C16 Logique

1. Syntaxe des formules logiques

Définition

Soit $\mathbb{1}$ un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur $\mathbb{1}$ par :

Définition

Soit $\mathbb{1}$ un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur $\mathbb{1}$ par :

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

Définition

Soit $\mathbb{1}$ un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur $\mathbb{1}$ par :

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

On notera avec les lettres minuscules p, q, … *les variables logiques* ie les éléments de $\mathbb{1}$ et avec les lettres majuscules $\mathbb{0}$, G, … *les formules logiques* ie les éléments de P.

Définition

Soit $\mathbb{1}$ un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur $\mathbb{1}$ 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* $\neg$: $F \mapsto \neg F$

On notera avec les lettres minuscules p, q, ... *les variables logiques* ie les éléments de $\mathbb{1}$ et avec les lettres majuscules $\mathbb{0}$, G, ... *les formules logiques* ie les éléments de P.

Définition

Soit $\mathbb{1}$ un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur $\mathbb{1}$ 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* $\neg$: $F \mapsto \neg F$
  - *conjonction* $\wedge$: $F, G \mapsto (F \wedge G)$

On notera avec les lettres minuscules p, q, ... *les variables logiques* ie les éléments de $\mathbb{1}$ et avec les lettres majuscules $\mathbb{0}$, G, ... *les formules logiques* ie les éléments de P.

Définition

Soit 𝟙 un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur 𝟙 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* $\neg$: $F \mapsto \neg F$
  - *conjonction* $\wedge$: $F, G \mapsto (F \wedge G)$
  - *disjonction* $\vee$: $F, G \mapsto (F \vee G)$

On notera avec les lettres minuscules p, q, … *les variables logiques* ie les éléments de 𝟙 et avec les lettres majuscules 𝟘, G, … *les formules logiques* ie les éléments de P.

Définition

Soit $\mathbb{1}$ un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur $\mathbb{1}$ 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* $\neg$: $F \mapsto \neg F$
  - *conjonction* $\wedge$: $F, G \mapsto (F \wedge G)$
  - *disjonction* $\vee$: $F, G \mapsto (F \vee G)$
  - *implication* $\rightarrow$: $F, G \mapsto (F \rightarrow G)$

On notera avec les lettres minuscules p, q, ... *les variables logiques* ie les éléments de $\mathbb{1}$ et avec les lettres majuscules $\mathbb{0}$, G, ... *les formules logiques* ie les éléments de P.

Définition

Soit 𝟙 un ensemble au plus dénombrable de variables logiques noté $V = \{p, q, r, \dots\}$. On définit inductivement l'ensemble P des formules logiques sur 𝟙 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* $\neg$: $F \mapsto \neg F$
  - *conjonction* $\wedge$: $F, G \mapsto (F \wedge G)$
  - *disjonction* $\vee$: $F, G \mapsto (F \vee G)$
  - *implication* $\rightarrow$: $F, G \mapsto (F \rightarrow G)$
  - *équivalence* $\leftrightarrow$: $F, G \mapsto (F \leftrightarrow G)$

On notera avec les lettres minuscules p, q, ... *les variables logiques* ie les éléments de 𝟙 et avec les lettres majuscules 𝟘, G, ... *les formules logiques* ie les éléments de P.

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 $(\neg p \vee \neg q) \wedge r$.
- $\wedge p \neg p q$ 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 :

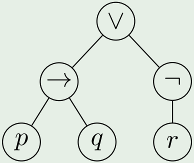
Arbre syntaxique

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

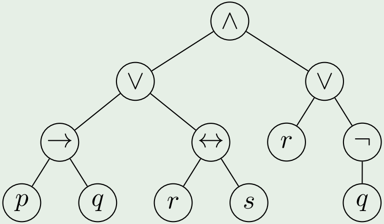
Exemples

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



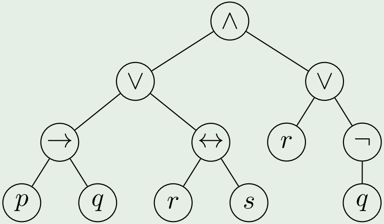
Exemple

- Quelle est la formule logique ayant pour représentation arborescente :



Exemple

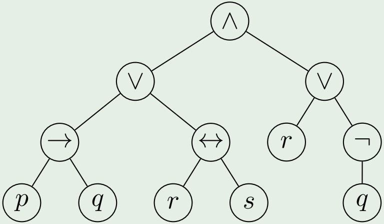
- Quelle est la formule logique ayant pour représentation arborescente :



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

Exemple

- Quelle est la formule logique ayant pour représentation arborescente :



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

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

Hauteur, taille et sous formule

Etant donnée une formule logique notée $\mathbb{0}$,

Hauteur, taille et sous formule

Etant donnée une formule logique notée $\mathbb{0}$,

- la hauteur de $\mathbb{0}$ est la hauteur de l'arbre syntaxique associé

Hauteur, taille et sous formule

Etant donnée une formule logique notée $\mathbb{0}$,

- la hauteur de $\mathbb{0}$ est la hauteur de l'arbre syntaxique associé
- la taille de $\mathbb{0}$ est le nombre de noeuds de l'arbre syntaxiqué associé

Hauteur, taille et sous formule

Etant donnée une formule logique notée $\mathbb{O}$,

- la hauteur de $\mathbb{O}$ est la hauteur de l'arbre syntaxique associé
- la taille de $\mathbb{O}$ est le nombre de noeuds de l'arbre syntaxiqué associé
- Une sous formule de $\mathbb{O}$ 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
 | Equ 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.

```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
| Equ 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
| Equ 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 représentée 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.

```
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
| Equ 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 $(p \wedge \neg q) \to r$ est alors représentée par :

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

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* :

```
let rec taille fl =
  match fl with
  | Top | Bot | Var _ -> 1
  | Non sf -> 1 + taille sf
  | Ou (sf1, sf2) -> 1 + taille sf1 + taille sf2
  | Et (sf1, sf2) -> 1 + taille sf1 + taille sf2
  | Imp (sf1, sf2) -> 1 + taille sf1 + taille sf2
  | Equ (sf1, sf2) -> 1 + taille sf1 + taille sf2;;
```

Valuation

On note $\mathbb{B} = \{\mathbb{0}, \mathbb{1}\}$, l'ensemble des valeurs de vérités
Une valuation est une attribution de l'une des deux valeurs de vérités à chaque variable propositionnelle. Une valuation φ est une donc une application de $\mathbb{1}$ de B.

Valuation

On note $\mathbb{B} = \{\mathbb{0}, \mathbb{1}\}$, l'ensemble des valeurs de vérités
Une valuation est une attribution de l'une des deux valeurs de vérités à chaque variable propositionnelle. Une valuation φ est une donc une application de $\mathbb{1}$ de B.

Exemple

Si l'ensemble des variables propositionnelle est $V = \{p, q, r\}$, une valuation possible est :
$\varphi : V \mapsto \mathbb{B}$, avec $\varphi(p) = \mathbb{1}$, $\varphi(q) = \mathbb{0}$ et $\varphi(r) = \mathbb{0}$.

Valuation

On note $\mathbb{B} = \{\mathbb{0}, \mathbb{1}\}$, l'ensemble des valeurs de vérités
Une valuation est une attribution de l'une des deux valeurs de vérités à chaque variable propositionnelle. Une valuation φ est une donc une application de $\mathbb{1}$ de B.

Exemple

Si l'ensemble des variables propositionnelle est $V = \{p, q, r\}$, une valuation possible est :
$\varphi : V \mapsto \mathbb{B}$, avec $\varphi(p) = \mathbb{1}$, $\varphi(q) = \mathbb{0}$ et $\varphi(r) = \mathbb{0}$.

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	$f_\wedge(x,y)$
$\mathbb{0}$	$\mathbb{0}$	$\mathbb{0}$
$\mathbb{0}$	$\mathbb{1}$	$\mathbb{0}$
$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$
$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$

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

x	y	$f_\vee(x,y)$
$\mathbb{0}$	$\mathbb{0}$	$\mathbb{0}$
$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$
$\mathbb{1}$	$\mathbb{0}$	$\mathbb{1}$
$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$

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

x	y	$f_\rightarrow(x,y)$
$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$
$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$
$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$
$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$

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

x	y	$f_\leftrightarrow(x,y)$
$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$
$\mathbb{0}$	$\mathbb{1}$	$\mathbb{0}$
$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$
$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$

Et $f_\neg : \mathbb{B} \mapsto \mathbb{B}$, définie par $f_\neg(\mathbb{0}) = \mathbb{1}$ et $f_\neg(\mathbb{1}) = \mathbb{0}$.

Valeur de vérité d'une formule

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

Valeur de vérité d'une formule

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

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

Valeur de vérité d'une formule

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

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

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = \mathbb{1}$, $\varphi(q) = \mathbb{0}$ et $\varphi(r) = \mathbb{0}$ on peut alors déterminer la valeur de vérité de la formule logique $F = (p \to q) \land (\neg p \lor r)$ associée à cette valuation :

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = \mathbb{1}$, $\varphi(q) = \mathbb{0}$ et $\varphi(r) = \mathbb{0}$ on peut alors déterminer la valeur de vérité de la formule logique $F = (p \to q) \land (\neg p \lor r)$ associée à cette valuation :
$[\![F]\!]_\varphi = f_\land([\![p \to q]\!]_\varphi, [\![\neg p \lor r]\!]_\varphi)$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = \mathbb{1}$, $\varphi(q) = \mathbb{0}$ et $\varphi(r) = \mathbb{0}$ on peut alors déterminer la valeur de vérité de la formule logique $F = (p \to q) \wedge (\neg p \vee r)$ associée à cette valuation :
$[\![F]\!]_\varphi = f_\wedge([\![p \to q]\!]_\varphi, [\![\neg p \vee r]\!]_\varphi)$
$[\![F]\!]_\varphi = f_\wedge(f_\to([\![p]\!]_\varphi, [\![q]\!]_\varphi)), f_\vee([\![\neg p]\!]_\varphi, [\![r]\!]_\varphi)$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = \mathbb{1}$, $\varphi(q) = \mathbb{0}$ et $\varphi(r) = \mathbb{0}$ on peut alors déterminer la valeur de vérité de la formule logique $F = (p \to q) \wedge (\neg p \vee r)$ associée à cette valuation :

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

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

$[\![F]\!]_\varphi = f_\wedge(f_\to(\mathbb{1}, \mathbb{0}), f_\vee(\mathbb{0}, \mathbb{0}))$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = \mathbb{1}$, $\varphi(q) = \mathbb{0}$ et $\varphi(r) = \mathbb{0}$ on peut alors déterminer la valeur de vérité de la formule logique $F = (p \to q) \wedge (\neg p \vee r)$ associée à cette valuation :

$\llbracket F \rrbracket_\varphi = f_\wedge(\llbracket p \to q \rrbracket_\varphi, \llbracket \neg p \vee r \rrbracket_\varphi)$

$\llbracket F \rrbracket_\varphi = f_\wedge(f_\to(\llbracket p \rrbracket_\varphi, \llbracket q \rrbracket_\varphi)), f_\vee(\llbracket \neg p \rrbracket_\varphi, \llbracket r \rrbracket_\varphi)$

$\llbracket F \rrbracket_\varphi = f_\wedge(f_\to(\mathbb{1}, \mathbb{0}), f_\vee(\mathbb{0}, \mathbb{0}))$

$\llbracket F \rrbracket_\varphi = f_\wedge(\mathbb{0}, \mathbb{0})$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = \mathbb{1}$, $\varphi(q) = \mathbb{0}$ et $\varphi(r) = \mathbb{0}$ on peut alors déterminer la valeur de vérité de la formule logique $F = (p \to q) \wedge (\neg p \vee r)$ associée à cette valuation :

$[\![F]\!]_\varphi = f_\wedge([\![p \to q]\!]_\varphi, [\![\neg p \vee r]\!]_\varphi)$
$[\![F]\!]_\varphi = f_\wedge(f_\to([\![p]\!]_\varphi, [\![q]\!]_\varphi)), f_\vee([\![\neg p]\!]_\varphi, [\![r]\!]_\varphi)$
$[\![F]\!]_\varphi = f_\wedge(f_\to(\mathbb{1}, \mathbb{0}), f_\vee(\mathbb{0}, \mathbb{0}))$
$[\![F]\!]_\varphi = f_\wedge(\mathbb{0}, \mathbb{0})$
$[\![F]\!]_\varphi = \mathbb{0}$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = \mathbb{1}$, $\varphi(q) = \mathbb{0}$ et $\varphi(r) = \mathbb{0}$ on peut alors déterminer la valeur de vérité de la formule logique $F = (p \to q) \wedge (\neg p \vee r)$ associée à cette valuation :

$[\![F]\!]_\varphi = f_\wedge([\![p \to q]\!]_\varphi, [\![\neg p \vee r]\!]_\varphi)$
$[\![F]\!]_\varphi = f_\wedge(f_\to([\![p]\!]_\varphi, [\![q]\!]_\varphi)), f_\vee([\![\neg p]\!]_\varphi, [\![r]\!]_\varphi)$
$[\![F]\!]_\varphi = f_\wedge(f_\to(\mathbb{1}, \mathbb{0}), f_\vee(\mathbb{0}, \mathbb{0}))$
$[\![F]\!]_\varphi = f_\wedge(\mathbb{0}, \mathbb{0})$
$[\![F]\!]_\varphi = \mathbb{0}$

- Donner la valeur de vérité de cette proposition pour la valuation φ' définie par : $\varphi'(p) = \mathbb{0}, \varphi'(q) = \mathbb{0}, \varphi'(r) = \mathbb{0}$

Exemple

Sur $V = \{p, q, r\}$, on considère la valuation $\varphi : V \mapsto \mathbb{B}$, telle que $\varphi(p) = \mathbb{1}$, $\varphi(q) = \mathbb{0}$ et $\varphi(r) = \mathbb{0}$ on peut alors déterminer la valeur de vérité de la formule logique $F = (p \to q) \land (\neg p \lor r)$ associée à cette valuation :

$[\![F]\!]_\varphi = f_\land([\![p \to q]\!]_\varphi, [\![\neg p \lor r]\!]_\varphi)$
$[\![F]\!]_\varphi = f_\land(f_\to([\![p]\!]_\varphi, [\![q]\!]_\varphi)), f_\lor([\![\neg p]\!]_\varphi, [\![r]\!]_\varphi)$
$[\![F]\!]_\varphi = f_\land(f_\to(\mathbb{1}, \mathbb{0}), f_\lor(\mathbb{0}, \mathbb{0}))$
$[\![F]\!]_\varphi = f_\land(\mathbb{0}, \mathbb{0})$
$[\![F]\!]_\varphi = \mathbb{0}$

- Donner la valeur de vérité de cette proposition pour la valuation φ' définie par : $\varphi'(p) = \mathbb{0}, \varphi'(q) = \mathbb{0}, \varphi'(r) = \mathbb{0}$
- Déterminer la valeur de vérité de $((p \to q) \lor (r \leftrightarrow s)) \land (r \lor (\neg q))$ pour la valuation φ.

Tautologie, antilogie, satisfiabilité

- Une formule est une tautologie si sa valeur de vérité est $\mathbb{1}$ pour toute valuation. c'est-à-dire que F est une tautologie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![F]\!]_\varphi = \mathbb{1}$.

Tautologie, antilogie, satisfiabilité

- Une formule est une tautologie si sa valeur de vérité est $\mathbb{1}$ pour toute valuation. c'est-à-dire que F est une tautologie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![F]\!]_\varphi = \mathbb{1}$.
- Une formule est une antilogie si sa valeur de vérité est $\mathbb{0}$ pour toute valuation. c'est-à-dire que F est une antilogie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![F]\!]_\varphi = \mathbb{0}$.

Tautologie, antilogie, satisfiabilité

- Une formule est une tautologie si sa valeur de vérité est $\mathbb{1}$ pour toute valuation. c'est-à-dire que F est une tautologie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![F]\!]_\varphi = \mathbb{1}$.
- Une formule est une antilogie si sa valeur de vérité est $\mathbb{0}$ pour toute valuation. c'est-à-dire que F est une antilogie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![F]\!]_\varphi = \mathbb{0}$.
- Une formule est satisfiable s'il existe une valuation pour laquelle sa valeur de vérité est $\mathbb{1}$. c'est-à-dire que F est satisfiable ssi il existe $\varphi \in \mathbb{B}^V$ tel que $[\![F]\!]_\varphi = \mathbb{1}$.

Tautologie, antilogie, satisfiabilité

- Une formule est une tautologie si sa valeur de vérité est $\mathbb{1}$ pour toute valuation. c'est-à-dire que F est une tautologie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![F]\!]_\varphi = \mathbb{1}$.
- Une formule est une antilogie si sa valeur de vérité est $\mathbb{0}$ pour toute valuation. c'est-à-dire que F est une antilogie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![F]\!]_\varphi = \mathbb{0}$.
- Une formule est satisfiable s'il existe une valuation pour laquelle sa valeur de vérité est $\mathbb{1}$. c'est-à-dire que F est satisfiable ssi il existe $\varphi \in \mathbb{B}^V$ tel que $[\![F]\!]_\varphi = \mathbb{1}$.

Remarques

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

Tautologie, antilogie, satisfiabilité

- Une formule est une tautologie si sa valeur de vérité est $\mathbb{1}$ pour toute valuation. c'est-à-dire que F est une tautologie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![F]\!]_\varphi = \mathbb{1}$.
- Une formule est une antilogie si sa valeur de vérité est $\mathbb{0}$ pour toute valuation. c'est-à-dire que F est une antilogie ssi pour tout $\varphi \in \mathbb{B}^V$, $[\![F]\!]_\varphi = \mathbb{0}$.
- Une formule est satisfiable s'il existe une valuation pour laquelle sa valeur de vérité est $\mathbb{1}$. c'est-à-dire que F est satisfiable ssi il existe $\varphi \in \mathbb{B}^V$ tel que $[\![F]\!]_\varphi = \mathbb{1}$.

Remarques

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

Tables de vérité

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

Tables de vérité

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

Exemple

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

Tables de vérité

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

Exemple

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

Tables de vérité

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

Exemple

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

p	q	$p \to q$	$\neg p \vee q$	F
$\mathbb{0}$	$\mathbb{0}$			
$\mathbb{0}$	$\mathbb{1}$			
$\mathbb{1}$	$\mathbb{0}$			
$\mathbb{1}$	$\mathbb{1}$			

- Que peut-on en déduire ?

Tables de vérité

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

Exemple

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

p	q	$p \to q$	$\neg p \vee q$	F
$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$		
$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$		
$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$		
$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$		

- Que peut-on en déduire ?

Tables de vérité

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

Exemple

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

p	q	$p \to q$	$\neg p \vee q$	F
$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	
$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	
$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$	$\mathbb{0}$	
$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	

- Que peut-on en déduire ?

Tables de vérité

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

Exemple

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

p	q	$p \to q$	$\neg p \vee q$	F
0	0	1	1	1
0	1	1	1	1
1	0	0	0	1
1	1	1	1	1

- Que peut-on en déduire ?

Tables de vérité

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

Exemple

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

p	q	$p \to q$	$\neg p \vee q$	F
$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$
$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$
$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$
$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$

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

Equivalence logique

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

Equivalence logique

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

Equivalence logique

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

Quelques équivalences à connaitre

- $\neg(\neg F) \equiv F$

Equivalence logique

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

Quelques équivalences à connaitre

- $\neg(\neg F) \equiv F$
- $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$

Equivalence logique

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

Quelques équivalences à connaitre

- $\neg(\neg F) \equiv F$
- $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$
- $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$

Equivalence logique

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

Quelques équivalences à connaitre

- $\neg(\neg F) \equiv F$
- $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$
- $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$
- $\neg(F \vee G) \equiv \neg F \wedge \neg G$ (loi de De Morgan)

Equivalence logique

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

Quelques équivalences à connaitre

- $\neg(\neg F) \equiv F$
- $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$
- $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$
- $\neg(F \vee G) \equiv \neg F \wedge \neg G$ (loi de De Morgan)
- $\neg(F \wedge G) \equiv \neg F \vee \neg G$ (loi de De Morgan)

Equivalence logique

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

Quelques équivalences à connaitre

- $\neg(\neg F) \equiv F$
- $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$
- $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$
- $\neg(F \vee G) \equiv \neg F \wedge \neg G$ (loi de De Morgan)
- $\neg(F \wedge G) \equiv \neg F \vee \neg G$ (loi de De Morgan)
- $F \rightarrow G \equiv \neg F \vee G$

Exemple

Montrer que :

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

Exemple

Montrer que :

- $P \vee \neg P \equiv \top$ (tiers exclu)
- $P \to Q \equiv \neg Q \to \neg P$ (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 G est conséquence logique d'un ensemble de formules $\Gamma = \{F_1, \dots F_n\}$ si pour toute valuation φ, qui rend vraies les formules $(F_i)_{1 \leqslant i \leqslant n}$ rend aussi vraie G. 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 G est conséquence logique d'un ensemble de formules $\Gamma = \{F_1, \dots F_n\}$ si pour toute valuation φ, qui rend vraies les formules $(F_i)_{1 \leqslant i \leqslant n}$ rend aussi vraie G. On notera alors $\Gamma \vDash Q$.
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 G est conséquence logique d'un ensemble de formules $\Gamma = \{F_1, \dots F_n\}$ si pour toute valuation φ, qui rend vraies les formules $(F_i)_{1 \leqslant i \leqslant n}$ rend aussi vraie G. On notera alors $\Gamma \models Q$.
A ne pas confondre avec $\to$ qui est un connecteur logique.

Remarques

- Une formule F est est une tautologie ssi $\varnothing \models F$, on notera simplement, $\models F$.

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 G est conséquence logique d'un ensemble de formules $\Gamma = \{F_1, \dots F_n\}$ si pour toute valuation φ, qui rend vraies les formules $(F_i)_{1 \leqslant i \leqslant n}$ rend aussi vraie G. On notera alors $\Gamma \models Q$.
A ne pas confondre avec $\to$ qui est un connecteur logique.

Remarques

- Une formule F est est une tautologie ssi $\varnothing \models F$, on notera simplement, $\models F$.
- $F \equiv G$ ssi $F \models G$ et $G \models Q$.

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

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 F, il existe une FNC G et une FND H telles que $F \equiv G \equiv H$.

Propositions

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

Propositions

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

Propositions

- Pour tout formule logique F, il existe une FNC G et une FND H telles que $F \equiv G \equiv H$.
- 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 équivalentes n'utilisant pas ces connecteurs : $A \rightarrow B \equiv \neg A \vee B$ et $A \leftrightarrow B \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$.

Propositions

- Pour tout formule logique F, il existe une FNC G et une FND H telles que $F \equiv G \equiv H$.
- 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 équivalentes n'utilisant pas ces connecteurs : $A \rightarrow B \equiv \neg A \vee B$ et $A \leftrightarrow B \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$.
  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 F, il existe une FNC G et une FND H telles que $F \equiv G \equiv H$.
- 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 équivalentes n'utilisant pas ces connecteurs : $A \rightarrow B \equiv \neg A \vee B$ et $A \leftrightarrow B \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$.
  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 F, il existe une FNC G et une FND H telles que $F \equiv G \equiv H$.
- 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 équivalentes n'utilisant pas ces connecteurs : $A \rightarrow B \equiv \neg A \vee B$ et $A \leftrightarrow B \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$.
  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 dans les clauses de littéraux ($v \wedge \neg v \equiv \perp$ et $v \vee \neg v = \top$)

Propositions

- Pour tout formule logique F, il existe une FNC G et une FND H telles que $F \equiv G \equiv H$.
- 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 équivalentes n'utilisant pas ces connecteurs : $A \rightarrow B \equiv \neg A \vee B$ et $A \leftrightarrow B \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$.
  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 dans les clauses de littéraux ($v \wedge \neg v \equiv \perp$ et $v \vee \neg v = \top$)
- Une autre méthode consiste à utiliser la table de vérité.

Exemple

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

- 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
  $P = ((p \to q) \wedge \neg r) \vee (\neg(p \to q) \wedge \neg\neg r)$ (équivalence sémantique de $\leftrightarrow$)
  $P = ((\neg p \vee q) \wedge \neg r) \vee (\neg(\neg p \vee q) \wedge r)$ (équivalence sémantique de $\to$)

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)

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 $\to$)
  $P = ((\neg p \lor q) \land \neg r) \lor ((p \land \neg q) \land r)$ (lois de DeMorgan)
  $P = (\neg p \land \neg r) \lor (q \land \neg r) \lor (p \land \neg q \land r)$ (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 $P = (p \to q) \leftrightarrow \neg r$

- 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	
0	0	0	1	1	1	$\longrightarrow (\neg p \wedge \neg q \wedge \neg r) \vee$
0	0	1	1	0	0	
0	1	0	1	1	1	
0	1	1	1	0	0	
1	0	0	0	1	0	
1	0	1	0	0	1	
1	1	0	1	1	1	
1	1	1	1	0	0	

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	
0	0	0	1	1	1	$\longrightarrow (\neg p \wedge \neg q \wedge \neg r) \vee$
0	0	1	1	0	0	
0	1	0	1	1	1	$\longrightarrow (\neg p \wedge q \wedge \neg r) \vee$
0	1	1	1	0	0	
1	0	0	0	1	0	
1	0	1	0	0	1	
1	1	0	1	1	1	
1	1	1	1	0	0	

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	
$\mathbb{0}$	$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\longrightarrow (\neg p \wedge \neg q \wedge \neg r) \vee$
$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$	
$\mathbb{0}$	$\mathbb{1}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\longrightarrow (\neg p \wedge q \wedge \neg r) \vee$
$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$	
$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{0}$	
$\mathbb{1}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$	$\longrightarrow (p \wedge \neg q \wedge r) \vee$
$\mathbb{1}$	$\mathbb{1}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	
$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$	

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 $\to$)
  $P = ((\neg p \lor q) \land \neg r) \lor ((p \land \neg q) \land r)$ (lois de DeMorgan)
  $P = (\neg p \land \neg r) \lor (q \land \neg r) \lor (p \land \neg q \land r)$ (distributivité et associativité)
- Méthode 2 : utilisation de la table de vérité

p	q	r	$p \to q$	$\neg r$	P	
$\mathbb{0}$	$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\longrightarrow (\neg p \land \neg q \land \neg r) \lor$
$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$	
$\mathbb{0}$	$\mathbb{1}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\longrightarrow (\neg p \land q \land \neg r) \lor$
$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$	
$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{0}$	
$\mathbb{1}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$	$\mathbb{1}$	$\longrightarrow (p \land \neg q \land r) \lor$
$\mathbb{1}$	$\mathbb{1}$	$\mathbb{0}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\longrightarrow (p \land q \land \neg r)$
$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{0}$	$\mathbb{0}$	

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'arrêt 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'arrêt 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 F 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 F une formule contenant les variables logiques $p_1, \dots p_n$.

- On fixe $\varphi(p_1) = \mathbb{0}$ et on teste récursivement la satisfiabilité de F dans laquelle toutes les occurences de p_1 sont remplacées par $\perp$ (notée $F[\perp/p_1]$).
- En cas d'échec, on fixe $\varphi(p_1) = \mathbb{1}$ 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 $F = (p \ \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee r)$

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :

Exemple

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

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :
  $F[\bot/p] = (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r)$

Exemple

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

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :
$$\begin{aligned} F[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \end{aligned}$$

Exemple

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

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :

$$\begin{aligned} F[\bot/p] &= (\bot \lor q) \land (\neg q \lor \neg r) \land (\neg \bot \lor r) \land (\bot \lor r) \\ &= q \land (\neg q \lor \neg r) \land \top \land r \\ &= q \land (\neg q \lor \neg r) \land r \end{aligned}$$

Exemple

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

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :

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

  - On affecte la valeur $\mathbb{0}$ à q et on teste la satisfiabilité de :

Exemple

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

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :
  $$\begin{aligned} F[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur $\mathbb{0}$ à q et on teste la satisfiabilité de : $\bot \wedge (\neg \bot \vee \neg r) \wedge r$ qui est non satisfiable

Exemple

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

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :
  $$\begin{aligned} F[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur $\mathbb{0}$ 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 $\mathbb{1}$ à q et on teste la satisfiabilité de :

Exemple

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

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :
  $$\begin{aligned} F[\bot/p] &= (\bot \lor q) \land (\neg q \lor \neg r) \land (\neg \bot \lor r) \land (\bot \lor r) \\ &= q \land (\neg q \lor \neg r) \land \top \land r \\ &= q \land (\neg q \lor \neg r) \land r \end{aligned}$$
  - On affecte la valeur $\mathbb{0}$ 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 $\mathbb{1}$ 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 $F = (p \lor q) \land (\neg q \lor \neg r) \land (\neg p \lor r) \land (p \lor r)$

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :
  $$\begin{aligned} F[\bot/p] &= (\bot \lor q) \land (\neg q \lor \neg r) \land (\neg\bot \lor r) \land (\bot \lor r) \\ &= q \land (\neg q \lor \neg r) \land \top \land r \\ &= q \land (\neg q \lor \neg r) \land r \end{aligned}$$
  - On affecte la valeur $\mathbb{0}$ 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 $\mathbb{1}$ à q et on teste la satisfiabilité de : $\top \land (\neg\top \lor \neg r) \land r = \neg r \land r$ donc non satisfiable.
- On affecte la valeur $\mathbb{1}$ à p et on teste la satisfiabilité de :

Exemple

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

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :
  $$\begin{aligned} F[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur $\mathbb{0}$ 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 $\mathbb{1}$ 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 $\mathbb{1}$ à p et on teste la satisfiabilité de :
  $F[\top/p] = (\top \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \top \vee r) \wedge (\top \vee r)$

Exemple

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

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :
  $$\begin{aligned} F[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur $\mathbb{0}$ 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 $\mathbb{1}$ 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 $\mathbb{1}$ à p et on teste la satisfiabilité de :
  $F[\top/p] = (\top \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \top \vee r) \wedge (\top \vee r)$
  $F[\top/p] = (\neg q \vee \neg r) \wedge r$

Exemple

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

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :
  $$\begin{aligned} F[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur $\mathbb{0}$ 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 $\mathbb{1}$ 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 $\mathbb{1}$ à p et on teste la satisfiabilité de :
  $F[\top/p] = (\top \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg \top \vee r) \wedge (\top \vee r)$
  $F[\top/p] = (\neg q \vee \neg r) \wedge r$
  - On affecte la valeur $\mathbb{0}$ à q et on teste la satisfiabilité de :

Exemple

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

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :
  $$\begin{aligned} F[\perp/p] &= (\perp \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg\perp \vee r) \wedge (\perp \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur $\mathbb{0}$ à q et on teste la satisfiabilité de : $\perp \wedge (\neg\perp \vee \neg r) \wedge r$ qui est non satisfiable
  - On affecte la valeur $\mathbb{1}$ 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 $\mathbb{1}$ à p et on teste la satisfiabilité de :
  $F[\top/p] = (\top \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg\top \vee r) \wedge (\top \vee r)$
  $F[\top/p] = (\neg q \vee \neg r) \wedge r$
  - On affecte la valeur $\mathbb{0}$ à q et on teste la satisfiabilité de : $(\neg\perp \vee \neg r) \wedge r = r$ qui est satisfiable.

Exemple

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

- On affecte la valeur $\mathbb{0}$ à p et on teste la satisfiabilité de :
  $$\begin{aligned} F[\bot/p] &= (\bot \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg\bot \vee r) \wedge (\bot \vee r) \\ &= q \wedge (\neg q \vee \neg r) \wedge \top \wedge r \\ &= q \wedge (\neg q \vee \neg r) \wedge r \end{aligned}$$
  - On affecte la valeur $\mathbb{0}$ 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 $\mathbb{1}$ 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 $\mathbb{1}$ à p et on teste la satisfiabilité de :
  $F[\top/p] = (\top \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg\top \vee r) \wedge (\top \vee r)$
  $F[\top/p] = (\neg q \vee \neg r) \wedge r$
  - On affecte la valeur $\mathbb{0}$ 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 $\varphi(r) = V$.