Chapitre 3 : Logique

Option Informatique – MP

Lycée Chateaubriand

Table des matières

Définitions

On se donne :

- un ensemble E,
- une partie B de E (la base)
- des constructeurs qui sont des fonctions f définie d'une partie de E^k dans E (ont dit qu'ils sont d'arité k).

On veut construire la partie de E de tout ce que l'on obtient en utilisant les éléments de la base et en appliquant les constructeurs (éventuellement plusieurs fois).

Definition (Stabilité)

Soit Ω un ensemble de constructeurs et X une partie de E, on dit que X est stable par Ω si pour tout constructeur f d'arité k de Ω et tout $(x_1, \ldots, x_k) \in X^k$, $f(x_1, \ldots, x_k) \in X$ (s'il existe)

On peut alors définir la fermeture inductive.

Definition (Fermeture inductive)

Avec les notations précédentes, on appelle **fermeture inductive** de B par Ω où Ω est un ensemble de constructeurs, le plus petit ensemble de E stable par Ω contenant B.

Il y a deux manières de voir la fermeture inductive :

- Celle des mathématiciens (dite aussi descendante) : la fermeture inductive est l'intersection de tous les ensembles contenant B stables par Ω (il y a au moins E en entier)

Fermeture inductive

Il y a deux manières de voir la fermeture inductive :

- Celle des mathématiciens (dite aussi descendante) : la fermeture inductive est l'intersection de tous les ensembles contenant B stables par Ω (il y a au moins E en entier)
- Celle des informaticiens (dite aussi ascendante) : On note $B_0 = B$. On construit alors $\Omega(B_0)$ l'ensemble des images par un élément de Ω des éléments de B_0 puis on pose $B_1 = B_0 \cup \Omega(B_0)$. On procède ainsi de suite :

$$\forall i \in \mathbb{N}, B_{i+1} = B_i \cup \Omega(B_i).$$

Pour finir la fermeture inductive est $\bigcup_{i \in \mathbb{N}} B_i$.

Un exemple

Les **expressions arithmétiques** se construisent de la sorte. On prend :

- L'ensemble E est l'ensemble de tous les mots s'écrivant avec des caractères ASCII .
- La base $B = \{a, b, c\}$ est l'ensemble des trois lettres qui apparaissent dans nos expressions.
- Les constructeurs sont alors $+$, $\times$, $-$ et $/$ qui sont tous les quatre d'arité 2.

L'ensemble des expressions arithmétiques est alors la fermeture inductive.

Par exemple

$$(a + b)/(a - c)$$

est obtenue en prenant la division des expressions $(a + b)$ et $(a - b)$.

Un exemple

Dans ce cas, que vaut B_1 et B_2 ?

Un exemple

Dans ce cas, que vaut B_1 et B_2 ?

L'ensemble B_1 est l'ensemble des expressions qui s'écrivent avec au plus un opérateur.

Les éléments de l'ensemble B_2 qui ne sont pas déjà dans B_1 sont les expressions qui s'écrivent $(x \text{ op } y)$ où x et y sont dans B_1. Elles peuvent avoir jusqu'à 3 opérateurs.

Représentation par un arbre

Les éléments d'une fermeture inductive se représentent naturellement par un arbre. Un élément est de deux sortes

Représentation par un arbre

Les éléments d'une fermeture inductive se représentent naturellement par un arbre. Un élément est de deux sortes

- un élément de B

Représentation par un arbre

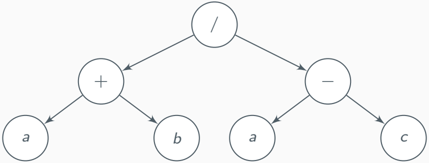
Les éléments d'une fermeture inductive se représentent naturellement par un arbre. Un élément est de deux sortes

- un élément de B
- obtenu par application d'un constructeur.

L'expression $(a + b)/(a - c)$ se représente par :

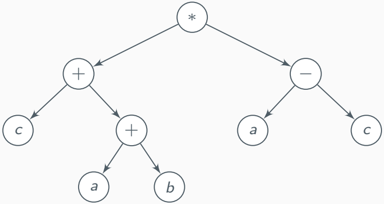
Représentation par un arbre

L'expression $(a+b)/(a-c)$ se représente par :



Exercice

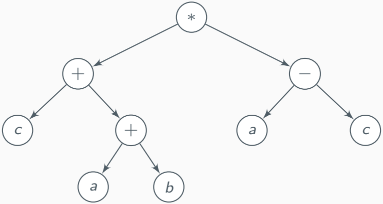
- **Exercice 1 :** Quelle est l'expression représentée par l'arbre suivant ?



- **Exercice 2 :** Dessiner les arbres correspondant aux expressions $((a+b)+c)\times c$ et $(a+(b+c))\times c$

Exercice 1

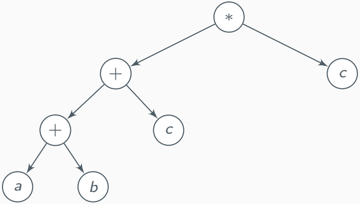
L'arbre



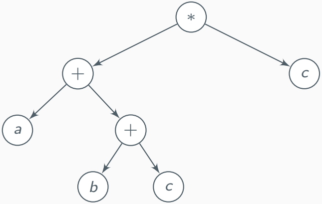
représente

$$(c + (a + b)) * (a - c)$$

Exercice 2



$$\boxed{((a+b)+c)\times c}$$



$$\boxed{(a+(b+c))\times c}$$

Remarque

L'écriture classique des opérations algébriques correspond à la lecture **infixe** de l'arbre (qui est possible car les opérateurs arithmétiques sont d'arité 2 et donc l'arbre est binaire) .

On peut aussi utiliser une écriture **préfixe** des opérateurs.

Dans ce cas l'expression

$$(a+b)/(a-c)$$

s'écrit

Remarque

L'écriture classique des opérations algébriques correspond à la lecture **infixe** de l'arbre (qui est possible car les opérateurs arithmétiques sont d'arité 2 et donc l'arbre est binaire) .

On peut aussi utiliser une écriture **préfixe** des opérateurs.

Dans ce cas l'expression

$$(a+b)/(a-c)$$

s'écrit

$$/(+(a,b),-(a,c)).$$

Remarque

On peut aussi utiliser l'écriture **postfixe** qui s'appelle écriture de **Lukasiewicz** ou **notation polonaise inverse**.

L'intérêt est qu'elle ne nécessite pas de parenthèses. L'expression

$$(a + b)/(a - c)$$

s'écrit alors :

Remarque

On peut aussi utiliser l'écriture **postfixe** qui s'appelle écriture de **Lukasiewicz** ou **notation polonaise inverse**.

L'intérêt est qu'elle ne nécessite pas de parenthèses. L'expression

$$(a+b)/(a-c)$$

s'écrit alors :

$$ab+ac-/$$

Le calcul peut aisément être effectué à l'aide d'une pile.

Attention

Attention

Ne pas confondre la syntaxe (abstraite) et la valeur de l'expression. Par exemple $(a+b)+c$ et $a+(b+c)$ sont deux expressions différentes (d'un point de vue de la syntaxe) par contre elles ont la même valeur par associativité de l'addition. Il est est de même pour $a \times (b+c)$ et $a \times b + a \times c$.

Définition 1 (Formule logique)

Soit $\mathscr{V}$ un ensemble (la plupart du temps fini) des variables, les formules logiques forment la fermeture inductive de $B = \mathscr{V} \cup \{V, F\}$ par l'ensemble formé des constructeurs $\wedge, \vee$ d'arité 2 et de $\neg$ d'arité 1.
De ce fait une formule logique est de la forme suivante :

- *V ou F*

Définition 1 (Formule logique)

Soit $\mathscr{V}$ un ensemble (la plupart du temps fini) des variables, les formules logiques forment la fermeture inductive de $B = \mathscr{V} \cup \{V, F\}$ par l'ensemble formé des constructeurs $\wedge, \vee$ d'arité 2 et de $\neg$ d'arité 1.
De ce fait une formule logique est de la forme suivante :

- *V ou F*
- *Un élément v de $\mathscr{V}$*

Définition 1 (Formule logique)

Soit $\mathscr{V}$ un ensemble (la plupart du temps fini) des variables, les formules logiques forment la fermeture inductive de $B = \mathscr{V} \cup \{V, F\}$ par l'ensemble formé des constructeurs $\wedge, \vee$ d'arité 2 et de $\neg$ d'arité 1.
De ce fait une formule logique est de la forme suivante :

- *V ou F*
- *Un élément v de $\mathscr{V}$*
- *$(p_1 \wedge p_2)$ où p_1 et p_2 sont des formules logiques*

Définition 1 (Formule logique)

Soit $\mathscr{V}$ un ensemble (la plupart du temps fini) des variables, les formules logiques forment la fermeture inductive de $B = \mathscr{V} \cup \{V, F\}$ par l'ensemble formé des constructeurs $\wedge, \vee$ d'arité 2 et de $\neg$ d'arité 1.
De ce fait une formule logique est de la forme suivante :

- *V ou F*
- *Un élément v de $\mathscr{V}$*
- *$(p_1 \wedge p_2)$ où p_1 et p_2 sont des formules logiques*
- *$(p_1 \vee p_2)$ où p_1 et p_2 sont des formules logiques*

Définitions

Définition 1 (Formule logique)

Soit $\mathscr{V}$ un ensemble (la plupart du temps fini) des variables, les formules logiques forment la fermeture inductive de $B = \mathscr{V} \cup \{V, F\}$ par l'ensemble formé des constructeurs $\wedge, \vee$ d'arité 2 et de $\neg$ d'arité 1.
De ce fait une formule logique est de la forme suivante :

- *V ou F*
- *Un élément v de $\mathscr{V}$*
- *$(p_1 \wedge p_2)$ où p_1 et p_2 sont des formules logiques*
- *$(p_1 \vee p_2)$ où p_1 et p_2 sont des formules logiques*
- *$\neg p$ où p est une formule logique.*

Terminologie

- Les éléments V et F s'appellent les constantes.
- Les éléments de $\mathscr{V}$ s'appellent les variables propositionnelles.
- L'opérateur $\wedge$ se lit / se note **et**.
- L'opérateur $\vee$ se lit / se note **ou**.
- L'opérateur $\neg$ se lit / se note **non**.

Remarques

- Il faut faire attention que dans un premier temps on ne donne pas de signification logique à ces expressions. On considère les formules pour elles mêmes. C'est ce que l'on appelle l'aspect *syntaxique*.

Remarques

- Il faut faire attention que dans un premier temps on ne donne pas de signification logique à ces expressions. On considère les formules pour elles mêmes. C'est ce que l'on appelle l'aspect *syntaxique*.
- On s'intéresse dans ce cours à ce que l'on appelle la logique propositionnelle. En particulier nous n'utiliserons pas de quantificateurs qui relèvent de la logique du premier ordre.

Remarques

- Il faut faire attention que dans un premier temps on ne donne pas de signification logique à ces expressions. On considère les formules pour elles mêmes. C'est ce que l'on appelle l'aspect *syntaxique*.
- On s'intéresse dans ce cours à ce que l'on appelle la logique propositionnelle. En particulier nous n'utiliserons pas de quantificateurs qui relèvent de la logique du premier ordre.
- Il existe des variantes. On peut par exemple utiliser aussi les connecteurs $\Rightarrow$ et $\Leftrightarrow$ tous les deux d'arité de 2.

Exemples

- Si on ne considère aucune variables propositionnelles ($\mathscr{V} = \emptyset$), on peut écrire

$$F, (F \wedge F), \neg(F \vee (F \wedge \neg V)).$$

Exemples

- Si on ne considère aucune variables propositionnelles ($\mathscr{V} = \emptyset$), on peut écrire

$$F, (F \wedge F), \neg(F \vee (F \wedge \neg V)).$$

- Si on considère un ensemble $\mathscr{V} = \{v_1, v_2, v_3\}$ on peut écrire

$$v_2, (v_2 \wedge \neg v_1) \text{ ou } ((v_1 \vee F) \vee (v_3 \wedge \neg v_3)).$$

Attention

On rappelle que l'on ne donne pas dans un premier de sens aux formules. Par exemple les formules

ne sont pas les mêmes formules

Attention

On rappelle que l'on ne donne pas dans un premier de sens aux formules. Par exemple les formules

$$\neg(v_1 \vee v_2) \text{ et } (\neg v_1) \wedge (\neg v_2)$$

ne sont pas les mêmes formules

Remarque

On peut éviter de parenthèser les expressions en utilisant les règles de priorités suivantes :

- $\neg$ est prioritaire sur $\wedge$;
- $\wedge$ est prioritaire sur $\vee$.

Cependant, dans un premier temps il est plus aisé de garder certaines parenthèses.

On peut bien évidemment représenter les formules logiques par des arbres **qui ne seront plus binaires** car l'opérateur $\neg$ est d'arité 1. Par exemple

$$((v_1 \vee F) \vee (v_3 \wedge \neg v_2))$$

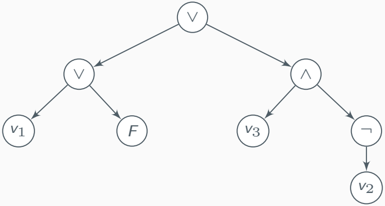
se représente par l'arbre :

Représentation par un arbre

On peut bien évidemment représenter les formules logiques par des arbres **qui ne seront plus binaires** car l'opérateur $\neg$ est d'arité 1. Par exemple

$$((v_1 \vee F) \vee (v_3 \wedge \neg v_2))$$

se représente par l'arbre :



Remarques

- On remarque que les feuilles sont étiquetées avec des variables propositionnelles ou des constantes alors que les nœuds sont étiquetés par des constructeurs.

Remarques

- On remarque que les feuilles sont étiquetées avec des variables propositionnelles ou des constantes alors que les nœuds sont étiquetés par des constructeurs.
- L'écriture classique des formules logiques ne correspondent plus à un parcours usuel de l'arbre. En effet les constructeurs $\wedge$ et $\vee$ s'utilisent de manière **infixe** alors que $\neg$ s'utilise de manière **préfixe**.

Remarques

- On remarque que les feuilles sont étiquetées avec des variables propositionnelles ou des constantes alors que les nœuds sont étiquetés par des constructeurs.
- L'écriture classique des formules logiques ne correspondent plus à un parcours usuel de l'arbre. En effet les constructeurs $\wedge$ et $\vee$ s'utilisent de manière **infixe** alors que $\neg$ s'utilise de manière **préfixe**.
- On peut noter les formules logiques en utilisant la notation **postfixe** de l'arbre. Cela a l'avantage de ne plus nécessiter de parenthèses mais cela rend la formule plus compliquée à lire. La formule ci-dessus serait notée

Remarques

- On remarque que les feuilles sont étiquetées avec des variables propositionnelles ou des constantes alors que les nœuds sont étiquetés par des constructeurs.
- L'écriture classique des formules logiques ne correspondent plus à un parcours usuel de l'arbre. En effet les constructeurs $\wedge$ et $\vee$ s'utilisent de manière **infixe** alors que $\neg$ s'utilise de manière **préfixe**.
- On peut noter les formules logiques en utilisant la notation **postfixe** de l'arbre. Cela a l'avantage de ne plus nécessiter de parenthèses mais cela rend la formule plus compliquée à lire. La formule ci-dessus serait notée

$$v_1\, F \vee v_3\, v_2 \neg \wedge \vee$$

Terminologie

- Soit p une formule logique, on appelle sous-formule de p toute formule dont l'arbre correspond à un sous-arbre de celui de p. Par exemple $(v_3 \wedge (\neg v_2))$ est une sous-formule de $((v_1 \vee F) \vee (v_3 \wedge \neg v_2))$.

- Soit p une formule logique, on appelle sous-formule de p toute formule dont l'arbre correspond à un sous-arbre de celui de p. Par exemple $(v_3 \wedge (\neg v_2))$ est une sous-formule de $((v_1 \vee F) \vee (v_3 \wedge \neg v_2))$.
- On appelle hauteur d'une formule la hauteur de son arbre associé. Par exemple $((v_1 \vee F) \wedge (v_3 \vee \neg v_2))$ est de hauteur 3. Si on se rappelle de la construction ascendante des fermetures inductives, dire qu'une formule est de hauteur k revient à dire qu'elle appartient à $B_k \setminus B_{k-1}$.

La définition des formules logiques comme fermeture inductive permet de les implémenter simplement un CAML via un type récursif.

```
type formule =
  | V
  | F
  | Var of int
  | Non of formule
  | Ou of formule * formule
  | Et of formule * formule;;
```

Maintenant que nous avons construit des formules logiques nous voulons leur « donner un sens ».

Précisément on veut savoir si elle est vraie ou fausse. C'est-à-dire lui associer un booléen.

Par la suite nous noterons $\mathscr{B}$ l'ensemble des booléens. Nous noterons `true`ou 1 pour dire vrai et `false`ou 0 pour dire faux.

Afin de savoir si une formule logique est vraie ou fausse il faut déjà associer des valeurs booléennes aux variables propositionnelles.

Définition 2 (Distribution de vérité)

On appelle distribution de vérité une application μ *de* $\mathscr{V}$ *dans l'ensemble* $\mathscr{B}$ *des booléens. On associe de la sorte une valeur booléenne à chaque variable propositionnelle.*

Exemples

Si on reprend $\mathscr{V} = \{v_1, v_2, v_3\}$ on peut par exemple prendre $\mu(v_1) = \mu(v_2) = 1$ et $\mu(v_3) = 0$.

Exemples

Si on reprend $\mathscr{V} = \{v_1, v_2, v_3\}$ on peut par exemple prendre $\mu(v_1) = \mu(v_2) = 1$ et $\mu(v_3) = 0$.

Exercice : Déterminer le nombre de distribution de vérité.

Exemples

Exercice : Déterminer le nombre de distribution de vérité.

L'ensemble des distributions de vérité est $\mathscr{B}^{\mathscr{V}}$ de cardinal $2^{\#\mathscr{V}}$.

Exemples

Exercice : Déterminer le nombre de distribution de vérité.

L'ensemble des distributions de vérité est $\mathscr{B}^{\mathscr{V}}$ de cardinal $2^{\#\mathscr{V}}$.

Le nombre de distributions de vérité croit **exponentiellement** avec le nombre de variables propositionnelles. Nous en reparlerons à la fin du cours.

Évaluation

On veut maintenant évaluer une formule logique, ce qui revient à associer un booléen à notre formule. On se base encore sur la structure des formules logiques.

Définition 3

Soit μ une distribution de vérité, on construit une fonction d'évaluation $\mathscr{E}_\mu$ (ou juste $\mathscr{E}$) qui va de l'ensemble des formules logique dans $\mathscr{B}$. Elle est définie (par induction) par :

Définition 3

Soit μ une distribution de vérité, on construit une fonction d'évaluation $\mathscr{E}_\mu$ (ou juste $\mathscr{E}$) qui va de l'ensemble des formules logique dans $\mathscr{B}$. Elle est définie (par induction) par :

- $\mathscr{E}(V) = 1$ *et* $\mathscr{E}(F) = 0$.

Définition 3

Soit μ une distribution de vérité, on construit une fonction d'évaluation $\mathscr{E}_\mu$ (ou juste $\mathscr{E}$) qui va de l'ensemble des formules logique dans $\mathscr{B}$. Elle est définie (par induction) par :

- $\mathscr{E}(V) = 1$ *et* $\mathscr{E}(F) = 0$.
- *Si v est une variable propositionnelle,* $\mathscr{E}(v) =$

Définition 3

Soit μ une distribution de vérité, on construit une fonction d'évaluation $\mathscr{E}_\mu$ (ou juste $\mathscr{E}$) qui va de l'ensemble des formules logique dans $\mathscr{B}$. Elle est définie (par induction) par :

- $\mathscr{E}(V) = 1$ *et* $\mathscr{E}(F) = 0$.
- *Si v est une variable propositionnelle,* $\mathscr{E}(v) = \mu(v)$.

Définition 3

Soit μ une distribution de vérité, on construit une fonction d'évaluation $\mathscr{E}_\mu$ (ou juste $\mathscr{E}$) qui va de l'ensemble des formules logique dans $\mathscr{B}$. Elle est définie (par induction) par :

- $\mathscr{E}(V) = 1$ *et* $\mathscr{E}(F) = 0$.
- *Si v est une variable propositionnelle,* $\mathscr{E}(v) = \mu(v)$.
- *Si* p_1 *et* p_2 *sont des formules* $\mathscr{E}(p_1 \wedge p_2) =$

Évaluation

Définition 3

Soit μ une distribution de vérité, on construit une fonction d'évaluation $\mathscr{E}_\mu$ (ou juste $\mathscr{E}$) qui va de l'ensemble des formules logique dans $\mathscr{B}$. Elle est définie (par induction) par :

- $\mathscr{E}(V) = 1$ *et* $\mathscr{E}(F) = 0$.
- *Si v est une variable propositionnelle,* $\mathscr{E}(v) = \mu(v)$.
- *Si* p_1 *et* p_2 *sont des formules* $\mathscr{E}(p_1 \wedge p_2) = \mathscr{E}(p_1) \times \mathscr{E}(p_2)$. *Cela signifie que* $\mathscr{E}(p_1 \wedge p_2)$ *ne vaut* 1 *que si* $\mathscr{E}(p_1)$ *et* $\mathscr{E}(p_2)$ *valent* 1.

Définition 3

Soit μ une distribution de vérité, on construit une fonction d'évaluation $\mathscr{E}_\mu$ (ou juste $\mathscr{E}$) qui va de l'ensemble des formules logique dans $\mathscr{B}$. Elle est définie (par induction) par :

- $\mathscr{E}(V) = 1$ *et* $\mathscr{E}(F) = 0$.
- *Si v est une variable propositionnelle,* $\mathscr{E}(v) = \mu(v)$.
- *Si p_1 et p_2 sont des formules* $\mathscr{E}(p_1 \wedge p_2) = \mathscr{E}(p_1) \times \mathscr{E}(p_2)$. *Cela signifie que* $\mathscr{E}(p_1 \wedge p_2)$ *ne vaut* 1 *que si* $\mathscr{E}(p_1)$ *et* $\mathscr{E}(p_2)$ *valent* 1.
- *Si p_1 et p_2 sont des formules* $\mathscr{E}(p_1 \vee p_2) =$

Définition 3

Soit μ une distribution de vérité, on construit une fonction d'évaluation $\mathscr{E}_\mu$ (ou juste $\mathscr{E}$) qui va de l'ensemble des formules logique dans $\mathscr{B}$. Elle est définie (par induction) par :

- $\mathscr{E}(V) = 1$ *et* $\mathscr{E}(F) = 0$.
- *Si v est une variable propositionnelle,* $\mathscr{E}(v) = \mu(v)$.
- *Si p_1 et p_2 sont des formules* $\mathscr{E}(p_1 \wedge p_2) = \mathscr{E}(p_1) \times \mathscr{E}(p_2)$. *Cela signifie que* $\mathscr{E}(p_1 \wedge p_2)$ *ne vaut* 1 *que si* $\mathscr{E}(p_1)$ *et* $\mathscr{E}(p_2)$ *valent* 1.
- *Si p_1 et p_2 sont des formules* $\mathscr{E}(p_1 \vee p_2) = \mathscr{E}(p_1) + \mathscr{E}(p_2) - \mathscr{E}(p_1) \times \mathscr{E}(p_2)$. *Cela signifie que* $\mathscr{E}(p_1 \vee p_2)$ *vaut* 1 *si* $\mathscr{E}(p_1)$ *ou* $\mathscr{E}(p_2)$ *valent* 1.

Définition 3

Soit μ une distribution de vérité, on construit une fonction d'évaluation $\mathscr{E}_\mu$ (ou juste $\mathscr{E}$) qui va de l'ensemble des formules logique dans $\mathscr{B}$. Elle est définie (par induction) par :

- $\mathscr{E}(V) = 1$ *et* $\mathscr{E}(F) = 0$.
- *Si v est une variable propositionnelle,* $\mathscr{E}(v) = \mu(v)$.
- *Si* p_1 *et* p_2 *sont des formules* $\mathscr{E}(p_1 \wedge p_2) = \mathscr{E}(p_1) \times \mathscr{E}(p_2)$. *Cela signifie que* $\mathscr{E}(p_1 \wedge p_2)$ *ne vaut* 1 *que si* $\mathscr{E}(p_1)$ *et* $\mathscr{E}(p_2)$ *valent* 1.
- *Si* p_1 *et* p_2 *sont des formules* $\mathscr{E}(p_1 \vee p_2) = \mathscr{E}(p_1) + \mathscr{E}(p_2) - \mathscr{E}(p_1) \times \mathscr{E}(p_2)$. *Cela signifie que* $\mathscr{E}(p_1 \vee p_2)$ *vaut* 1 *si* $\mathscr{E}(p_1)$ *ou* $\mathscr{E}(p_2)$ *valent* 1.
- *Si p est une formule* $\mathscr{E}(\neg p) =$

Évaluation

Définition 3

Soit μ une distribution de vérité, on construit une fonction d'évaluation $\mathscr{E}_\mu$ (ou juste $\mathscr{E}$) qui va de l'ensemble des formules logique dans $\mathscr{B}$. Elle est définie (par induction) par :

- $\mathscr{E}(V) = 1$ *et* $\mathscr{E}(F) = 0$.
- *Si v est une variable propositionnelle,* $\mathscr{E}(v) = \mu(v)$.
- *Si* p_1 *et* p_2 *sont des formules* $\mathscr{E}(p_1 \wedge p_2) = \mathscr{E}(p_1) \times \mathscr{E}(p_2)$. *Cela signifie que* $\mathscr{E}(p_1 \wedge p_2)$ *ne vaut* 1 *que si* $\mathscr{E}(p_1)$ *et* $\mathscr{E}(p_2)$ *valent* 1.
- *Si* p_1 *et* p_2 *sont des formules* $\mathscr{E}(p_1 \vee p_2) = \mathscr{E}(p_1) + \mathscr{E}(p_2) - \mathscr{E}(p_1) \times \mathscr{E}(p_2)$. *Cela signifie que* $\mathscr{E}(p_1 \vee p_2)$ *vaut* 1 *si* $\mathscr{E}(p_1)$ *ou* $\mathscr{E}(p_2)$ *valent* 1.
- *Si p est une formule* $\mathscr{E}(\neg p) = 1 - \mathscr{E}(p)$.

Reprenons la formule $p = (v_1 \vee F) \wedge (v_3 \vee \neg v_2)$ et la distribution μ définie par $\mu(v_1) = \mu(v_2) = 1$ et $\mu(v_3) = 0$.

On a donc $\mathscr{E}(v_1 \vee F) = 1$ et $\mathscr{E}(v_3 \vee \neg v_2) = 0$ donc $\mathscr{E}(p) = 0$.

Remarque

Les formules ci-dessus correspondent à la signification classique des connecteurs et et ou.

Exercice

On considère μ telle que $\mu(v_1) = \mu(v_3) = 1$ et $\mu(v_2) = 0$.

Déterminer $\mathscr{E}\left((\neg v_1 \vee v_2) \wedge \neg(v_1 \wedge v_3)\right)$

On considère μ telle que $\mu(v_1) = \mu(v_3) = 1$ et $\mu(v_2) = 0$.

Déterminer $\mathscr{E}\left((\neg v_1 \lor v_2) \land \neg(v_1 \land v_3)\right)$
On a $\mathscr{E}\left((\neg v_1 \lor v_2\right) = 0$ donc $\mathscr{E}\left((\neg v_1 \lor v_2) \land \neg(v_1 \land v_3)\right) = 0$.

Exercice

Écrire une fonction `eval : formule -> (bool array) -> bool` telle que l'instruction `eval f mu` renvoie le booléen $\mathscr{E}_\mu(f)$.

Écrire une fonction `eval : formule -> (bool array) -> bool` telle que l'instruction `eval f mu` renvoie le booléen $\mathscr{E}_\mu(f)$.

```
let rec eval formule mu = match formule with
  | V -> true
  | F -> false
  | Var(x) -> mu.(x)
  | Non(f) -> not (eval f mu)
  | Ou(f1,f2) -> (eval f1 mu) || (eval f2 mu)
  | Ey(f1,f2) -> (eval f1 mu) && (eval f2 mu);;
```

Définition 4

Soit $\mathscr{V}$ un ensemble et p une formule sur $\mathscr{V}$.

1. *Soit μ une distribution de vérité, on dit que p est* ***satisfaite*** *pour μ ou que μ satisfait p si $\mathscr{E}_\mu(p) = 1$. On note alors $\mu \vDash p$.*

Définition 4

Soit $\mathscr{V}$ un ensemble et p une formule sur $\mathscr{V}$.

1. *Soit μ une distribution de vérité, on dit que p est* ***satisfaite*** *pour μ ou que μ satisfait p si $\mathscr{E}_\mu(p) = 1$. On note alors $\mu \vDash p$.*
2. *On dit que la formule p est* ***satisfiable*** *s'il existe une distribution μ telle que p soit satisfaite pour μ.*

Dans le cas général il est difficile de savoir si une formule est satisfiable. L'algorithme naïf consiste en effet à tester toutes les 2^n distributions de vérités (dans le cas où $\mathscr{V}$ est de cardinal n). Nous en reparlerons plus loin.

On sera amené à établir la *table de vérité* d'un formule qui est la table qui donne l'évaluation d'une formule p pour toutes les distributions de vérités (on placera ces dernières dans l'ordre lexicographique pour ne pas en oublier).

Reprenons l'exemple de $p = (v_1 \vee F) \vee (v_3 \vee \neg v_2)$.

v_1	v_2	v_3	$v_1 \vee F$	$v_3 \vee \neg v_2$	p
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	1	1

Exercices

Donner les tables de vérités de $\neg v$, $v_1 \wedge v_2$, $v_1 \vee v_2$.

v	$\neg v$
0	1
1	0

Exercices

v_1	v_2	$v_1 \wedge v_2$	$v_1 \wedge v_2$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Définition 5

*Soit p_1 et p_2 deux formules logiques sur le même ensemble $\mathscr{V}$. On dit que p_1 et p_2 sont **équivalentes** et on note $p_1 \equiv p_2$ si pour toute distribution de vérité μ, $\mathscr{E}_\mu(p_1) = \mathscr{E}_\mu(p_2)$.*

Définition 5

Soit p_1 et p_2 deux formules logiques sur le même ensemble $\mathscr{V}$. On dit que p_1 et p_2 sont ***équivalentes*** *et on note $p_1 \equiv p_2$ si pour toute distribution de vérité μ, $\mathscr{E}_\mu(p_1) = \mathscr{E}_\mu(p_2)$.*

Cela signifie que p_1 est vrai si et seulement si p_2 l'est aussi. C'est-à-dire :

$$\forall \mu \in \mathscr{B}^{\mathscr{V}}, \mu \models p_1 \iff \mu \models p_2.$$

Exemples

- Les formules $v_1 \wedge v_2$ et $v_2 \wedge v_1$ sont équivalentes. Il suffit d'établir les tables de vérités.

- Les formules $v_1 \wedge v_2$ et $v_2 \wedge v_1$ sont équivalentes. Il suffit d'établir les tables de vérités.
- Les formules v_1 et $v_1 \vee (v_2 \wedge \neg v_2)$ sont équivalentes. Il suffit d'établir les tables de vérités.

Attention

Ne pas confondre des formules égales et des formules équivalentes. Les deux formules

sont équivalentes mais elles ne sont pas égales.
Elles sont égales **sémantiquement** mais pas **syntaxiquement.**

Attention

Ne pas confondre des formules égales et des formules équivalentes. Les deux formules

$$v_1 \text{ et } v_1 \vee (v_2 \wedge \neg v_2)$$

sont équivalentes mais elles ne sont pas égales.
Elles sont égales **sémantiquement** mais pas **syntaxiquement.**

Notations

Définition 6

Soit p_1 et p_2 deux formules, on utilise les abréviations suivantes :

- *on note $p_1 \Rightarrow p_2$ pour $p_2 \vee \neg p_1$*
- *on note $p_1 \Leftrightarrow p_2$ pour $(p_1 \Rightarrow p_2) \wedge (p_2 \Rightarrow p_1)$*
- *on note p_1 xor p_2 pour $(p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge p_2)$*

Exercice

Déterminer, via des tables de vérités les valeurs de $p_1 \Rightarrow p_2$, $p_1 \Leftrightarrow p_2$ et p_1 xor p_2 en fonction de celles de p_1 et p_2.

$$(p_1 \Rightarrow p_2) = (p_2 \vee \neg p_1)$$

p_1	p_2	$\neg p_1$	$p_1 \Rightarrow p_2$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

Exercice

$$(p_1 \Leftrightarrow p_2) = (p_1 \Rightarrow p_2) \wedge (p_2 \Rightarrow p_1)$$

p_1	p_2	$p_1 \Rightarrow p_2$	$p_1 \Rightarrow p_2$	$p_1 \Leftrightarrow p_2$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	1	1	1

Exercice

$$(p_1 \text{ xor } p_2) = (p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge p_2)$$

p_1	p_2	$p_1 \wedge \neg p_2$	$\neg p_1 \wedge p_2$	p_1 xor p_2
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

Remarque

Quand on ne s'intéresse qu'à la sémantique et pas à la syntaxe, on pourra utiliser ces notations pour des formules qui y sont équivalentes.

Par exemple on pourra noter $v_2 \Rightarrow v_1$ pour $v_1 \vee ((\neg v_2 \wedge v_1) \vee (\neg v_2 \wedge \neg v_1))$

Défintion

Définition 7

*Une formule logique définie sur $\mathscr{V}$ est une **tautologie** si elle est satisfaite pour tout distribution de vérité.*

Remarques

- Une tautologie est donc quelque chose qui est toujours vrai

Remarques

- Une tautologie est donc quelque chose qui est toujours vrai
- On appelle à l'inverse une **antilogie** une formule qui n'est pas satisfiable.

Remarques

- Une tautologie est donc quelque chose qui est toujours vrai
- On appelle à l'inverse une **antilogie** une formule qui n'est pas satisfiable.
- Si p_1 et p_2 sont des propositions logiques, $p_1 \equiv p_2$ si et seulement si $p_1 \Leftrightarrow p_2$ est une tautologie.

Attention

Attention

Ne pas confondre « p est satisfiable » (**il existe** μ telle que $\mu \models p$) avec « p est une tautologie » (**pour tout** μ, $\mu \models p$).

Théorème 1 (Propriétés de $\wedge$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies

- $(p_1 \wedge F) \iff$

Théorème 1 (Propriétés de $\wedge$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies

- $(p_1 \wedge F) \iff F$

Théorème 1 (Propriétés de $\wedge$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies

- $(p_1 \wedge F) \iff F$
- $(p_1 \wedge V) \iff$

Théorème 1 (Propriétés de $\wedge$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies

- $(p_1 \wedge F) \iff F$
- $(p_1 \wedge V) \iff p_1$

Théorème 1 (Propriétés de $\wedge$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies

- $(p_1 \wedge F) \iff F$
- $(p_1 \wedge V) \iff p_1$
- $(p_1 \wedge p_2) \iff$

Théorème 1 (Propriétés de $\wedge$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies

- $(p_1 \wedge F) \iff F$
- $(p_1 \wedge V) \iff p_1$
- $(p_1 \wedge p_2) \iff (p_2 \wedge p_1)$

Théorème 1 (Propriétés de $\wedge$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies

- $(p_1 \wedge F) \iff F$
- $(p_1 \wedge V) \iff p_1$
- $(p_1 \wedge p_2) \iff (p_2 \wedge p_1)$
- $(p_1 \wedge (p_2 \wedge p_3)) \iff$

Théorème 1 (Propriétés de $\wedge$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies

- $(p_1 \wedge F) \iff F$
- $(p_1 \wedge V) \iff p_1$
- $(p_1 \wedge p_2) \iff (p_2 \wedge p_1)$
- $(p_1 \wedge (p_2 \wedge p_3)) \iff ((p_1 \wedge p_2) \wedge p_3)$

Théorème 1 (Propriétés de $\wedge$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies

- $(p_1 \wedge F) \iff F$
- $(p_1 \wedge V) \iff p_1$
- $(p_1 \wedge p_2) \iff (p_2 \wedge p_1)$
- $(p_1 \wedge (p_2 \wedge p_3)) \iff ((p_1 \wedge p_2) \wedge p_3)$
- $(p_1 \wedge p_1) \iff$

Théorème 1 (Propriétés de $\wedge$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies

- $(p_1 \wedge F) \iff F$
- $(p_1 \wedge V) \iff p_1$
- $(p_1 \wedge p_2) \iff (p_2 \wedge p_1)$
- $(p_1 \wedge (p_2 \wedge p_3)) \iff ((p_1 \wedge p_2) \wedge p_3)$
- $(p_1 \wedge p_1) \iff p_1$

Théorème 2 (Propriétés de $\vee$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \vee F) \iff$

Théorème 2 (Propriétés de $\vee$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \vee F) \iff p_1$

Théorème 2 (Propriétés de $\vee$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \vee F) \iff p_1$
- $(p_1 \vee V) \iff$

Propriétés

Théorème 2 (Propriétés de $\vee$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \vee F) \iff p_1$
- $(p_1 \vee V) \iff V$

Théorème 2 (Propriétés de $\vee$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \vee F) \iff p_1$
- $(p_1 \vee V) \iff V$
- $(p_1 \vee p_2) \iff$

Théorème 2 (Propriétés de $\vee$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \vee F) \iff p_1$
- $(p_1 \vee V) \iff V$
- $(p_1 \vee p_2) \iff (p_2 \vee p_1)$

Théorème 2 (Propriétés de $\vee$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \vee F) \iff p_1$
- $(p_1 \vee V) \iff V$
- $(p_1 \vee p_2) \iff (p_2 \vee p_1)$
- $(p_1 \vee (p_2 \vee p_3)) \iff$

Théorème 2 (Propriétés de $\vee$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \vee F) \iff p_1$
- $(p_1 \vee V) \iff V$
- $(p_1 \vee p_2) \iff (p_2 \vee p_1)$
- $(p_1 \vee (p_2 \vee p_3)) \iff ((p_1 \vee p_2) \vee p_3)$

Théorème 2 (Propriétés de $\vee$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \vee F) \iff p_1$
- $(p_1 \vee V) \iff V$
- $(p_1 \vee p_2) \iff (p_2 \vee p_1)$
- $(p_1 \vee (p_2 \vee p_3)) \iff ((p_1 \vee p_2) \vee p_3)$
- $(p_1 \vee p_1) \iff$

Théorème 2 (Propriétés de $\vee$)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \vee F) \iff p_1$
- $(p_1 \vee V) \iff V$
- $(p_1 \vee p_2) \iff (p_2 \vee p_1)$
- $(p_1 \vee (p_2 \vee p_3)) \iff ((p_1 \vee p_2) \vee p_3)$
- $(p_1 \vee p_1) \iff p_1$

Propriétés

Théorème 3 (Autres tautologies)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \wedge (p_2 \vee p_3)) \iff$

Théorème 3 (Autres tautologies)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \wedge (p_2 \vee p_3)) \iff (p_1 \wedge p_2) \vee (p_1 \wedge p_3)$

Théorème 3 (Autres tautologies)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \wedge (p_2 \vee p_3)) \iff (p_1 \wedge p_2) \vee (p_1 \wedge p_3)$
- $(p_1 \vee (p_2 \wedge p_3)) \iff$

Théorème 3 (Autres tautologies)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \land (p_2 \lor p_3)) \iff (p_1 \land p_2) \lor (p_1 \land p_3)$
- $(p_1 \lor (p_2 \land p_3)) \iff (p_1 \lor p_2) \land (p_1 \lor p_3)$ (Distributivité)

Théorème 3 (Autres tautologies)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \wedge (p_2 \vee p_3)) \iff (p_1 \wedge p_2) \vee (p_1 \wedge p_3)$
- $(p_1 \vee (p_2 \wedge p_3)) \iff (p_1 \vee p_2) \wedge (p_1 \vee p_3)$ (Distributivité)
- $\neg(p_1 \wedge p_2) \iff$

Théorème 3 (Autres tautologies)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \wedge (p_2 \vee p_3)) \iff (p_1 \wedge p_2) \vee (p_1 \wedge p_3)$
- $(p_1 \vee (p_2 \wedge p_3)) \iff (p_1 \vee p_2) \wedge (p_1 \vee p_3)$ (Distributivité)
- $\neg(p_1 \wedge p_2) \iff (\neg p_1 \vee \neg p_2)$

Théorème 3 (Autres tautologies)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \wedge (p_2 \vee p_3)) \iff (p_1 \wedge p_2) \vee (p_1 \wedge p_3)$
- $(p_1 \vee (p_2 \wedge p_3)) \iff (p_1 \vee p_2) \wedge (p_1 \vee p_3)$ (Distributivité)
- $\neg(p_1 \wedge p_2) \iff (\neg p_1 \vee \neg p_2)$
- $\neg(p_1 \vee p_2) \iff$

Théorème 3 (Autres tautologies)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \wedge (p_2 \vee p_3)) \iff (p_1 \wedge p_2) \vee (p_1 \wedge p_3)$
- $(p_1 \vee (p_2 \wedge p_3)) \iff (p_1 \vee p_2) \wedge (p_1 \vee p_3)$ (Distributivité)
- $\neg(p_1 \wedge p_2) \iff (\neg p_1 \vee \neg p_2)$
- $\neg(p_1 \vee p_2) \iff (\neg p_1 \wedge \neg p_2)$ (Loi de De Morgan)

Théorème 3 (Autres tautologies)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \wedge (p_2 \vee p_3)) \iff (p_1 \wedge p_2) \vee (p_1 \wedge p_3)$
- $(p_1 \vee (p_2 \wedge p_3)) \iff (p_1 \vee p_2) \wedge (p_1 \vee p_3)$ (Distributivité)
- $\neg(p_1 \wedge p_2) \iff (\neg p_1 \vee \neg p_2)$
- $\neg(p_1 \vee p_2) \iff (\neg p_1 \wedge \neg p_2)$ (Loi de De Morgan)
- $(p_1 \vee \neg p_1) \iff$

Théorème 3 (Autres tautologies)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \wedge (p_2 \vee p_3)) \iff (p_1 \wedge p_2) \vee (p_1 \wedge p_3)$
- $(p_1 \vee (p_2 \wedge p_3)) \iff (p_1 \vee p_2) \wedge (p_1 \vee p_3)$ (Distributivité)
- $\neg(p_1 \wedge p_2) \iff (\neg p_1 \vee \neg p_2)$
- $\neg(p_1 \vee p_2) \iff (\neg p_1 \wedge \neg p_2)$ (Loi de De Morgan)
- $(p_1 \vee \neg p_1) \iff V$ (Tiers exclus)

Théorème 3 (Autres tautologies)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \wedge (p_2 \vee p_3)) \iff (p_1 \wedge p_2) \vee (p_1 \wedge p_3)$
- $(p_1 \vee (p_2 \wedge p_3)) \iff (p_1 \vee p_2) \wedge (p_1 \vee p_3)$ (Distributivité)
- $\neg(p_1 \wedge p_2) \iff (\neg p_1 \vee \neg p_2)$
- $\neg(p_1 \vee p_2) \iff (\neg p_1 \wedge \neg p_2)$ (Loi de De Morgan)
- $(p_1 \vee \neg p_1) \iff V$ (Tiers exclus)
- $(p_1 \wedge (p_1 \Rightarrow p_2)) \Rightarrow$

Théorème 3 (Autres tautologies)

On désigne par p_1, p_2 et p_3 des formules logiques. Les propositions suivantes sont des tautologies :

- $(p_1 \wedge (p_2 \vee p_3)) \iff (p_1 \wedge p_2) \vee (p_1 \wedge p_3)$
- $(p_1 \vee (p_2 \wedge p_3)) \iff (p_1 \vee p_2) \wedge (p_1 \vee p_3)$ (Distributivité)
- $\neg(p_1 \wedge p_2) \iff (\neg p_1 \vee \neg p_2)$
- $\neg(p_1 \vee p_2) \iff (\neg p_1 \wedge \neg p_2)$ (Loi de De Morgan)
- $(p_1 \vee \neg p_1) \iff V$ (Tiers exclus)
- $(p_1 \wedge (p_1 \Rightarrow p_2)) \Rightarrow p_2$ (Modus Ponens)