Définition

Soit V un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur V par :

Définition

Soit V un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur V par :

- L'ensemble d'axiomes $P_0 = \{\top, \bot\} \cup V$

Définition

Soit V un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur V par :

- L'ensemble d'axiomes $P_0 = \{\top, \bot\} \cup V$
  $\top$ se lit « top » et $\bot$ se lit « bottom »

Définition

Soit V un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur V par :

- L'ensemble d'axiomes $P_0 = \{\top, \bot\} \cup V$
  $\top$ se lit « top » et $\bot$ se lit « bottom »
- Les règles d'inférence :
  - *négation* $\neg$: $p \mapsto \neg p$

Définition

Soit V un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur V par :

- L'ensemble d'axiomes $P_0 = \{\top, \bot\} \cup V$
  $\top$ se lit « top » et $\bot$ se lit « bottom »
- Les règles d'inférence :
  - *négation* $\neg$: $p \mapsto \neg p$
  - *conjonction* $\wedge$: $p, q \mapsto (p \wedge q)$

Définition

Soit V un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur V par :

- L'ensemble d'axiomes $P_0 = \{\top, \bot\} \cup V$
  $\top$ se lit « top » et $\bot$ se lit « bottom »
- Les règles d'inférence :
  - *négation* $\neg$: $p \mapsto \neg p$
  - *conjonction* $\wedge$: $p, q \mapsto (p \wedge q)$
  - *disjonction* $\vee$: $p, q \mapsto (p \vee q)$

Définition

Soit V un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur V par :

- L'ensemble d'axiomes $P_0 = \{\top, \bot\} \cup V$
  $\top$ se lit « top » et $\bot$ se lit « bottom »
- Les règles d'inférence :
  - *négation* $\neg : p \mapsto \neg p$
  - *conjonction* $\wedge : p, q \mapsto (p \wedge q)$
  - *disjonction* $\vee : p, q \mapsto (p \vee q)$
  - *implication* $\rightarrow : p, q \mapsto (p \rightarrow q)$

Définition

Soit V un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur V par :

- L'ensemble d'axiomes $P_0 = \{\top, \bot\} \cup V$
  $\top$ se lit « top » et $\bot$ se lit « bottom »
- Les règles d'inférence :
  - *négation* $\neg$: $p \mapsto \neg p$
  - *conjonction* $\wedge$: $p, q \mapsto (p \wedge q)$
  - *disjonction* $\vee$: $p, q \mapsto (p \vee q)$
  - *implication* $\rightarrow$: $p, q \mapsto (p \rightarrow q)$
  - *équivalence* $\leftrightarrow$: $p, q \mapsto (p \leftrightarrow q)$

Remarques

- Afin d'éviter de surcharger les écritures, on pourra omettre certaines parenthèses :

Remarques

- Afin d'éviter de surcharger les écritures, on pourra omettre certaines parenthèses :
  - En utilisant l'associativité à droite de $\wedge, \vee$:

Remarques

- Afin d'éviter de surcharger les écritures, on pourra omettre certaines parenthèses :
  - En utilisant l'associativité à droite de $\wedge, \vee$:
    Par exemple, $(p \vee (q \vee r))$ s'écrit plus simplement $p \vee q \vee r$.

Remarques

- Afin d'éviter de surcharger les écritures, on pourra omettre certaines parenthèses :
    - En utilisant l'associativité à droite de $\wedge, \vee$:
      Par exemple, $(p \vee (q \vee r))$ s'écrit plus simplement $p \vee q \vee r$.
    - En utilisant l'ordre de priorité suivant sur les connecteurs : $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

Remarques

- Afin d'éviter de surcharger les écritures, on pourra omettre certaines parenthèses :
  - En utilisant l'associativité à droite de $\wedge, \vee$:
    Par exemple, $(p \vee (q \vee r))$ s'écrit plus simplement $p \vee q \vee r$.
  - En utilisant l'ordre de priorité suivant sur les connecteurs : $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
    Par exemple $((\neg p) \vee (q \wedge r))$ s'écrit plus simplement $\neg p \vee q \wedge r$.

Remarques

- Afin d'éviter de surcharger les écritures, on pourra omettre certaines parenthèses :
  - En utilisant l'associativité à droite de $\wedge, \vee$:
    Par exemple, $(p \vee (q \vee r))$ s'écrit plus simplement $p \vee q \vee r$.
  - En utilisant l'ordre de priorité suivant sur les connecteurs : $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
    Par exemple $((\neg p) \vee (q \wedge r))$ s'écrit plus simplement $\neg p \vee q \wedge r$.

En cas de doute, on laissera les parenthèses afin de lever toute ambiguïté.

Remarques

- Afin d'éviter de surcharger les écritures, on pourra omettre certaines parenthèses :
  - En utilisant l'associativité à droite de $\wedge, \vee$:
    Par exemple, $(p \vee (q \vee r))$ s'écrit plus simplement $p \vee q \vee r$.
  - En utilisant l'ordre de priorité suivant sur les connecteurs : $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
    Par exemple $((\neg p) \vee (q \wedge r))$ s'écrit plus simplement $\neg p \vee q \wedge r$.

  En cas de doute, on laissera les parenthèses afin de lever toute ambiguïté.
- On a définit pour le moment simplement les propositions logiques valables d'un point de vue *syntaxique*, sans leur donner de sens ou de valeur.

Remarques

- Afin d'éviter de surcharger les écritures, on pourra omettre certaines parenthèses :
  - En utilisant l'associativité à droite de $\wedge, \vee$:
    Par exemple, $(p \vee (q \vee r))$ s'écrit plus simplement $p \vee q \vee r$.
  - En utilisant l'ordre de priorité suivant sur les connecteurs : $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
    Par exemple $((\neg p) \vee (q \wedge r))$ s'écrit plus simplement $\neg p \vee q \wedge r$.

  En cas de doute, on laissera les parenthèses afin de lever toute ambiguïté.
- On a définit pour le moment simplement les propositions logiques valables d'un point de vue *syntaxique*, sans leur donner de sens ou de valeur.

Exemples

- $(((\neg p) \vee (\neg q)) \wedge r)$ est une formule logique qu'on pourra écrire plus simplement $(\neg p \vee \neg q) \wedge r$.
- $\wedge p \neg pq$ ou encore $(p \wedge q) \rightarrow (r$ ne sont pas des formules logiques.

Remarques

- Afin d'éviter de surcharger les écritures, on pourra omettre certaines parenthèses :
  - En utilisant l'associativité à droite de $\wedge, \vee$:
    Par exemple, $(p \vee (q \vee r))$ s'écrit plus simplement $p \vee q \vee r$.
  - En utilisant l'ordre de priorité suivant sur les connecteurs : $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
    Par exemple $((\neg p) \vee (q \wedge r))$ s'écrit plus simplement $\neg p \vee q \wedge r$.

  En cas de doute, on laissera les parenthèses afin de lever toute ambiguïté.
- On a définit pour le moment simplement les propositions logiques valables d'un point de vue *syntaxique*, sans leur donner de sens ou de valeur.

Exemples

- $(((\neg p) \vee (\neg q)) \wedge r)$ est une formule logique qu'on pourra écrire plus simplement $(\neg p \vee \neg q) \wedge r$.
- $\wedge p \neg pq$ ou encore $(p \wedge q) \rightarrow (r$ ne sont pas des formules logiques.
- $(p \rightarrow q) \wedge (q \rightarrow p)$ et $p \leftrightarrow q$ sont deux formules logiques *différentes*.

Arbre syntaxique

Les formules logiques admettent naturellement une représentation sous forme d'arbre :

Arbre syntaxique

Les formules logiques admettent naturellement une représentation sous forme d'arbre :

- les variables logiques et les constantes $\top$ et $\bot$ sont les étiquettes des feuilles

Arbre syntaxique

Les formules logiques admettent naturellement une représentation sous forme d'arbre :

- les variables logiques et les constantes $\top$ et $\bot$ sont les étiquettes des feuilles
- les noeuds internes ont pour étiquette les règles d'inférence

Arbre syntaxique

Les formules logiques admettent naturellement une représentation sous forme d'arbre :

- les variables logiques et les constantes $\top$ et $\bot$ sont les étiquettes des feuilles
- les noeuds internes ont pour étiquette les règles d'inférence

Exemples

La formule logique $(p \rightarrow q) \vee (\neg r)$ admet la représentation :

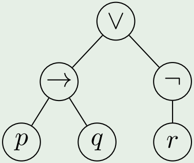
Arbre syntaxique

Les formules logiques admettent naturellement une représentation sous forme d'arbre :

- les variables logiques et les constantes $\top$ et $\bot$ sont les étiquettes des feuilles
- les noeuds internes ont pour étiquette les règles d'inférence

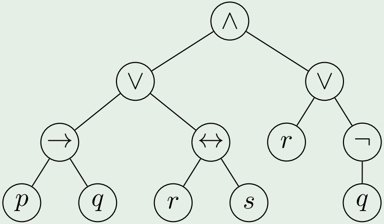
Exemples

La formule logique $(p \to q) \vee (\neg r)$ admet la représentation :



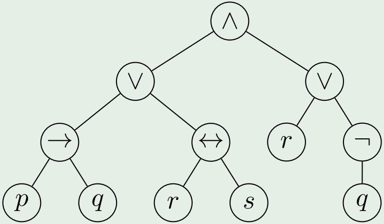
Exemple

- Quelle est la formule logique ayant pour représentation arborescente :



Exemple

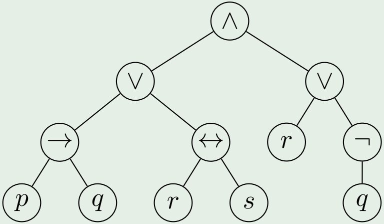
- Quelle est la formule logique ayant pour représentation arborescente :



$((p \rightarrow q) \vee (r \leftrightarrow s)) \wedge (r \vee (\neg q))$

Exemple

- Quelle est la formule logique ayant pour représentation arborescente :



$((p \rightarrow q) \vee (r \leftrightarrow s)) \wedge (r \vee (\neg q))$

- Dessiner la représentation arborescente de $\neg(\top \leftrightarrow (p \vee q))$.

Hauteur, taille et sous formule

Etant donnée une formule logique notée P,

Hauteur, taille et sous formule

Etant donnée une formule logique notée P,

- la hauteur de P est la hauteur de l'arbre syntaxique associé

Hauteur, taille et sous formule

Etant donnée une formule logique notée P,

- la hauteur de P est la hauteur de l'arbre syntaxique associé
- la taille de P est le nombre de noeuds de l'arbre syntaxiqué associé

Hauteur, taille et sous formule

Etant donnée une formule logique notée P,

- la hauteur de P est la hauteur de l'arbre syntaxique associé
- la taille de P est le nombre de noeuds de l'arbre syntaxiqué associé
- Une sous formule de P est un sous-arbre de l'arbre syntaxique associé

Implémentation en OCaml

Le type somme de OCaml associé à la correspondance de motif permettent de représenter et de travailler efficacement sur les formules logiques.

Implémentation en OCaml

Le type somme de OCaml associé à la correspondance de motif permettent de représenter et de travailler efficacement sur les formules logiques.

```ocaml

type fl =
| Top | Bot
| Var of int (*les variables propositionnelles*)
| Non of fl
| Ou of fl*fl
| Et of fl*fl
| Imp of fl*fl
```

Implémentation en OCaml

Le type somme de OCaml associé à la correspondance de motif permettent de représenter et de travailler efficacement sur les formules logiques.

```

type fl =
| Top | Bot
| Var of int (*les variables propositionnelles*)
| Non of fl
| Ou of fl*fl
| Et of fl*fl
| Imp of fl*fl
```

On a représenté ici une variable logique par un entier, on pourrait choisir un caractère, ou un type option 'a.

Implémentation en OCaml

Le type somme de OCaml associé à la correspondance de motif permettent de représenter et de travailler efficacement sur les formules logiques.

```ocaml

type fl =
| Top | Bot
| Var of int (*les variables propositionnelles*)
| Non of fl
| Ou of fl*fl
| Et of fl*fl
| Imp of fl*fl
```

On a représenté ici une variable logique par un entier, on pourrait choisir un caractère, ou un type option 'a.

La formule logique $(p \wedge \neg q) \rightarrow r$ est alors est par :

Implémentation en OCaml

Le type somme de OCaml associé à la correspondance de motif permettent de représenter et de travailler efficacement sur les formules logiques.

```ocaml

type fl =
| Top | Bot
| Var of int (*les variables propositionnelles*)
| Non of fl
| Ou of fl*fl
| Et of fl*fl
| Imp of fl*fl
```

On a représenté ici une variable logique par un entier, on pourrait choisir un caractère, ou un type option 'a.

La formule logique $(p \wedge \neg q) \rightarrow r$ est alors est par :

```ocaml

```

Implémentation en OCaml

Le calcul de la taille s'obtient alors via un *pattern matching* :

Implémentation en OCaml

Le calcul de la taille s'obtient alors via un *pattern matching* :

```
      t : bool array; (* le tableau de booléen *)
      mutable parite : bool (* la parité du nombre de 1*)
  }


let ex = Imp (Et ((Var 1),(Non (Var 2))), (Var 3)) ;;

let rec taille fl =
```

Valuation

On note $\mathbb{B} = \{V, F\}$, l'ensemble des valeurs de vérités Une textcolorbluevaluation est une attribution de l'une des deux valeurs de vérités à chaque variable propositionnelle. Une valuation φ est une donc une application de V de B.

Valuation

On note $\mathbb{B} = \{V, F\}$, l'ensemble des valeurs de vérités Une textcolorbluevaluation est une attribution de l'une des deux valeurs de vérités à chaque variable propositionnelle. Une valuation φ est une donc une application de V de B.

Exemple

Si l'ensemble des variables propositionnelle est $V = \{p, q, r\}$, une valuation possible est :
$\varphi : V \mapsto \mathbb{B}$, avec $\varphi(p) = V$, $\varphi(q) = F$ et $\varphi(r) = F$.

Valuation

On note $\mathbb{B} = \{V, F\}$, l'ensemble des valeurs de vérités Une textcolorbluevaluation est une attribution de l'une des deux valeurs de vérités à chaque variable propositionnelle. Une valuation φ est une donc une application de V de B.

Exemple

Si l'ensemble des variables propositionnelle est $V = \{p, q, r\}$, une valuation possible est :
$\varphi : V \mapsto \mathbb{B}$, avec $\varphi(p) = V$, $\varphi(q) = F$ et $\varphi(r) = F$.

Remarque

En nontant $|V| = n$, il y a 2^n valuations possibles.

Fonction booléennes usuelles

On rappelle les fonction booléennes usuelles associées à chaque connecteur :

$f_\wedge : \mathbb{B}^2 \mapsto \mathbb{B}$

x	y	$f_\wedge(x,y)$
F	F	F
F	V	F
V	F	F
V	V	V

$f_\vee : \mathbb{B}^2 \mapsto \mathbb{B}$

x	y	$f_\vee(x,y)$
F	F	F
F	V	V
V	F	V
V	V	V

$f_\rightarrow : \mathbb{B}^2 \mapsto \mathbb{B}$

x	y	$f_\rightarrow(x,y)$
F	F	V
F	V	V
V	F	F
V	V	V

$f_\leftrightarrow : \mathbb{B}^2 \mapsto \mathbb{B}$

x	y	$f_\leftrightarrow(x,y)$
F	F	F
F	V	F
V	F	F
V	V	V

Et $f_\neg : \mathbb{B} \mapsto \mathbb{B}$, définie par $f_\neg(F) = V$ et $f_\neg(V) = F$.

Valeur de vérité d'une formule

Pour une valuation φ, on définit la valeur de vérité d'une formule P notée $[\![P]\!]_\varphi$ par :

Valeur de vérité d'une formule

Pour une valuation φ, on définit la valeur de vérité d'une formule P notée $[\![P]\!]_\varphi$ par :

- $[\![\top]\!]_\varphi = V$
- $[\![\bot]\!]_\varphi = F$

Valeur de vérité d'une formule

Pour une valuation φ, on définit la valeur de vérité d'une formule P notée $[\![P]\!]_\varphi$ par :

- $[\![\top]\!]_\varphi = V$
- $[\![\bot]\!]_\varphi = F$
- si $v \in V$, $[\![v]\!]_\varphi = \varphi(v)$

Valeur de vérité d'une formule

Pour une valuation φ, on définit la valeur de vérité d'une formule P notée $[\![P]\!]_\varphi$ par :

- $[\![\top]\!]_\varphi = V$
- $[\![\bot]\!]_\varphi = F$
- si $v \in V$, $[\![v]\!]_\varphi = \varphi(v)$
- $[\![\neg Q]\!]_\varphi = f_\neg([\![Q]\!]_\varphi)$

Valeur de vérité d'une formule

Pour une valuation φ, on définit la valeur de vérité d'une formule P notée $[\![P]\!]_\varphi$ par :

- $[\![\top]\!]_\varphi = V$
- $[\![\bot]\!]_\varphi = F$
- si $v \in V$, $[\![v]\!]_\varphi = \varphi(v)$
- $[\![\neg Q]\!]_\varphi = f_\neg([\![Q]\!]_\varphi)$
- $[\![(Q \wedge R)]\!]_\varphi = f_\wedge([\![Q]\!]_\varphi, [\![R]\!]_\varphi)$
- $[\![(Q \vee R)]\!]_\varphi = f_\vee([\![Q]\!]_\varphi, [\![R]\!]_\varphi)$
- $[\![(Q \rightarrow R)]\!]_\varphi = f_\rightarrow([\![Q]\!]_\varphi, [\![R]\!]_\varphi)$
- $[\![(Q \leftrightarrow R)]\!]_\varphi = f_\leftrightarrow([\![Q]\!]_\varphi, [\![R]\!]_\varphi)$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = V$, $\varphi(q) = F$ et $\varphi(r) = F$ on peut alors déterminer la valeur de vérité de la formule logique $P = (p \to q) \wedge (\neg p \vee r)$ associée à cette valuation :

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = V$, $\varphi(q) = F$ et $\varphi(r) = F$ on peut alors déterminer la valeur de vérité de la formule logique $P = (p \to q) \wedge (\neg p \vee r)$ associée à cette valuation :
$[\![P]\!]_\varphi = f_\wedge([\![p \to q]\!]_\varphi, [\![\neg p \vee r]\!]_\varphi)$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = V$, $\varphi(q) = F$ et $\varphi(r) = F$ on peut alors déterminer la valeur de vérité de la formule logique $P = (p \to q) \wedge (\neg p \vee r)$ associée à cette valuation :

$[\![P]\!]_\varphi = f_\wedge([\![p \to q]\!]_\varphi, [\![\neg p \vee r]\!]_\varphi)$

$[\![P]\!]_\varphi = f_\wedge(f_\to([\![p]\!]_\varphi, [\![q]\!]_\varphi)), f_\vee([\![\neg p]\!]_\varphi, [\![r]\!]_\varphi)$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = V$, $\varphi(q) = F$ et $\varphi(r) = F$ on peut alors déterminer la valeur de vérité de la formule logique $P = (p \rightarrow q) \wedge (\neg p \vee r)$ associée à cette valuation :

$[\![P]\!]_\varphi = f_\wedge([\![p \rightarrow q]\!]_\varphi, [\![\neg p \vee r]\!]_\varphi)$

$[\![P]\!]_\varphi = f_\wedge(f_\rightarrow([\![p]\!]_\varphi, [\![q]\!]_\varphi)), f_\vee([\![\neg p]\!]_\varphi, [\![r]\!]_\varphi)$

$[\![P]\!]_\varphi = f_\wedge(f_\rightarrow(V, F), f_\vee(F, F))$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = V$, $\varphi(q) = F$ et $\varphi(r) = F$ on peut alors déterminer la valeur de vérité de la formule logique $P = (p \to q) \wedge (\neg p \vee r)$ associée à cette valuation :

$[\![P]\!]_\varphi = f_\wedge([\![p \to q]\!]_\varphi, [\![\neg p \vee r]\!]_\varphi)$
$[\![P]\!]_\varphi = f_\wedge(f_\to([\![p]\!]_\varphi, [\![q]\!]_\varphi)), f_\vee([\![\neg p]\!]_\varphi, [\![r]\!]_\varphi)$
$[\![P]\!]_\varphi = f_\wedge(f_\to(V, F), f_\vee(F, F))$
$[\![P]\!]_\varphi = f_\wedge(F, F)$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = V$, $\varphi(q) = F$ et $\varphi(r) = F$ on peut alors déterminer la valeur de vérité de la formule logique $P = (p \rightarrow q) \wedge (\neg p \vee r)$ associée à cette valuation :

$[\![P]\!]_\varphi = f_\wedge([\![p \rightarrow q]\!]_\varphi, [\![\neg p \vee r]\!]_\varphi)$
$[\![P]\!]_\varphi = f_\wedge(f_\rightarrow([\![p]\!]_\varphi, [\![q]\!]_\varphi)), f_\vee([\![\neg p]\!]_\varphi, [\![r]\!]_\varphi)$
$[\![P]\!]_\varphi = f_\wedge(f_\rightarrow(V, F), f_\vee(F, F))$
$[\![P]\!]_\varphi = f_\wedge(F, F)$
$[\![P]\!]_\varphi = F$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = V$, $\varphi(q) = F$ et $\varphi(r) = F$ on peut alors déterminer la valeur de vérité de la formule logique $P = (p \to q) \wedge (\neg p \vee r)$ associée à cette valuation :

$[\![P]\!]_\varphi = f_\wedge([\![p \to q]\!]_\varphi, [\![\neg p \vee r]\!]_\varphi)$
$[\![P]\!]_\varphi = f_\wedge(f_\to([\![p]\!]_\varphi, [\![q]\!]_\varphi)), f_\vee([\![\neg p]\!]_\varphi, [\![r]\!]_\varphi)$
$[\![P]\!]_\varphi = f_\wedge(f_\to(V, F), f_\vee(F, F))$
$[\![P]\!]_\varphi = f_\wedge(F, F)$
$[\![P]\!]_\varphi = F$

- Donner la valeur de vérité de cette proposition pour la valuation φ' définie par : $\varphi'(p) = F, \varphi'(q) = F, \varphi'(r) = F$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = V$, $\varphi(q) = F$ et $\varphi(r) = F$ on peut alors déterminer la valeur de vérité de la formule logique $P = (p \rightarrow q) \wedge (\neg p \vee r)$ associée à cette valuation :

$[\![P]\!]_\varphi = f_\wedge([\![p \rightarrow q]\!]_\varphi, [\![\neg p \vee r]\!]_\varphi)$
$[\![P]\!]_\varphi = f_\wedge(f_\rightarrow([\![p]\!]_\varphi, [\![q]\!]_\varphi)), f_\vee([\![\neg p]\!]_\varphi, [\![r]\!]_\varphi)$
$[\![P]\!]_\varphi = f_\wedge(f_\rightarrow(V, F), f_\vee(F, F))$
$[\![P]\!]_\varphi = f_\wedge(F, F)$
$[\![P]\!]_\varphi = F$

- Donner la valeur de vérité de cette proposition pour la valuation φ' définie par : $\varphi'(p) = F, \varphi'(q) = F, \varphi'(r) = F$
- Déterminer la valeur de vérité de $((p \rightarrow q) \vee (r \leftrightarrow s)) \wedge (r \vee (\neg q))$ pour la valuation φ.

Tautologie, antilogie, satisfiabilité

- Une formule est une tautologie si sa valeur de vérité est V pour toute valuation. C'est à dire que P est une tautologie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![P]\!]_\varphi = V$.

Tautologie, antilogie, satisfiabilité

- Une formule est une tautologie si sa valeur de vérité est V pour toute valuation. C'est à dire que P est une tautologie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![P]\!]_\varphi = V$.
- Une formule est une antilogie si sa valeur de vérité est F pour toute valuation. C'est à dire que P est une antilogie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![P]\!]_\varphi = F$.

Tautologie, antilogie, satisfiabilité

- Une formule est une tautologie si sa valeur de vérité est V pour toute valuation. C'est à dire que P est une tautologie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![P]\!]_\varphi = V$.
- Une formule est une antilogie si sa valeur de vérité est F pour toute valuation. C'est à dire que P est une antilogie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![P]\!]_\varphi = F$.
- Une formule est satisfiable s'il existe une valuation pour laquelle sa valeur de vérité est V. C'est à dire que P est satisfiable ssi il existe $\varphi \in \mathbb{B}^V$ tel que $[\![P]\!]_\varphi = V$.

Tautologie, antilogie, satisfiabilité

- Une formule est une tautologie si sa valeur de vérité est V pour toute valuation. C'est à dire que P est une tautologie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![P]\!]_\varphi = V$.
- Une formule est une antilogie si sa valeur de vérité est F pour toute valuation. C'est à dire que P est une antilogie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![P]\!]_\varphi = F$.
- Une formule est satisfiable s'il existe une valuation pour laquelle sa valeur de vérité est V. C'est à dire que P est satisfiable ssi il existe $\varphi \in \mathbb{B}^V$ tel que $[\![P]\!]_\varphi = V$.

Remarques

- P est une tautologie ssi $\neg P$ n'est pas satisfiable.

Tautologie, antilogie, satisfiabilité

- Une formule est une tautologie si sa valeur de vérité est V pour toute valuation. C'est à dire que P est une tautologie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![P]\!]_\varphi = V$.
- Une formule est une antilogie si sa valeur de vérité est F pour toute valuation. C'est à dire que P est une antilogie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![P]\!]_\varphi = F$.
- Une formule est satisfiable s'il existe une valuation pour laquelle sa valeur de vérité est V. C'est à dire que P est satisfiable ssi il existe $\varphi \in \mathbb{B}^V$ tel que $[\![P]\!]_\varphi = V$.

Remarques

- P est une tautologie ssi $\neg P$ n'est pas satisfiable.
- P est satisfiable ssi $\neg P$ n'est pas une tautologie.

Tables de vérité

La table de vérité d'une formule logique P contenant n variables logiques consiste à présenter sous forme de tableau la valeur de vérité de ces formules pour chacune des 2^n valuations possibles.

Tables de vérité

La table de vérité d'une formule logique P contenant n variables logiques consiste à présenter sous forme de tableau la valeur de vérité de ces formules pour chacune des 2^n valuations possibles.

Exemple

- Dresser la table de vérité de $P = (p \rightarrow q) \leftrightarrow (\neg p \vee q)$

Tables de vérité

La table de vérité d'une formule logique P contenant n variables logiques consiste à présenter sous forme de tableau la valeur de vérité de ces formules pour chacune des 2^n valuations possibles.

Exemple

- Dresser la table de vérité de $P = (p \to q) \leftrightarrow (\neg p \vee q)$
- Que peut-on en déduire ?

Tables de vérité

La table de vérité d'une formule logique P contenant n variables logiques consiste à présenter sous forme de tableau la valeur de vérité de ces formules pour chacune des 2^n valuations possibles.

Exemple

- Dresser la table de vérité de $P = (p \to q) \leftrightarrow (\neg p \lor q)$

p	q	$p \to q$	$\neg p \lor q$	P
F	F			
F	V			
V	F			
V	V			

- Que peut-on en déduire ?

Tables de vérité

La table de vérité d'une formule logique P contenant n variables logiques consiste à présenter sous forme de tableau la valeur de vérité de ces formules pour chacune des 2^n valuations possibles.

Exemple

- Dresser la table de vérité de $P = (p \to q) \leftrightarrow (\neg p \vee q)$

p	q	$p \to q$	$\neg p \vee q$	P
F	F	V		
F	V	V		
V	F	F		
V	V	V		

- Que peut-on en déduire ?

Tables de vérité

La table de vérité d'une formule logique P contenant n variables logiques consiste à présenter sous forme de tableau la valeur de vérité de ces formules pour chacune des 2^n valuations possibles.

Exemple

- Dresser la table de vérité de $P = (p \to q) \leftrightarrow (\neg p \vee q)$

p	q	$p \to q$	$\neg p \vee q$	P
F	F	V	V	
F	V	V	V	
V	F	F	F	
V	V	V	V	

- Que peut-on en déduire ?

Tables de vérité

La table de vérité d'une formule logique P contenant n variables logiques consiste à présenter sous forme de tableau la valeur de vérité de ces formules pour chacune des 2^n valuations possibles.

Exemple

- Dresser la table de vérité de $P = (p \to q) \leftrightarrow (\neg p \vee q)$

p	q	$p \to q$	$\neg p \vee q$	P
F	F	V	V	V
F	V	V	V	V
V	F	F	F	V
V	V	V	V	V

- Que peut-on en déduire ?

Tables de vérité

La table de vérité d'une formule logique P contenant n variables logiques consiste à présenter sous forme de tableau la valeur de vérité de ces formules pour chacune des 2^n valuations possibles.

Exemple

- Dresser la table de vérité de $P = (p \to q) \leftrightarrow (\neg p \vee q)$

p	q	$p \to q$	$\neg p \vee q$	P
F	F	V	V	V
F	V	V	V	V
V	F	F	F	V
V	V	V	V	V

- Que peut-on en déduire ? P est une tautologie.

Equivalence logique

On dit que deux formules logiques P et Q sont logiquement équivalentes si pour toute valuation φ, $[\![P]\!]_\varphi = [\![Q]\!]_\varphi$. On notera alors $P \equiv Q$.

Equivalence logique

On dit que deux formules logiques P et Q sont logiquement équivalentes si pour toute valuation φ, $[\![P]\!]_\varphi = [\![Q]\!]_\varphi$. On notera alors $P \equiv Q$.
Cela traduit l'égalité sémantique de P et Q, et permet de simplifier les formules.

Equivalence logique

On dit que deux formules logiques P et Q sont logiquement équivalentes si pour toute valuation φ, $[\![P]\!]_\varphi = [\![Q]\!]_\varphi$. On notera alors $P \equiv Q$.
Cela traduit l'égalité sémantique de P et Q, et permet de simplifier les formules.
A ne pas confondre avec $\leftrightarrow$ qui est un connecteur logique.

Quelques équivalences à connaitre

- $\neg(\neg P) \equiv P$

Equivalence logique

On dit que deux formules logiques P et Q sont logiquement équivalentes si pour toute valuation φ, $[\![P]\!]_\varphi = [\![Q]\!]_\varphi$. On notera alors $P \equiv Q$.
Cela traduit l'égalité sémantique de P et Q, et permet de simplifier les formules.
A ne pas confondre avec $\leftrightarrow$ qui est un connecteur logique.

Quelques équivalences à connaitre

- $\neg(\neg P) \equiv P$
- $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

Equivalence logique

On dit que deux formules logiques P et Q sont logiquement équivalentes si pour toute valuation φ, $[\![P]\!]_\varphi = [\![Q]\!]_\varphi$. On notera alors $P \equiv Q$.
Cela traduit l'égalité sémantique de P et Q, et permet de simplifier les formules.
A ne pas confondre avec $\leftrightarrow$ qui est un connecteur logique.

Quelques équivalences à connaitre

- $\neg(\neg P) \equiv P$
- $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$
- $P \wedge (Q \vee R) \equiv (P \vee Q) \wedge (P \vee R)$

Equivalence logique

On dit que deux formules logiques P et Q sont logiquement équivalentes si pour toute valuation φ, $[\![P]\!]_\varphi = [\![Q]\!]_\varphi$. On notera alors $P \equiv Q$.
Cela traduit l'égalité sémantique de P et Q, et permet de simplifier les formules.
A ne pas confondre avec $\leftrightarrow$ qui est un connecteur logique.

Quelques équivalences à connaitre

- $\neg(\neg P) \equiv P$
- $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$
- $P \wedge (Q \vee R) \equiv (P \vee Q) \wedge (P \vee R)$
- $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$ (loi de De Morgan)

Equivalence logique

On dit que deux formules logiques P et Q sont logiquement équivalentes si pour toute valuation φ, $[\![P]\!]_\varphi = [\![Q]\!]_\varphi$. On notera alors $P \equiv Q$.
Cela traduit l'égalité sémantique de P et Q, et permet de simplifier les formules.
A ne pas confondre avec $\leftrightarrow$ qui est un connecteur logique.

Quelques équivalences à connaitre

- $\neg(\neg P) \equiv P$
- $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$
- $P \wedge (Q \vee R) \equiv (P \vee Q) \wedge (P \vee R)$
- $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$ (loi de De Morgan)
- $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$ (loi de De Morgan)

Equivalence logique

On dit que deux formules logiques P et Q sont logiquement équivalentes si pour toute valuation φ, $[\![P]\!]_\varphi = [\![Q]\!]_\varphi$. On notera alors $P \equiv Q$.
Cela traduit l'égalité sémantique de P et Q, et permet de simplifier les formules.
A ne pas confondre avec $\leftrightarrow$ qui est un connecteur logique.

Quelques équivalences à connaitre

- $\neg(\neg P) \equiv P$
- $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$
- $P \wedge (Q \vee R) \equiv (P \vee Q) \wedge (P \vee R)$
- $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$ (loi de De Morgan)
- $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$ (loi de De Morgan)
- $P \rightarrow Q \equiv \neg P \vee Q$

Exemple

Montrer que :

- $P \vee \neg P \equiv \top$ (tiers exclu)

Exemple

Montrer que :

- $P \vee \neg P \equiv \top$ (tiers exclu)
- $P \rightarrow Q \equiv \neg Q \rightarrow \neg P$ (contraposition)

Exemple

Montrer que :

- $P \vee \neg P \equiv \top$ (tiers exclu)
- $P \to Q \equiv \neg Q \to \neg P$ (contraposition)

Conséquence logique

On dit que qu'une formule Q est conséquence logique d'un ensemble de formules $\Gamma = \{P_1, \dots P_n\}$ si pour toute valuation φ, qui rend vraies les formules $(P_i)_{1 \leqslant i \leqslant n}$ rend aussi vraie Q. On notera alors $\Gamma \vDash Q$.

Exemple

Montrer que :

- $P \vee \neg P \equiv \top$ (tiers exclu)
- $P \to Q \equiv \neg Q \to \neg P$ (contraposition)

Conséquence logique

On dit que qu'une formule Q est conséquence logique d'un ensemble de formules $\Gamma = \{P_1, \dots P_n\}$ si pour toute valuation φ, qui rend vraies les formules $(P_i)_{1 \leqslant i \leqslant n}$ rend aussi vraie Q. On notera alors $\Gamma \models Q$.
A ne pas confondre avec $\to$ qui est un connecteur logique.

Exemple

Montrer que :

- $P \vee \neg P \equiv \top$ (tiers exclu)
- $P \to Q \equiv \neg Q \to \neg P$ (contraposition)

Conséquence logique

On dit que qu'une formule Q est conséquence logique d'un ensemble de formules $\Gamma = \{P_1, \dots P_n\}$ si pour toute valuation φ, qui rend vraies les formules $(P_i)_{1 \leqslant i \leqslant n}$ rend aussi vraie Q. On notera alors $\Gamma \vDash Q$.
A ne pas confondre avec $\to$ qui est un connecteur logique.

Remarques

- Une formule P est est une tautologie ssi $\varnothing \vDash P$, on notera simplement, $\vDash P$.

Exemple

Montrer que :

- $P \vee \neg P \equiv \top$ (tiers exclu)
- $P \to Q \equiv \neg Q \to \neg P$ (contraposition)

Conséquence logique

On dit que qu'une formule Q est conséquence logique d'un ensemble de formules $\Gamma = \{P_1, \dots P_n\}$ si pour toute valuation φ, qui rend vraies les formules $(P_i)_{1 \leqslant i \leqslant n}$ rend aussi vraie Q. On notera alors $\Gamma \vDash Q$.
A ne pas confondre avec $\to$ qui est un connecteur logique.

Remarques

- Une formule P est est une tautologie ssi $\varnothing \vDash P$, on notera simplement, $\vDash P$.
- $P \equiv Q$ ssi $P \vDash Q$ et $Q \vDash P$.

Définitions

- Un littéral est une formule qui est soit une variable propositionnelle p, soit sa négation $\neg p$.

Définitions

- Un littéral est une formule qui est soit une variable propositionnelle p, soit sa négation $\neg p$.
- Une forme normale conjonctive est une formule qui est une conjonction de disjonctions de littéraux.

$$(p_{1,1} \vee p_{1,2} \ldots \vee p_{1,k_1}) \wedge \underbrace{(p_{2,1} \vee p_{2,2} \ldots \vee p_{2,k_2})}_{\text{une clause}} \wedge \ldots \wedge (p_{m,1} \vee p_{m,2} \ldots \vee p_{m,k_m})$$

Définitions

- Un littéral est une formule qui est soit une variable propositionnelle p, soit sa négation $\neg p$.
- Une forme normale conjonctive est une formule qui est une conjonction de disjonctions de littéraux.
$$(p_{1,1} \vee p_{1,2} \ldots \vee p_{1,k_1}) \wedge \underbrace{(p_{2,1} \vee p_{2,2} \ldots \vee p_{2,k_2})}_{\text{une clause}} \wedge \ldots \wedge (p_{m,1} \vee p_{m,2} \ldots \vee p_{m,k_m})$$
- Une forme normale disjonctive est une formule qui est une disjonction de conjonctions de littéraux.
$$(p_{1,1} \wedge p_{1,2} \ldots \wedge p_{1,k_1}) \vee (p_{2,1} \wedge p_{2,2} \ldots \wedge p_{2,k_2}) \vee \ldots \vee (p_{m,1} \wedge p_{m,2} \ldots \wedge p_{m,k_m})$$

Définitions

- Un littéral est une formule qui est soit une variable propositionnelle p, soit sa négation $\neg p$.
- Une forme normale conjonctive est une formule qui est une conjonction de disjonctions de littéraux.
$(p_{1,1} \vee p_{1,2} \ldots \vee p_{1,k_1}) \wedge \underbrace{(p_{2,1} \vee p_{2,2} \ldots \vee p_{2,k_2})}_{\text{une clause}} \wedge \ldots \wedge (p_{m,1} \vee p_{m,2} \ldots \vee p_{m,k_m})$
- Une forme normale disjonctive est une formule qui est une disjonction de conjonctions de littéraux.
$(p_{1,1} \wedge p_{1,2} \ldots \wedge p_{1,k_1}) \vee (p_{2,1} \wedge p_{2,2} \ldots \wedge p_{2,k_2}) \vee \ldots \vee (p_{m,1} \wedge p_{m,2} \ldots \wedge p_{m,k_m})$

Exemple

$(p \wedge q \wedge \neg r) \vee (\neg p \wedge r) \vee (p \wedge \neg q \wedge \neg r)$ est une FND.

Propositions

- Pour tout formule logique P, il existe une FNC Q et une FND R telles que $P \equiv Q \equiv R$.

Propositions

- Pour tout formule logique P, il existe une FNC Q et une FND R telles que $P \equiv Q \equiv R$.
- On dispose d'un algorithme pour calculer une forme normale :

Propositions

- Pour tout formule logique P, il existe une FNC Q et une FND R telles que $P \equiv Q \equiv R$.
- On dispose d'un algorithme pour calculer une forme normale :
  1. supprimer les $\perp$ et les $\top$

Propositions

- Pour tout formule logique P, il existe une FNC Q et une FND R telles que $P \equiv Q \equiv R$.
- On dispose d'un algorithme pour calculer une forme normale :
  1. supprimer les $\perp$ et les $\top$
  2. Remplacer $\rightarrow$ et $\leftrightarrow$ par des formules sémantiquement égales n'utilisant pas ces connecteurs : $p \rightarrow q \equiv \neg p \vee q$ et $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$.

Propositions

- Pour tout formule logique P, il existe une FNC Q et une FND R telles que $P \equiv Q \equiv R$.
- On dispose d'un algorithme pour calculer une forme normale :
  1. supprimer les $\perp$ et les $\top$
  2. Remplacer $\rightarrow$ et $\leftrightarrow$ par des formules sémantiquement égales n'utilisant pas ces connecteurs : $p \rightarrow q \equiv \neg p \vee q$ et $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$.
  3. Utiliser les lois de De Morgan afin de faire descendre les $\neg$ au niveau des feuilles de l'arbre syntaxique

Propositions

- Pour tout formule logique P, il existe une FNC Q et une FND R telles que $P \equiv Q \equiv R$.
- On dispose d'un algorithme pour calculer une forme normale :
  1. supprimer les $\perp$ et les $\top$
  2. Remplacer $\rightarrow$ et $\leftrightarrow$ par des formules sémantiquement égales n'utilisant pas ces connecteurs : $p \rightarrow q \equiv \neg p \vee q$ et $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$.
  3. Utiliser les lois de De Morgan afin de faire descendre les $\neg$ au niveau des feuilles de l'arbre syntaxique
  4. Appliquer les propriétés d'associativité et de distributivité des connecteurs $\wedge$ et $\vee$.

Propositions

- Pour tout formule logique P, il existe une $\textsc{fnc}$ Q et une $\textsc{fnd}$ R telles que $P \equiv Q \equiv R$.
- On dispose d'un algorithme pour calculer une forme normale :
  1. supprimer les $\perp$ et les $\top$
  2. Remplacer $\rightarrow$ et $\leftrightarrow$ par des formules sémantiquement égales n'utilisant pas ces connecteurs : $p \rightarrow q \equiv \neg p \vee q$ et $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$.
  3. Utiliser les lois de De Morgan afin de faire descendre les $\neg$ au niveau des feuilles de l'arbre syntaxique
  4. Appliquer les propriétés d'associativité et de distributivité des connecteurs $\wedge$ et $\vee$.
  5. simplifier les doublons éventuelles ($v \wedge \neg v \equiv \perp$ et $v \vee \neg v = \top$)

Propositions

- Pour tout formule logique P, il existe une FNC Q et une FND R telles que $P \equiv Q \equiv R$.
- On dispose d'un algorithme pour calculer une forme normale :
  1. supprimer les $\perp$ et les $\top$
  2. Remplacer $\rightarrow$ et $\leftrightarrow$ par des formules sémantiquement égales n'utilisant pas ces connecteurs : $p \rightarrow q \equiv \neg p \vee q$ et $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$.
  3. Utiliser les lois de De Morgan afin de faire descendre les $\neg$ au niveau des feuilles de l'arbre syntaxique
  4. Appliquer les propriétés d'associativité et de distributivité des connecteurs $\wedge$ et $\vee$.
  5. simplifier les doublons éventuelles ($v \wedge \neg v \equiv \perp$ et $v \vee \neg v = \top$)
- Une autre méthode consiste à utiliser la table de vérité.

Exemple

Mise sous forme normale de $P = (p \rightarrow q) \leftrightarrow \neg r$

- Méthode 1 : utilisation de l'algorithme

Exemple

Mise sous forme normale de $P = (p \to q) \leftrightarrow \neg r$

- Méthode 1 : utilisation de l'algorithme
  $P = ((p \to q) \wedge \neg r) \vee (\neg(p \to q) \wedge \neg\neg r)$ (équivalence sémantique de $\leftrightarrow$)

Exemple

Mise sous forme normale de $P = (p \to q) \leftrightarrow \neg r$

- Méthode 1 : utilisation de l'algorithme
  $P = ((p \to q) \land \neg r) \lor (\neg(p \to q) \land \neg\neg r)$ (équivalence sémantique de $\leftrightarrow$)
  $P = ((\neg p \lor q) \land \neg r) \lor (\neg(\neg p \lor q) \land r)$ (équivalence sémantique de $\to$)

Exemple

Mise sous forme normale de $P = (p \rightarrow q) \leftrightarrow \neg r$

- Méthode 1 : utilisation de l'algorithme
  $P = ((p \rightarrow q) \wedge \neg r) \vee (\neg(p \rightarrow q) \wedge \neg\neg r)$ (équivalence sémantique de $\leftrightarrow$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee (\neg(\neg p \vee q) \wedge r)$ (équivalence sémantique de $\rightarrow$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee ((p \wedge \neg q) \wedge r)$ (lois de DeMorgan)
  $P = (\neg p \wedge \neg r) \vee (q \wedge \neg r) \vee (p \wedge \neg q \wedge r)$ (distributivité et associativité)

Exemple

Mise sous forme normale de $P = (p \to q) \leftrightarrow \neg r$

- Méthode 1 : utilisation de l'algorithme
  $P = ((p \to q) \wedge \neg r) \vee (\neg(p \to q) \wedge \neg\neg r)$ (équivalence sémantique de $\leftrightarrow$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee (\neg(\neg p \vee q) \wedge r)$ (équivalence sémantique de $\to$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee ((p \wedge \neg q) \wedge r)$ (lois de DeMorgan)
  $P = (\neg p \wedge \neg r) \vee (q \wedge \neg r) \vee (p \wedge \neg q \wedge r)$ (distributivité et associativité)
- Méthode 2 : utilisation de la table de vérité

Exemple

Mise sous forme normale de $P = (p \rightarrow q) \leftrightarrow \neg r$

- Méthode 1 : utilisation de l'algorithme
  $P = ((p \rightarrow q) \wedge \neg r) \vee (\neg(p \rightarrow q) \wedge \neg\neg r)$ (équivalence sémantique de $\leftrightarrow$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee (\neg(\neg p \vee q) \wedge r)$ (équivalence sémantique de $\rightarrow$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee ((p \wedge \neg q) \wedge r)$ (lois de DeMorgan)
  $P = (\neg p \wedge \neg r) \vee (q \wedge \neg r) \vee (p \wedge \neg q \wedge r)$ (distributivité et associativité)
- Méthode 2 : utilisation de la table de vérité

p	q	r	$p \rightarrow q$	$\neg r$	P
F	F	F	V	V	V
F	F	V	V	F	F
F	V	F	V	V	V
F	V	V	V	F	F
V	F	F	F	V	F
V	F	V	F	F	V
V	V	F	V	V	V
V	V	V	V	F	F

Exemple

Mise sous forme normale de $P = (p \to q) \leftrightarrow \neg r$

- Méthode 1 : utilisation de l'algorithme
  $P = ((p \to q) \wedge \neg r) \vee (\neg(p \to q) \wedge \neg\neg r)$ (équivalence sémantique de $\leftrightarrow$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee (\neg(\neg p \vee q) \wedge r)$ (équivalence sémantique de $\to$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee ((p \wedge \neg q) \wedge r)$ (lois de DeMorgan)
  $P = (\neg p \wedge \neg r) \vee (q \wedge \neg r) \vee (p \wedge \neg q \wedge r)$ (distributivité et associativité)
- Méthode 2 : utilisation de la table de vérité

p	q	r	$p \to q$	$\neg r$	P
F	F	F	V	V	V
F	F	V	V	F	F
F	V	F	V	V	V
F	V	V	V	F	F
V	F	F	F	V	F
V	F	V	F	F	V
V	V	F	V	V	V
V	V	V	V	F	F

$\longrightarrow (\neg p \wedge \neg q \wedge \neg r) \vee$

Exemple

Mise sous forme normale de $P = (p \to q) \leftrightarrow \neg r$

- Méthode 1 : utilisation de l'algorithme
  $P = ((p \to q) \wedge \neg r) \vee (\neg(p \to q) \wedge \neg\neg r)$ (équivalence sémantique de $\leftrightarrow$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee (\neg(\neg p \vee q) \wedge r)$ (équivalence sémantique de $\to$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee ((p \wedge \neg q) \wedge r)$ (lois de DeMorgan)
  $P = (\neg p \wedge \neg r) \vee (q \wedge \neg r) \vee (p \wedge \neg q \wedge r)$ (distributivité et associativité)
- Méthode 2 : utilisation de la table de vérité

p	q	r	$p \to q$	$\neg r$	P	
F	F	F	V	V	V	$\longrightarrow (\neg p \wedge \neg q \wedge \neg r) \vee$
F	F	V	V	F	F	
F	V	F	V	V	V	$\longrightarrow (\neg p \wedge q \wedge \neg r) \vee$
F	V	V	V	F	F	
V	F	F	F	V	F	
V	F	V	F	F	V	
V	V	F	V	V	V	
V	V	V	V	F	F	

Exemple

Mise sous forme normale de $P = (p \to q) \leftrightarrow \neg r$

- Méthode 1 : utilisation de l'algorithme
  $P = ((p \to q) \wedge \neg r) \vee (\neg(p \to q) \wedge \neg\neg r)$ (équivalence sémantique de $\leftrightarrow$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee (\neg(\neg p \vee q) \wedge r)$ (équivalence sémantique de $\to$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee ((p \wedge \neg q) \wedge r)$ (lois de DeMorgan)
  $P = (\neg p \wedge \neg r) \vee (q \wedge \neg r) \vee (p \wedge \neg q \wedge r)$ (distributivité et associativité)
- Méthode 2 : utilisation de la table de vérité

p	q	r	$p \to q$	$\neg r$	P	
F	F	F	V	V	V	$\longrightarrow (\neg p \wedge \neg q \wedge \neg r)\vee$
F	F	V	V	F	F	
F	V	F	V	V	V	$\longrightarrow (\neg p \wedge q \wedge \neg r)\vee$
F	V	V	V	F	F	
V	F	F	F	V	F	
V	F	V	F	F	V	$\longrightarrow (p \wedge \neg q \wedge r)\vee$
V	V	F	V	V	V	
V	V	V	V	F	F	

Exemple

Mise sous forme normale de $P = (p \to q) \leftrightarrow \neg r$

- Méthode 1 : utilisation de l'algorithme
  $P = ((p \to q) \wedge \neg r) \vee (\neg(p \to q) \wedge \neg\neg r)$ (équivalence sémantique de $\leftrightarrow$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee (\neg(\neg p \vee q) \wedge r)$ (équivalence sémantique de $\to$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee ((p \wedge \neg q) \wedge r)$ (lois de DeMorgan)
  $P = (\neg p \wedge \neg r) \vee (q \wedge \neg r) \vee (p \wedge \neg q \wedge r)$ (distributivité et associativité)
- Méthode 2 : utilisation de la table de vérité

p	q	r	$p \to q$	$\neg r$	P	
F	F	F	V	V	V	$\longrightarrow (\neg p \wedge \neg q \wedge \neg r)\vee$
F	F	V	V	F	F	
F	V	F	V	V	V	$\longrightarrow (\neg p \wedge q \wedge \neg r)\vee$
F	V	V	V	F	F	
V	F	F	F	V	F	
V	F	V	F	F	V	$\longrightarrow (p \wedge \neg q \wedge r)\vee$
V	V	F	V	V	V	$\longrightarrow (p \wedge q \wedge \neg r)$
V	V	V	V	F	F	

Définitions

- Un problème de décision sur un ensemble E, est une question sur les éléments de E à laquelle on répond par *oui* ou *non*.
  Par exemple sur $\mathbb{N}$, savoir si un entier n est premier ou non est un problème de décision.

Définitions

- Un problème de décision sur un ensemble E, est une question sur les éléments de E à laquelle on répond par *oui* ou *non*.
  Par exemple sur $\mathbb{N}$, savoir si un entier n est premier ou non est un problème de décision.
- La théorie de la calculabilité étudie l'existence ou non d'un algorithme capable de répondre à un problème de décision.
  Par exemple le problème de l'arrête est indécidable

Définitions

- Un problème de décision sur un ensemble E, est une question sur les éléments de E à laquelle on répond par *oui* ou *non*.
  Par exemple sur $\mathbb{N}$, savoir si un entier n est premier ou non est un problème de décision.
- La théorie de la calculabilité étudie l'existence ou non d'un algorithme capable de répondre à un problème de décision.
  Par exemple le problème de l'arrête est indécidable
- La théorie de la complexité s'intéresse à la complexité des algorithmes lorsqu'un problème de décision est décidable.

Problème SAT

Le problème SAT (pour satisfiabilité) est le problème de savoir si une formule logique P définie sur un ensemble de variable logique $V = \{p_1, \dots p_n\}$ est satisfiable ou non.

Algorithme de Quine

Pour tester la satisfiabilité d'une formule logique, on peut construire sa table de vérité ou utiliser l'algorithme de Quine. Soit P une formule contenant les variables logiques $p_1, \dots p_n$.

- On fixe $\varphi(p_1) = F$ et on teste récursivement la satisfiabilité de P dans laquelle toutes les occurences de p_1 sont remplacées par $\perp$ (notée $P[\perp/p_1]$).
- En cas d'échec, on fixe $\varphi(p_1) = V$ et on teste récursivement la satisfiabilité de $P[\top/p_1]$.
- En cas d'échec la formule n'est pas satisfiable.

Exemple

Dérouler l'algorithme de Quine sur $P = (p \lor q) \land (\neg q \lor \neg r) \land (\neg p \lor r) \land (p \lor r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :

Exemple

Dérouler l'algorithme de Quine sur $P = (p \ \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :
  $P[\bot/p] = (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r)$

Exemple

Dérouler l'algorithme de Quine sur $P = (p \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :

$$\begin{aligned} P[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \end{aligned}$$

Exemple

Dérouler l'algorithme de Quine sur $P = (p \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :

$$\begin{aligned} P[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$

Exemple

Dérouler l'algorithme de Quine sur $P = (p \ \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :
  $$\begin{aligned} P[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur F à q et on teste la satisfiabilité de :

Exemple

Dérouler l'algorithme de Quine sur $P = (p \lor q) \land (\neg q \lor \neg r) \land (\neg p \lor r) \land (p \lor r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :
  $$\begin{aligned} P[\bot/p] &= (\bot \lor q) \land (\neg q \lor \neg r) \land (\neg \bot \lor r) \land (\bot \lor r) \\ &= q \land (\neg q \lor \neg r) \land \top \land r \\ &= q \land (\neg q \lor \neg r) \land r \end{aligned}$$
  - On affecte la valeur F à q et on teste la satisfiabilité de :
    $\bot \land (\neg \bot \lor \neg r) \land r$ qui est non satisfiable

Exemple

Dérouler l'algorithme de Quine sur $P = (p \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :
  $$\begin{aligned} P[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur F à q et on teste la satisfiabilité de : $\bot \wedge (\neg \bot \vee \neg r) \wedge r$ qui est non satisfiable
  - On affecte la valeur V à q et on teste la satisfiabilité de :

Exemple

Dérouler l'algorithme de Quine sur $P = (p \lor q) \land (\neg q \lor \neg r) \land (\neg p \lor r) \land (p \lor r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :
  $$\begin{aligned} P[\bot/p] &= (\bot \lor q) \land (\neg q \lor \neg r) \land (\neg \bot \lor r) \land (\bot \lor r) \\ &= q \land (\neg q \lor \neg r) \land \top \land r \\ &= q \land (\neg q \lor \neg r) \land r \end{aligned}$$
  - On affecte la valeur F à q et on teste la satisfiabilité de :
    $\bot \land (\neg \bot \lor \neg r) \land r$ qui est non satisfiable
  - On affecte la valeur V à q et on teste la satisfiabilité de :
    $\top \land (\neg \top \lor \neg r) \land r = \neg r \land r$ donc non satisfiable.

Exemple

Dérouler l'algorithme de Quine sur $P = (p \ \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :
  $$\begin{aligned} P[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur F à q et on teste la satisfiabilité de :
    $\bot \wedge (\neg \bot \vee \neg r) \wedge r$ qui est non satisfiable
  - On affecte la valeur V à q et on teste la satisfiabilité de :
    $\top \wedge (\neg \top \vee \neg r) \wedge r = \neg r \wedge r$ donc non satisfiable.
- On affecte la valeur V à p et on teste la satisfiabilité de :

Exemple

Dérouler l'algorithme de Quine sur $P = (p \ \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :
  $$\begin{aligned} P[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur F à q et on teste la satisfiabilité de : $\bot \wedge (\neg \bot \vee \neg r) \wedge r$ qui est non satisfiable
  - On affecte la valeur V à q et on teste la satisfiabilité de : $\top \wedge (\neg \top \vee \neg r) \wedge r = \neg r \wedge r$ donc non satisfiable.
- On affecte la valeur V à p et on teste la satisfiabilité de :
  $P[\top/p] = (\top \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \top \vee r) \wedge (\top \vee r)$

Exemple

Dérouler l'algorithme de Quine sur $P = (p \ \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :
  $$\begin{aligned} P[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg\bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur F à q et on teste la satisfiabilité de : $\bot \wedge (\neg\bot \vee \neg r) \wedge r$ qui est non satisfiable
  - On affecte la valeur V à q et on teste la satisfiabilité de : $\top \wedge (\neg\top \vee \neg r) \wedge r = \neg r \wedge r$ donc non satisfiable.
- On affecte la valeur V à p et on teste la satisfiabilité de :
  $P[\top/p] = (\top \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg\top \vee r) \wedge (\top \vee r)$
  $P[\top/p] = (\neg q \vee \neg r) \wedge r$

Exemple

Dérouler l'algorithme de Quine sur $P = (p \ \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :
  $$\begin{aligned} P[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur F à q et on teste la satisfiabilité de : $\bot \wedge (\neg \bot \vee \neg r) \wedge r$ qui est non satisfiable
  - On affecte la valeur V à q et on teste la satisfiabilité de : $\top \wedge (\neg \top \vee \neg r) \wedge r = \neg r \wedge r$ donc non satisfiable.
- On affecte la valeur V à p et on teste la satisfiabilité de :
  $P[\top/p] = (\top \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \top \vee r) \wedge (\top \vee r)$
  $P[\top/p] = (\neg q \vee \neg r) \wedge r$
  - On affecte la valeur F à q et on teste la satisfiabilité de :

Exemple

Dérouler l'algorithme de Quine sur $P = (p \ \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :
  $$\begin{aligned} P[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur F à q et on teste la satisfiabilité de : $\bot \wedge (\neg \bot \vee \neg r) \wedge r$ qui est non satisfiable
  - On affecte la valeur V à q et on teste la satisfiabilité de : $\top \wedge (\neg \top \vee \neg r) \wedge r = \neg r \wedge r$ donc non satisfiable.
- On affecte la valeur V à p et on teste la satisfiabilité de :
  $P[\top/p] = (\top \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \top \vee r) \wedge (\top \vee r)$
  $P[\top/p] = (\neg q \vee \neg r) \wedge r$
  - On affecte la valeur F à q et on teste la satisfiabilité de : $(\neg \bot \vee \neg r) \wedge r = r$ qui est satisfiable.

Exemple

Dérouler l'algorithme de Quine sur $P = (p \ \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee r)$

- On affecte la valeur F à p et on teste la satisfiabilité de :
  $$\begin{aligned} P[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur F à q et on teste la satisfiabilité de : $\bot \wedge (\neg \bot \vee \neg r) \wedge r$ qui est non satisfiable
  - On affecte la valeur V à q et on teste la satisfiabilité de : $\top \wedge (\neg \top \vee \neg r) \wedge r = \neg r \wedge r$ donc non satisfiable.
- On affecte la valeur V à p et on teste la satisfiabilité de :
  $P[\top/p] = (\top \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \top \vee r) \wedge (\top \vee r)$
  $P[\top/p] = (\neg q \vee \neg r) \wedge r$
  - On affecte la valeur F à q et on teste la satisfiabilité de : $(\neg \bot \vee \neg r) \wedge r = r$ qui est satisfiable.

On dispose à la fin d'un valuation φ telle que $[\![P]\!]_\varphi = V$: $\varphi(p) = V, \varphi(q) = F$ et $\varphi(r) = V$.