute autre variable la valeur 0.
Son support est $\{x_1, x_3\}$.

Interprétation d'un formule

L'interprétation $[\![p]\!]v$ d'une formule p par rapport à l'affectation v est définie par récurrence sur la structure de p:

$$
\begin{array}{lcl}
[\![x]\!]v & = & v(x) \\
[\![\neg p_1]\!]v & = & \begin{cases} 0 & \text{si } [\![p_1]\!]v = 1 \\ 1 & \text{si } [\![p_1]\!]v = 0 \end{cases} \\
[\![(p_1 \wedge p_2)]\!]v & = & \begin{cases} 0 & \text{si } [\![p_1]\!]v = 0 \text{ ou } [\![p_2]\!]v = 0 \\ 1 & \text{si } [\![p_1]\!]v = 1 \text{ et } [\![p_2]\!]v = 1 \end{cases} \\
[\![(p_1 \vee p_2)]\!]v & = & \begin{cases} 0 & \text{si } [\![p_1]\!]v = 0 \text{ et } [\![p_2]\!]v = 0 \\ 1 & \text{si } [\![p_1]\!]v = 1 \text{ ou } [\![p_2]\!]v = 1 \end{cases}
\end{array}
$$

Interprétation d'un formule

L'interprétation $[\![p]\!]v$ d'une formule p par rapport à l'affectation v est définie par récurrence sur la structure de p:

$$
\begin{array}{lcl}
[\![x]\!]v & = & v(x) \\
[\![\neg p_1]\!]v & = & \begin{cases} 0 & \text{si } [\![p_1]\!]v = 1 \\ 1 & \text{si } [\![p_1]\!]v = 0 \end{cases} \\
[\![(p_1 \wedge p_2)]\!]v & = & \begin{cases} 0 & \text{si } [\![p_1]\!]v = 0 \text{ ou } [\![p_2]\!]v = 0 \\ 1 & \text{si } [\![p_1]\!]v = 1 \text{ et } [\![p_2]\!]v = 1 \end{cases} \\
[\![(p_1 \vee p_2)]\!]v & = & \begin{cases} 0 & \text{si } [\![p_1]\!]v = 0 \text{ et } [\![p_2]\!]v = 0 \\ 1 & \text{si } [\![p_1]\!]v = 1 \text{ ou } [\![p_2]\!]v = 1 \end{cases}
\end{array}
$$

Interprétation d'un formule

L'interprétation $[\![p]\!]v$ d'une formule p par rapport à l'affectation v est définie par récurrence sur la structure de p:

$$
\begin{array}{lcl}
[\![x]\!]v & = & v(x) \\
[\![\neg p_1]\!]v & = & \begin{cases} 0 & \text{si } [\![p_1]\!]v = 1 \\ 1 & \text{si } [\![p_1]\!]v = 0 \end{cases} \\
[\![(p_1 \wedge p_2)]\!]v & = & \begin{cases} 0 & \text{si } [\![p_1]\!]v = 0 \text{ ou } [\![p_2]\!]v = 0 \\ 1 & \text{si } [\![p_1]\!]v = 1 \text{ et } [\![p_2]\!]v = 1 \end{cases} \\
[\![(p_1 \vee p_2)]\!]v & = & \begin{cases} 0 & \text{si } [\![p_1]\!]v = 0 \text{ et } [\![p_2]\!]v = 0 \\ 1 & \text{si } [\![p_1]\!]v = 1 \text{ ou } [\![p_2]\!]v = 1 \end{cases}
\end{array}
$$

Interprétation d'un formule

L'interprétation $[\![p]\!]v$ d'une formule p par rapport à l'affectation v est définie par récurrence sur la structure de p:

$$
\begin{array}{lcl}
[\![x]\!]v & = & v(x) \\
[\![\neg p_1]\!]v & = & \begin{cases} 0 & \text{si } [\![p_1]\!]v = 1 \\ 1 & \text{si } [\![p_1]\!]v = 0 \end{cases} \\
[\![(p_1 \wedge p_2)]\!]v & = & \begin{cases} 0 & \text{si } [\![p_1]\!]v = 0 \text{ ou } [\![p_2]\!]v = 0 \\ 1 & \text{si } [\![p_1]\!]v = 1 \text{ et } [\![p_2]\!]v = 1 \end{cases} \\
[\![(p_1 \vee p_2)]\!]v & = & \begin{cases} 0 & \text{si } [\![p_1]\!]v = 0 \text{ et } [\![p_2]\!]v = 0 \\ 1 & \text{si } [\![p_1]\!]v = 1 \text{ ou } [\![p_2]\!]v = 1 \end{cases}
\end{array}
$$

Exemples

Pour l'affectation $v_1 = [y \mapsto 1]$ on a que

- $[\![x]\!]v_1 = 0$ (car $v_1(x) = 0$)
- $[\![y]\!]v_1 = 1$ (car $v_1(y) = 1$)
- Donc : $[\![(x \wedge y)]\!]v_1 = 0$
- et $[\![(x \vee y)]\!]v_1 = 1$.

Exemples

Pour l'affectation $v_1 = [y \mapsto 1]$ on a que

- $[\![x]\!]v_1 = 0$ (car $v_1(x) = 0$)
- $[\![y]\!]v_1 = 1$ (car $v_1(y) = 1$)
- Donc : $[\![(x \wedge y)]\!]v_1 = 0$
- et $[\![(x \vee y)]\!]v_1 = 1$.

Exemples

Pour l'affectation $v_1 = [y \mapsto 1]$ on a que

- $[\![x]\!]v_1 = 0$ (car $v_1(x) = 0$)
- $[\![y]\!]v_1 = 1$ (car $v_1(y) = 1$)
- Donc : $[\![(x \wedge y)]\!]v_1 = 0$
- et $[\![(x \vee y)]\!]v_1 = 1$.

Exemples

Pour l'affectation $v_1 = [y \mapsto 1]$ on a que

- $[\![x]\!]v_1 = 0$ (car $v_1(x) = 0$)
- $[\![y]\!]v_1 = 1$ (car $v_1(y) = 1$)
- Donc : $[\![(x \wedge y)]\!]v_1 = 0$
- et $[\![(x \vee y)]\!]v_1 = 1$.

Stratégie d'interprétation

Théorème : Soient p et q des formules propositionnelles et v une affectation, alors

$$[\![(p \wedge q)]\!]v = \begin{cases} 0 & \text{si } [\![p]\!]v = 0 \\ [\![q]\!]v & \text{si } [\![p]\!]v = 1 \end{cases}$$

$$[\![(p \vee q)]\!]v = \begin{cases} 1 & \text{si } [\![p]\!]v = 1 \\ [\![q]\!]v & \text{si } [\![p]\!]v = 0 \end{cases}$$

Démonstration

Nous démontrons seulement le premier des deux énoncés; le second se montre de façon analogue. Il y deux cas, selon la valeur de $[\![p]\!]v$:

Cas $[\![p]\!]v = 0$: Nous avons à montrer que dans ce cas $[\![(p \wedge q)]\!]v = 0$. C'est une conséquence immédiate de la définition.

Cas $[\![p]\!]v = 1$: Nous avons à montrer que dans ce cas $[\![(p \wedge q)]\!]v = [\![q]\!]v$. Il y a deux cas, selon la valeur de $[\![q]\!]v$:

- Cas $[\![q]\!]v = 0$: Nous avons $[\![(p \wedge q)]\!]v = 0 = [\![q]\!]v$
- Cas $[\![q]\!]v = 1$: Nous avons $[\![(p \wedge q)]\!]v = 1 = [\![q]\!]v$

Démonstration

Nous démontrons seulement le premier des deux énoncés; le second se montre de façon analogue. Il y deux cas, selon la valeur de $[\![p]\!]v$:

Cas $[\![p]\!]v = 0$: Nous avons à montrer que dans ce cas $[\![(p \wedge q)]\!]v = 0$. C'est une conséquence immédiate de la définition.

Cas $[\![p]\!]v = 1$: Nous avons à montrer que dans ce cas $[\![(p \wedge q)]\!]v = [\![q]\!]v$. Il y a deux cas, selon la valeur de $[\![q]\!]v$:

Cas $[\![q]\!]v = 0$: Nous avons $[\![(p \wedge q)]\!]v = 0 = [\![q]\!]v$

Cas $[\![q]\!]v = 1$: Nous avons $[\![(p \wedge q)]\!]v = 1 = [\![q]\!]v$

Démonstration

Nous démontrons seulement le premier des deux énoncés; le second se montre de façon analogue. Il y deux cas, selon la valeur de $[\![p]\!]v$:

Cas $[\![p]\!]v = 0$: Nous avons à montrer que dans ce cas $[\![(p \wedge q)]\!]v = 0$. C'est une conséquence immédiate de la définition.

Cas $[\![p]\!]v = 1$: Nous avons à montrer que dans ce cas $[\![(p \wedge q)]\!]v = [\![q]\!]v$. Il y a deux cas, selon la valeur de $[\![q]\!]v$:

- **Cas $[\![q]\!]v = 0$:** Nous avons $[\![(p \wedge q)]\!]v = 0 = [\![q]\!]v$
- **Cas $[\![q]\!]v = 1$:** Nous avons $[\![(p \wedge q)]\!]v = 1 = [\![q]\!]v$

Démonstration

Nous démontrons seulement le premier des deux énoncés; le second se montre de façon analogue. Il y deux cas, selon la valeur de $[\![p]\!]v$:

Cas $[\![p]\!]v = 0$: Nous avons à montrer que dans ce cas $[\![(p \wedge q)]\!]v = 0$. C'est une conséquence immédiate de la définition.

Cas $[\![p]\!]v = 1$: Nous avons à montrer que dans ce cas $[\![(p \wedge q)]\!]v = [\![q]\!]v$. Il y a deux cas, selon la valeur de $[\![q]\!]v$:

- Cas $[\![q]\!]v = 0$: Nous avons $[\![(p \wedge q)]\!]v = 0 = [\![q]\!]v$
- Cas $[\![q]\!]v = 1$: Nous avons $[\![(p \wedge q)]\!]v = 1 = [\![q]\!]v$

L'intérêt de ce théorème

- Pour évaluer $[\![(p \wedge q)]\!]v$ on évalue d'abord $[\![p]\!]v$, et selon le résultat obtenu il se peut qu'il ne soit plus nécessaire d'évaluer $[\![q]\!]v$, ce qui est bon à savoir quand q est une très grande expression.
- On aurait pu donner une variante du théorème dans laquelle on interprète d'abord q au lieu de p, ou encore des variantes avec des critères plus sophistiqués (par exemple: on commence avec l'interprétation de la formule parmi p, q qui est est la plus petite).

L'intérêt de ce théorème

- Pour évaluer $[\![(p \wedge q)]\!]v$ on évalue d'abord $[\![p]\!]v$, et selon le résultat obtenu il se peut qu'il ne soit plus nécessaire d'évaluer $[\![q]\!]v$, ce qui est bon à savoir quand q est une très grande expression.
- On aurait pu donner une variante du théorème dans laquelle on interprète d'abord q au lieu de p, ou encore des variantes avec des critères plus sophistiqués (par exemple: on commence avec l'interprétation de la formule parmi p, q qui est est la plus petite).

Évaluation et interprétation

- En général, une application d'une fonction à des arguments est évaluée.
- Plus spécifiquement, une formule propositionnelle est interprétée par rapport à une affectation.

L'interprétation de p par rapport à v est obtenue par l'évaluation de $[\![p]\!]v$.

Définition

Soit p une formule propositionnelle.

- On écrit $v \models p$ si $[\![p]\!]v = 1$, et on dit « p est vraie par rapport à v ».
- On écrit $v \not\models p$ si $[\![p]\!]v = 0$, et on dit « p est fausse par rapport à v ».
- On dit que p est satisfaisable s'il existe une affectation v telle que $v \models p$.
- On dit que p est falsifiable s'il existe une affectation v telle que $v \not\models p$.
- On écrit $\models p$ si $v \models p$ pour toute affectation v, et on dit que p est valide (ou une *tautologie*).
- On écrit $\not\models p$ si $v \not\models p$ pour toute affectation v, et on dit que p est contradictoire.

Définition

Soit p une formule propositionnelle.

- On écrit $v \models p$ si $[\![p]\!]v = 1$, et on dit « p est vraie par rapport à v ».
- On écrit $v \not\models p$ si $[\![p]\!]v = 0$, et on dit « p est fausse par rapport à v ».
- On dit que p est satisfaisable s'il existe une affectation v telle que $v \models p$.
- On dit que p est falsifiable s'il existe une affectation v telle que $v \not\models p$.
- On écrit $\models p$ si $v \models p$ pour toute affectation v, et on dit que p est valide (ou une *tautologie*).
- On écrit $\not\models p$ si $v \not\models p$ pour toute affectation v, et on dit que p est contradictoire.

Définition

Soit p une formule propositionnelle.

- On écrit $v \models p$ si $[\![p]\!]v = 1$, et on dit « p est vraie par rapport à v ».
- On écrit $v \not\models p$ si $[\![p]\!]v = 0$, et on dit « p est fausse par rapport à v ».
- On dit que p est satisfaisable s'il existe une affectation v telle que $v \models p$.
- On dit que p est falsifiable s'il existe une affectation v telle que $v \not\models p$.
- On écrit $\models p$ si $v \models p$ pour toute affectation v, et on dit que p est valide (ou une *tautologie*).
- On écrit $\not\models p$ si $v \not\models p$ pour toute affectation v, et on dit que p est contradictoire.

Définition

Soit p une formule propositionnelle.

- On écrit $v \models p$ si $[\![p]\!]v = 1$, et on dit « p est vraie par rapport à v ».
- On écrit $v \not\models p$ si $[\![p]\!]v = 0$, et on dit « p est fausse par rapport à v ».
- On dit que p est satisfaisable s'il existe une affectation v telle que $v \models p$.
- On dit que p est falsifiable s'il existe une affectation v telle que $v \not\models p$.
- On écrit $\models p$ si $v \models p$ pour toute affectation v, et on dit que p est valide (ou une *tautologie*).
- On écrit $\not\models p$ si $v \not\models p$ pour toute affectation v, et on dit que p est contradictoire.

Définition

Soit p une formule propositionnelle.

- On écrit $v \models p$ si $[\![p]\!]v = 1$, et on dit « p est vraie par rapport à v ».
- On écrit $v \not\models p$ si $[\![p]\!]v = 0$, et on dit « p est fausse par rapport à v ».
- On dit que p est satisfaisable s'il existe une affectation v telle que $v \models p$.
- On dit que p est falsifiable s'il existe une affectation v telle que $v \not\models p$.
- On écrit $\models p$ si $v \models p$ pour toute affectation v, et on dit que p est valide (ou une *tautologie*).
- On écrit $\not\models p$ si $v \not\models p$ pour toute affectation v, et on dit que p est contradictoire.

Définition

Soit p une formule propositionnelle.

- On écrit $v \models p$ si $[\![p]\!]v = 1$, et on dit « p est vraie par rapport à v ».
- On écrit $v \not\models p$ si $[\![p]\!]v = 0$, et on dit « p est fausse par rapport à v ».
- On dit que p est satisfaisable s'il existe une affectation v telle que $v \models p$.
- On dit que p est falsifiable s'il existe une affectation v telle que $v \not\models p$.
- On écrit $\models p$ si $v \models p$ pour toute affectation v, et on dit que p est valide (ou une *tautologie*).
- On écrit $\not\models p$ si $v \not\models p$ pour toute affectation v, et on dit que p est contradictoire.

Exemples

La formule $(x \vee y)$ est

- satisfaisable (elle est vraie par rapport à $[x \mapsto 1]$)
- falsifiable (elle est fausse par rapport à $[]$)
- pas une tautologie
- pas contradictoire

Exemples

La formule $(x \vee y)$ est

- satisfaisable (elle est vraie par rapport à $[x \mapsto 1]$)
- falsifiable (elle est fausse par rapport à $[]$)
- pas une tautologie
- pas contradictoire

Exemples

La formule $(x \vee y)$ est

- satisfaisable (elle est vraie par rapport à $[x \mapsto 1]$)
- falsifiable (elle est fausse par rapport à $[]$)
- pas une tautologie
- pas contradictoire

Exemples

La formule $(x \vee y)$ est

- satisfaisable (elle est vraie par rapport à $[x \mapsto 1]$)
- falsifiable (elle est fausse par rapport à $[]$)
- pas une tautologie
- pas contradictoire

Exemples

- $(x \vee \neg x)$ est une tautologie.
- $(x \wedge \neg x)$ est contradictoire.

Exemples

- $(x \vee \neg x)$ est une tautologie.
- $(x \wedge \neg x)$ est contradictoire.

Notions de sémantique

Ne pas confondre les notations:

- Une formule est vraie ou fausse toujours par rapport à une affectation.
- Une formule peut être satisfaisable ou falsifiable tout court. Il n'y a pas de « satisfaisable par une affectation ».
- Une formule peut être valide ou contradictoire.

Proposition :

Une formule p est valide si et seulement si $\neg p$ n'est pas satisfaisable.

Démonstration :

On a la chaîne d'équivalences suivante :

	p est valide
ssi	$v \models p$ pour toute affectation v
ssi	$[\![p]\!]v = 1$ pour toute affectation v
ssi	$[\![\neg p]\!]v = 0$ pour toute affectation v
ssi	$v \models \neg p$ pour aucune affectation v
ssi	$\neg p$ n'est pas satisfaisable

Proposition :

Une formule p est valide si et seulement si $\neg p$ n'est pas satisfaisable.

Démonstration :

On a la chaîne d'équivalences suivante :

	p est valide
ssi	$v \models p$ pour toute affectation v
ssi	$[\![p]\!]v = 1$ pour toute affectation v
ssi	$[\![\neg p]\!]v = 0$ pour toute affectation v
ssi	$v \models \neg p$ pour aucune affectation v
ssi	$\neg p$ n'est pas satisfaisable

Proposition :

Une formule p est valide si et seulement si $\neg p$ n'est pas satisfaisable.

Démonstration :

On a la chaîne d'équivalences suivante :

	p est valide
ssi	$v \models p$ pour toute affectation v
ssi	$[\![p]\!]v = 1$ pour toute affectation v
ssi	$[\![\neg p]\!]v = 0$ pour toute affectation v
ssi	$v \models \neg p$ pour aucune affectation v
ssi	$\neg p$ n'est pas satisfaisable

Proposition :

Une formule p est valide si et seulement si $\neg p$ n'est pas satisfaisable.

Démonstration :

On a la chaîne d'équivalences suivante :

	p est valide
ssi	$v \models p$ pour toute affectation v
ssi	$[\![p]\!]v = 1$ pour toute affectation v
ssi	$[\![\neg p]\!]v = 0$ pour toute affectation v
ssi	$v \models \neg p$ pour aucune affectation v
ssi	$\neg p$ n'est pas satisfaisable

Proposition :

Une formule p est valide si et seulement si $\neg p$ n'est pas satisfaisable.

Démonstration :

On a la chaîne d'équivalences suivante :

	p est valide
ssi	$v \models p$ pour toute affectation v
ssi	$[\![p]\!]v = 1$ pour toute affectation v
ssi	$[\![\neg p]\!]v = 0$ pour toute affectation v
ssi	$v \models \neg p$ pour aucune affectation v
ssi	$\neg p$ n'est pas satisfaisable

Proposition :

Une formule p est valide si et seulement si $\neg p$ n'est pas satisfaisable.

Démonstration :

On a la chaîne d'équivalences suivante :

	p est valide
ssi	$v \models p$ pour toute affectation v
ssi	$[\![p]\!]v = 1$ pour toute affectation v
ssi	$[\![\neg p]\!]v = 0$ pour toute affectation v
ssi	$v \models \neg p$ pour aucune affectation v
ssi	$\neg p$ n'est pas satisfaisable

Proposition :

Une formule p est valide si et seulement si $\neg p$ n'est pas satisfaisable.

Démonstration :

On a la chaîne d'équivalences suivante :

	p est valide
ssi	$v \models p$ pour toute affectation v
ssi	$[\![p]\!]v = 1$ pour toute affectation v
ssi	$[\![\neg p]\!]v = 0$ pour toute affectation v
ssi	$v \models \neg p$ pour aucune affectation v
ssi	$\neg p$ n'est pas satisfaisable

Proposition : Une formule p est contradictoire si et seulement si $\neg p$ est valide.

(Exercice !)

Décider validité etc.

- Pour savoir si une formule propositionnelle donnée est satisfaisable ou valide il faut donc en principe évaluer la formule sur toutes les affectations possibles.
- Problème : il y a un nombre infini d'affectations possibles car il y a un nombre infini de variables propositionnelles !
- Heureusement, le théorème suivant dit que seulement les variables qui aparaissent dans les formules sont pertinentes.

Décider validité etc.

- Pour savoir si une formule propositionnelle donnée est satisfaisable ou valide il faut donc en principe évaluer la formule sur toutes les affectations possibles.
- Problème : il y a un nombre infini d'affectations possibles car il y a un nombre infini de variables propositionnelles !
- Heureusement, le théorème suivant dit que seulement les variables qui aparaissent dans les formules sont pertinentes.

Décider validité etc.

- Pour savoir si une formule propositionnelle donnée est satisfaisable ou valide il faut donc en principe évaluer la formule sur toutes les affectations possibles.
- Problème : il y a un nombre infini d'affectations possibles car il y a un nombre infini de variables propositionnelles !
- Heureusement, le théorème suivant dit que seulement les variables qui aparaissent dans les formules sont pertinentes.

D'abord une proposition

Proposition : Soit p une formule propositionnelle et v_1, v_2 des affectations telles que $v_1(x) = v_2(x)$ pour toute variable $x \in \mathcal{V}(p)$. Alors $[\![p]\!]v_1 = [\![p]\!]v_2$.

Exercice (sera fait en TD) !

Le théorème de coïncidence

Théorème : Une formule p est

1. satisfaisable si et seulement s'il existe une affectation v telle que $supp(v) \subseteq \mathcal{V}(p)$ et $v \models p$.
2. valide si et seulement si $v \models p$ pour toute affectation v avec $supp(v) \subseteq \mathcal{V}(p)$.

Démonstration du premier énoncé

Si $v \models p$ avec $supp(v) \subseteq \mathcal{V}(p)$ alors p est, par définition satisfaisable.

Si p est satisfaisable il y a une affectation w telle que $w \models p$. Nous construisons une nouvelle affectation v comme suit:

$$v(x) = \begin{cases} w(x) & \text{si } x \in \mathcal{V}(p) \\ 0 & \text{si } x \notin \mathcal{V}(p) \end{cases}$$

On a que $supp(v) \subseteq \mathcal{V}(p)$, et $v \models p$ par la proposition précédente.

Démonstration du premier énoncé

Si $v \models p$ avec $supp(v) \subseteq \mathcal{V}(p)$ alors p est, par définition satisfaisable.

Si p est satisfaisable il y a une affectation w telle que $w \models p$. Nous construisons une nouvelle affectation v comme suit:

$$v(x) = \begin{cases} w(x) & \text{si } x \in \mathcal{V}(p) \\ 0 & \text{si } x \notin \mathcal{V}(p) \end{cases}$$

On a que $supp(v) \subseteq \mathcal{V}(p)$, et $v \models p$ par la proposition précédente.

Démonstration du premier énoncé

Si $v \models p$ avec $supp(v) \subseteq \mathcal{V}(p)$ alors p est, par définition satisfaisable.

Si p est satisfaisable il y a une affectation w telle que $w \models p$. Nous construisons une nouvelle affectation v comme suit:

$$v(x) = \begin{cases} w(x) & \text{si } x \in \mathcal{V}(p) \\ 0 & \text{si } x \notin \mathcal{V}(p) \end{cases}$$

On a que $supp(v) \subseteq \mathcal{V}(p)$, et $v \models p$ par la proposition précédente.

Une méthode pour décider satisfaisabilité

1. Calculer $\mathcal{V}(p)$
2. Engendrer l'ensemble A des affectations v avec $supp(v) \subseteq \mathcal{V}(p)$
3. Évaluer $[\![p]\!]v$ pour toute affectation $v \in A$. Dès qu'on tombe sur un v tel que $[\![p]\!]v = 1$ on sait que p est satisfaisable, si on n'en trouve pas alors p n'est pas satisfaisable.

Combien d'affectations est-ce qu'il y a tester, si la formule a n variables ?

Une méthode pour décider satisfaisabilité

1. Calculer $\mathcal{V}(p)$
2. Engendrer l'ensemble A des affectations v avec $supp(v) \subseteq \mathcal{V}(p)$
3. Évaluer $[\![p]\!]v$ pour toute affectation $v \in A$. Dès qu'on tombe sur un v tel que $[\![p]\!]v = 1$ on sait que p est satisfaisable, si on n'en trouve pas alors p n'est pas satisfaisable.

Combien d'affectations est-ce qu'il y a tester, si la formule a n variables ?

Une méthode pour décider satisfaisabilité

1. Calculer $\mathcal{V}(p)$
2. Engendrer l'ensemble A des affectations v avec $supp(v) \subseteq \mathcal{V}(p)$
3. Évaluer $[\![p]\!]v$ pour toute affectation $v \in A$. Dès qu'on tombe sur un v tel que $[\![p]\!]v = 1$ on sait que p est satisfaisable, si on n'en trouve pas alors p n'est pas satisfaisable.

Combien d'affectations est-ce qu'il y a tester, si la formule a n variables ?

Une méthode pour décider satisfaisabilité

1. Calculer $\mathcal{V}(p)$
2. Engendrer l'ensemble A des affectations v avec $supp(v) \subseteq \mathcal{V}(p)$
3. Évaluer $[\![p]\!]v$ pour toute affectation $v \in A$. Dès qu'on tombe sur un v tel que $[\![p]\!]v = 1$ on sait que p est satisfaisable, si on n'en trouve pas alors p n'est pas satisfaisable.

Combien d'affectations est-ce qu'il y a tester, si la formule a n variables ?

Décider validité d'une formule

1. Calculer $\mathcal{V}(p)$
2. Engendrer l'ensemble A des affectations v avec $supp(v) \subseteq \mathcal{V}(p)$
3. Évaluer $[\![p]\!]v$ pour tout $v \in A$. Dès qu'on tombe sur un v tel que $[\![p]\!]v = 0$ on sait que p n'est pas valide, si on n'en trouve pas alors p est valide.

Décider validité d'une formule

1. Calculer $\mathcal{V}(p)$
2. Engendrer l'ensemble A des affectations v avec $supp(v) \subseteq \mathcal{V}(p)$
3. Évaluer $[\![p]\!]v$ pour tout $v \in A$. Dès qu'on tombe sur un v tel que $[\![p]\!]v = 0$ on sait que p n'est pas valide, si on n'en trouve pas alors p est valide.

Décider validité d'une formule

1. Calculer $\mathcal{V}(p)$
2. Engendrer l'ensemble A des affectations v avec $supp(v) \subseteq \mathcal{V}(p)$
3. Évaluer $[\![p]\!]v$ pour tout $v \in A$. Dès qu'on tombe sur un v tel que $[\![p]\!]v = 0$ on sait que p n'est pas valide, si on n'en trouve pas alors p est valide.

Tables de vérité

x_1	x_2	x_3	$\neg x_2$	$(x_1 \wedge x_2)$	$(x_3 \wedge \neg x_2)$	$((x_1 \wedge x_2) \vee (x_3 \wedge \neg x_2))$
(1)	(2)	(3)	(4)	(5)	(6)	(7)
			$\neg$(2)	(1) $\wedge$ (2)	(3) $\wedge$ (4)	(5) $\vee$ (6)
0	0	0	1	0	0	0
1	0	0	1	0	0	0
0	1	0	0	0	0	0
1	1	0	0	1	0	1
0	0	1	1	0	1	1
1	0	1	1	0	1	1
0	1	1	0	0	0	0
1	1	1	0	1	0	1

Tables de vérité

x_1	x_2	x_3	$\neg x_2$	$(x_1 \wedge x_2)$	$(x_3 \wedge \neg x_2)$	$((x_1 \wedge x_2) \vee (x_3 \wedge \neg x_2))$
(1)	(2)	(3)	(4)	(5)	(6)	(7)
			$\neg$(2)	(1) $\wedge$ (2)	(3) $\wedge$ (4)	(5) $\vee$ (6)
0	0	0	1	0	0	0
1	0	0	1	0	0	0
0	1	0	0	0	0	0
1	1	0	0	1	0	1
0	0	1	1	0	1	1
1	0	1	1	0	1	1
0	1	1	0	0	0	0
1	1	1	0	1	0	1

Tables de vérité

x_1	x_2	x_3	$\neg x_2$	$(x_1 \wedge x_2)$	$(x_3 \wedge \neg x_2)$	$((x_1 \wedge x_2) \vee (x_3 \wedge \neg x_2))$
(1)	(2)	(3)	(4)	(5)	(6)	(7)
			$\neg(2)$	$(1) \wedge (2)$	$(3) \wedge (4)$	$(5) \vee (6)$
0	0	0	1	0	0	0
1	0	0	1	0	0	0
0	1	0	0	0	0	0
1	1	0	0	1	0	1
0	0	1	1	0	1	1
1	0	1	1	0	1	1
0	1	1	0	0	0	0
1	1	1	0	1	0	1

Tables de vérité

x_1	x_2	x_3	$\neg x_2$	$(x_1 \wedge x_2)$	$(x_3 \wedge \neg x_2)$	$((x_1 \wedge x_2) \vee (x_3 \wedge \neg x_2))$
(1)	(2)	(3)	(4)	(5)	(6)	(7)
			$\neg$(2)	(1) $\wedge$ (2)	(3) $\wedge$ (4)	(5) $\vee$ (6)
0	0	0	1	0	0	0
1	0	0	1	0	0	0
0	1	0	0	0	0	0
1	1	0	0	1	0	1
0	0	1	1	0	1	1
1	0	1	1	0	1	1
0	1	1	0	0	0	0
1	1	1	0	1	0	1

Tables de vérité

x_1	x_2	x_3	$\neg x_2$	$(x_1 \wedge x_2)$	$(x_3 \wedge \neg x_2)$	$((x_1 \wedge x_2) \vee (x_3 \wedge \neg x_2))$
(1)	(2)	(3)	(4)	(5)	(6)	(7)
			$\neg(2)$	$(1) \wedge (2)$	$(3) \wedge (4)$	$(5) \vee (6)$
0	0	0	1	0	0	0
1	0	0	1	0	0	0
0	1	0	0	0	0	0
1	1	0	0	1	0	1
0	0	1	1	0	1	1
1	0	1	1	0	1	1
0	1	1	0	0	0	0
1	1	1	0	1	0	1

Tables de vérité

x_1	x_2	x_3	$\neg x_2$	$(x_1 \wedge x_2)$	$(x_3 \wedge \neg x_2)$	$((x_1 \wedge x_2) \vee (x_3 \wedge \neg x_2))$
(1)	(2)	(3)	(4)	(5)	(6)	(7)
			$\neg$(2)	(1) $\wedge$ (2)	(3) $\wedge$ (4)	(5) $\vee$ (6)
0	0	0	1	0	0	0
1	0	0	1	0	0	0
0	1	0	0	0	0	0
1	1	0	0	1	0	1
0	0	1	1	0	1	1
1	0	1	1	0	1	1
0	1	1	0	0	0	0
1	1	1	0	1	0	1

Tables de vérité

x_1	x_2	x_3	$\neg x_2$	$(x_1 \wedge x_2)$	$(x_3 \wedge \neg x_2)$	$((x_1 \wedge x_2) \vee (x_3 \wedge \neg x_2))$
(1)	(2)	(3)	(4)	(5)	(6)	(7)
			$\neg(2)$	$(1) \wedge (2)$	$(3) \wedge (4)$	$(5) \vee (6)$
0	0	0	1	0	0	0
1	0	0	1	0	0	0
0	1	0	0	0	0	0
1	1	0	0	1	0	1
0	0	1	1	0	1	1
1	0	1	1	0	1	1
0	1	1	0	0	0	0
1	1	1	0	1	0	1

Tables de vérité

x_1	x_2	x_3	$\neg x_2$	$(x_1 \wedge x_2)$	$(x_3 \wedge \neg x_2)$	$((x_1 \wedge x_2) \vee (x_3 \wedge \neg x_2))$
(1)	(2)	(3)	(4)	(5)	(6)	(7)
			$\neg(2)$	$(1) \wedge (2)$	$(3) \wedge (4)$	$(5) \vee (6)$
0	0	0	1	0	0	0
1	0	0	1	0	0	0
0	1	0	0	0	0	0
1	1	0	0	1	0	1
0	0	1	1	0	1	1
1	0	1	1	0	1	1
0	1	1	0	0	0	0
1	1	1	0	1	0	1

Tables de vérité

x_1	x_2	x_3	$\neg x_2$	$(x_1 \wedge x_2)$	$(x_3 \wedge \neg x_2)$	$((x_1 \wedge x_2) \vee (x_3 \wedge \neg x_2))$
(1)	(2)	(3)	(4)	(5)	(6)	(7)
			$\neg$(2)	(1) $\wedge$ (2)	(3) $\wedge$ (4)	(5) $\vee$ (6)
0	0	0	1	0	0	0
1	0	0	1	0	0	0
0	1	0	0	0	0	0
1	1	0	0	1	0	1
0	0	1	1	0	1	1
1	0	1	1	0	1	1
0	1	1	0	0	0	0
1	1	1	0	1	0	1

Efficacité de notre algorithme

- Si la formule a n variables : 2^n affectations possibles à essayer (dans le pire des cas)
- Temps d'exécution exponentiel
- Acceptable seulement pour des petites formules.
- Comment faire pour des formules avec 10.000 variables ? Voir dans quelques semaines !

Efficacité de notre algorithme

- Si la formule a n variables : 2^n affectations possibles à essayer (dans le pire des cas)
- Temps d'exécution exponentiel
- Acceptable seulement pour des petites formules.
- Comment faire pour des formules avec 10.000 variables ? Voir dans quelques semaines !

Efficacité de notre algorithme

- Si la formule a n variables : 2^n affectations possibles à essayer (dans le pire des cas)
- Temps d'exécution exponentiel
- Acceptable seulement pour des petites formules.
- Comment faire pour des formules avec 10.000 variables ? Voir dans quelques semaines !

Efficacité de notre algorithme

- Si la formule a n variables : 2^n affectations possibles à essayer (dans le pire des cas)
- Temps d'exécution exponentiel
- Acceptable seulement pour des petites formules.
- Comment faire pour des formules avec 10.000 variables ? Voir dans quelques semaines !

Raccourcis

- On a le droit d'enchaîner des applications de l'opérateur $\wedge$:

$$(p_1 \wedge p_2 \wedge \ldots \wedge p_n)$$

ainsi que de l'opérateur $\vee$:

$$(p_1 \vee p_2 \vee \ldots \vee p_n)$$

- On se permet d'omettre la paire de parenthèses qui est autour de la formule entière.

Raccourcis

- On a le droit d'enchaîner des applications de l'opérateur $\wedge$:

$$(p_1 \wedge p_2 \wedge \ldots \wedge p_n)$$

  ainsi que de l'opérateur $\vee$:

$$(p_1 \vee p_2 \vee \ldots \vee p_n)$$

- On se permet d'omettre la paire de parenthèses qui est autour de la formule entière.

Récupérer la syntaxe stricte

- Remplacer

$$(p_1 \wedge p_2 \wedge p_3 \wedge \ldots \wedge p_n)$$

par

$$(p_1 \wedge (p_2 \wedge (p_3 \ldots \wedge p_n) \ldots))$$

et pareil pour les chaînes $\vee$.

- Mettre une paire de parenthèses extérieures autour de la formule si nécessaire

Récupérer la syntaxe stricte

- Remplacer
$$(p_1 \wedge p_2 \wedge p_3 \wedge \ldots \wedge p_n)$$
par
$$(p_1 \wedge (p_2 \wedge (p_3 \ldots \wedge p_n) \ldots))$$
et pareil pour les chaînes $\vee$.
- Mettre une paire de parenthèses extérieures autour de la formule si nécessaire

Exemple

$$(x_1 \wedge x_2 \wedge x_3) \vee y_1 \vee (z_1 \wedge z_2 \wedge z_3 \wedge z_4)$$

s'écrit en syntaxe stricte comme

$$\Big((x_1 \wedge (x_2 \wedge x_3)) \vee \Big(y_1 \vee (z_1 \wedge (z_2 \wedge (z_3 \wedge z_4)))\Big)\Big)$$

Exemple

$$(x_1 \wedge x_2 \wedge x_3) \vee y_1 \vee (z_1 \wedge z_2 \wedge z_3 \wedge z_4)$$

s'écrit en syntaxe stricte comme

$$\Big((x_1 \wedge (x_2 \wedge x_3)) \vee \Big(y_1 \vee (z_1 \wedge (z_2 \wedge (z_3 \wedge z_4)))\Big)\Big)$$

Syntaxe stricte ou raccourcie ?

- On autorise la syntaxe raccourcie dans les exemples.
- Par contre, quand on vous demande de démontrer une propriété de la syntaxe (comme: $|w|_(= |w|_)$ pour toute $w \in Form$) c'est toujours la syntaxe stricte !

Syntaxe stricte ou raccourcie ?

- On autorise la syntaxe raccourcie dans les exemples.
- Par contre, quand on vous demande de démontrer une propriété de la syntaxe (comme: $|w|_{(} = |w|_{)}$ pour toute $w \in Form$) c'est toujours la syntaxe stricte !