

Introduction à la logique

Stéphane Devismes Pascal Lafourcade Michel Lévy

Université Joseph Fourier, Grenoble I

27 Janvier 2010

Organisation

12 semaines :

- ▶ Cours magistral, 1h30 par semaine
- ▶ TD 2 * 1h30 = 3h par semaine

Matériel :

- ▶ Poly (à trous)
- ▶ Sujet du projet

Planning

Dates importantes

- ▶ *Vacances hiver* : du 28 février au 6 mars
- ▶ **Partiels** : 22 mars
- ▶ *Vacances printemps* : du 25 avril au 1 mai
- ▶ **Soutenance Projet** : Mardi 10 mai
- ▶ **Examen** : Semaine du 23 mai
- ▶ **Deuxième session** : Semaine du 20 juin

Note

Evaluations

- ▶ Contrôle continu **40%** : 4 interros **10%**, partiel **40%** et projet **50%**
- ▶ Examen : **60%**

Projet par groupe de 3 ou 4.

- ▶ Phase 1 : Modélisation 3-Sat
- ▶ Phase 2 : DP ou SC, et Walk-Sat ou G-Sat

Plus exemples : Squaro, Sudoku, Master Mind ...

Plan

Préambule

Introduction à la Logique

Logique propositionnelle

Syntaxe

Sens des formules

Equivalences remarquables

Conclusion

Plan

Préambule

Introduction à la Logique

Logique propositionnelle

Syntaxe

Sens des formules

Equivalences remarquables

Conclusion

Logique

Définitions

- ▶ La **logique** précise ce qu'est un raisonnement correct, indépendamment du domaine d'application.

Logique

Définitions

- ▶ La **logique** précise ce qu'est un raisonnement correct, indépendamment du domaine d'application.
- ▶ Un **raisonnement** est un moyen d'obtenir une conclusion à partir d'hypothèses données.

Logique

Définitions

- ▶ La **logique** précise ce qu'est un raisonnement correct, indépendamment du domaine d'application.
- ▶ Un **raisonnement** est un moyen d'obtenir une conclusion à partir d'hypothèses données.
- ▶ Un raisonnement **correct** ne dit rien sur la vérité de hypothèses, il dit seulement que **de la vérité des hypothèses, on peut déduire la vérité de la conclusion.**

Exemples

Exemple I

- ▶ **Hypothèse I** : Tous les hommes sont mortels
- ▶ **Hypothèse II** : Socrate est un homme
- ▶ **Conclusion** : Socrate est mortel

Exemples

Exemple I

- ▶ **Hypothèse I** : Tous les hommes sont mortels
- ▶ **Hypothèse II** : Socrate est un homme
- ▶ **Conclusion** : Socrate est mortel

Exemple II

- ▶ **Hypothèse I** : Tout ce qui est rare est cher
- ▶ **Hypothèse II** : Un cheval bon marché est rare
- ▶ **Conclusion** : Un cheval bon marché est cher !

Ajout d'une hypothèse

Ajout d'une hypothèse

Exemple III

- ▶ **Hypothèse I** : Tout ce qui est rare est cher
- ▶ **Hypothèse II** : Un cheval bon marché est rare
- ▶ **Hypothèse III** : Toute chose bon marché est « pas chère »

Ajout d'une hypothèse

Exemple III

- ▶ **Hypothèse I** : Tout ce qui est rare est cher
- ▶ **Hypothèse II** : Un cheval bon marché est rare
- ▶ **Hypothèse III** : Toute chose bon marché est « pas chère »
- ▶ **Conclusion** : Hypothèses contradictoires ! Car :
 - ▶ **Hypothèse I + hypothèse II** : Un cheval bon marché est cher
 - ▶ **hypothèse III** : Un cheval bon marché est pas cher

Petit historique...

- ▶ **George Boole** (1815-1864), créateur de la logique moderne (notamment l'algèbre de Boole)
- ▶ **Friedrich Ludwig Gottlob Frege** (1848-1925), travaux sur le calcul propositionnel moderne, le calcul des prédicats, théorie de la démonstration
- ▶ **Bertrand Arthur William Russell** (1872-1970), application de la logique à la question du fondement des mathématiques (logicisme)
- ▶ **Alonzo Church** (1903-1995), *lambda-calcul*
- ▶ **Kurt Gödel** (1906-1978), le *théorème d'incomplétude de Gödel*, complétude du calcul des prédicats du premier ordre
- ▶ **Alan Mathison Turing** (1912-1954)

Applications

- ▶ **Hardware** : L'unité arithmétique et logique (UAL) est construite à partir de « portes logiques »
- ▶ **Vérification et correction des programmes** :
 - ▶ **Meteor** (ligne 14)
 - ▶ Outils : prouveurs COQ, PVS, Prover9, MACE, ...
- ▶ **Intelligence artificielle** :
 - ▶ Test de **Turing**
 - ▶ Outil d'aide à la décision : **système expert** (*MyCin*), **ontologie**
 - ▶ **Web Semantique**
- ▶ **Problème SAT** :
 - ▶ Codage de problème de décision sous la forme d'une expression booléenne
 - ▶ Applications en planification, model checking, diagnostic, ...
 - ▶ Solveurs : **zchaff**, **satz**, ...
- ▶ **Programmation** : *Prolog* est utilisé dans de nombreux programmes d'intelligence artificielle et dans le traitement de la linguistique par ordinateur
- ▶ **Preuves mathématiques, Sécurité, ...**

Plan du Semestre

AUJOURD'HUI

- ▶ Logique propositionnelle
- ▶ Résolution propositionnelle
- ▶ Dédution naturelle propositionnelle

PARTIEL

- ▶ Logique du premier ordre
- ▶ Base de la démonstration automatique (“résolution au premier ordre”)
- ▶ Dédution naturelle au premier ordre

EXAMEN

Objectifs du cours

- ▶ **Comprendre un raisonnement**, en particulier, être capable de déterminer si un raisonnement logique est correct ou non.

Objectifs du cours

- ▶ **Comprendre un raisonnement**, en particulier, être capable de déterminer si un raisonnement logique est correct ou non.
- ▶ **Raisonner**, c'est-à-dire, construire un raisonnement correct utilisant les outils de la logique propositionnelle et du premier ordre.

Objectifs du cours

- ▶ **Comprendre un raisonnement**, en particulier, être capable de déterminer si un raisonnement logique est correct ou non.
- ▶ **Raisonner**, c'est-à-dire, construire un raisonnement correct utilisant les outils de la logique propositionnelle et du premier ordre.
- ▶ **Modéliser et formaliser un problème.**

Objectifs du cours

- ▶ **Comprendre un raisonnement**, en particulier, être capable de déterminer si un raisonnement logique est correct ou non.
- ▶ **Raisonner**, c'est-à-dire, construire un raisonnement correct utilisant les outils de la logique propositionnelle et du premier ordre.
- ▶ **Modéliser et formaliser un problème.**
- ▶ **Ecrire une preuve rigoureuse.**

Plan

Préambule

Introduction à la Logique

Logique propositionnelle

Syntaxe

Sens des formules

Equivalences remarquables

Conclusion

Logique propositionnelle

Définition

La **logique propositionnelle** est la logique qui s'intéresse aux seules lois gouvernant les opérations logiques suivantes :

- ▶ \neg (négation),
- ▶ \wedge (conjonction, autrement dit le “et”),
- ▶ \vee (disjonction, autrement dit le “ou”),
- ▶ \Rightarrow (implication) et
- ▶ \Leftrightarrow (équivalence).

Remarque

Nous restreindrons notre étude à la logique **classique**, qui est la logique à deux valeurs de vérité : VRAI et FAUX

Exemple : **Raisonnement formel**

Hypothèses :

- ▶ (H1) : Si Pierre est grand, alors Jean n'est pas le fils de Pierre
- ▶ (H2) : Si Pierre n'est pas grand, alors Jean est le fils de Pierre
- ▶ (H3) : Si Jean est le fils de Pierre alors Marie est la soeur de Jean

Conclusion (C) : Marie est la soeur de Jean ou Pierre est grand.

Exemple : Raisonnement formel

Hypothèses :

- ▶ (H1) : Si Pierre est grand, alors Jean n'est pas le fils de Pierre
- ▶ (H2) : Si Pierre n'est pas grand, alors Jean est le fils de Pierre
- ▶ (H3) : Si Jean est le fils de Pierre alors Marie est la soeur de Jean

Conclusion (C) : Marie est la soeur de Jean ou Pierre est grand.

- | | |
|--------------------------------------|---------------------------------|
| ▶ p : "Pierre est grand" | ▶ (H1) : $p \Rightarrow \neg j$ |
| ▶ j : "Jean est le fils de Pierre" | ▶ (H2) : $\neg p \Rightarrow j$ |
| ▶ m : "Marie est la soeur de Jean" | ▶ (H3) : $j \Rightarrow m$ |

$$(C) : m \vee p$$

Nous montrons que $H1 \wedge H2 \wedge H3 \Rightarrow C$:

$$(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m) \Rightarrow m \vee p$$

est vraie quelque soit la vérité des propositions p, j, m .

Plan

Préambule

Introduction à la Logique

Logique propositionnelle

Syntaxe

Sens des formules

Equivalences remarquables

Conclusion

Vocabulaire du langage

- ▶ Constantes : \top et \perp représentant respectivement le vrai et le faux.
- ▶ Variables : une variable est un identificateur, avec ou sans indice.
- ▶ Parenthèses : ouvrante (, fermante).
- ▶ Connecteurs : $\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$ appelées aussi négation, disjonction (ou), conjonction (et), implication et équivalence.

Formule (stricte)

Définition 1.1.1

Une **formule stricte** est définie de manière inductive par :

- ▶ \top et \perp sont des formules.
- ▶ Une variable est une formule.
- ▶ Si A est une formule alors $\neg A$ est une formule
- ▶ Si A et B sont des formules et si \circ est une des opérations $\vee, \wedge, \Rightarrow, \Leftrightarrow$ alors $(A \circ B)$ est une formule.

Formule (stricte)

Définition 1.1.1

Une **formule stricte** est définie de manière inductive par :

- ▶ \top et \perp sont des formules.
- ▶ Une variable est une formule.
- ▶ Si A est une formule alors $\neg A$ est une formule
- ▶ Si A et B sont des formules et si \circ est une des opérations $\vee, \wedge, \Rightarrow, \Leftrightarrow$ alors $(A \circ B)$ est une formule.

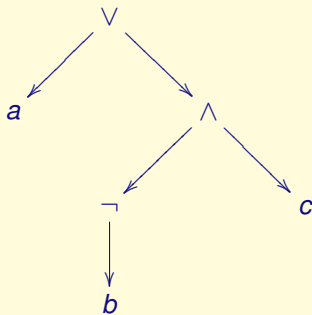
Exemple 1.1.2

$(a \vee (\neg b \wedge c))$ est une formule stricte, mais pas $a \vee (\neg b \wedge c)$.

Arbre

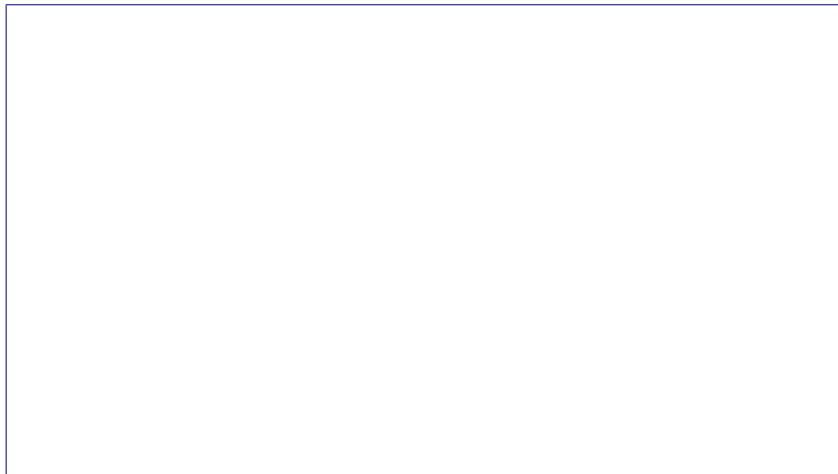
Exemple 1.1.3

La structure de la formule $(a \vee (\neg b \wedge c))$ est mise en évidence par l'arbre suivant :



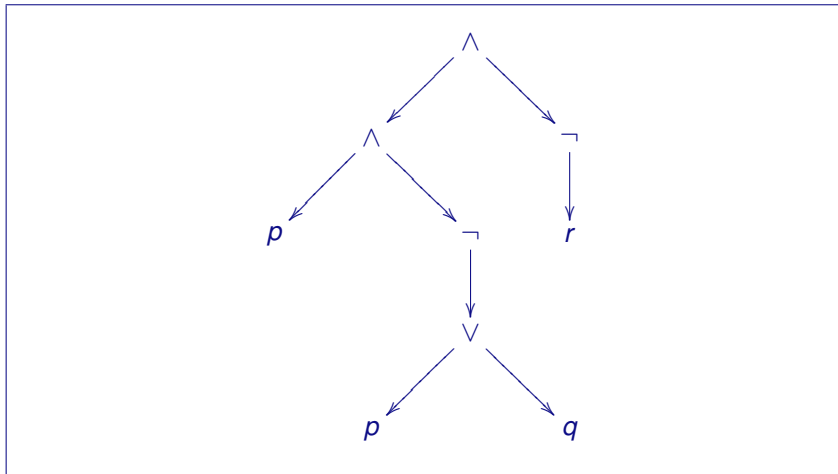
Exercice

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



Exercice

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



Premier résultat

Les formules strictes **se décomposent de manière unique** en leurs sous-formules

(théorème 1.1.11)

Longueur

Définition 1.1.4

La **longueur** d'une formule A est le nombre de symboles utilisés pour écrire une formule, dénotée $l(A)$.

Longueur

Définition 1.1.4

La **longueur** d'une formule A est le nombre de symboles utilisés pour écrire une formule, dénotée $l(A)$.

Exemple 1.1.5

Soit la formule $A = (a \vee b)$ et $B = (A \wedge \neg A)$,

Longueur

Définition 1.1.4

La **longueur** d'une formule A est le nombre de symboles utilisés pour écrire une formule, dénotée $l(A)$.

Exemple 1.1.5

Soit la formule $A = (a \vee b)$ et $B = (A \wedge \neg A)$,

nous avons $l(A) = 5$ et $l(B) = 14$.

Résultats intermédiaires (1/2)

Lemme 1.1.6

Toute formule a un nombre égal de parenthèses ouvrantes et de parenthèses fermantes.

Démonstration.

Toute parenthèse ouvrante est associée à une parenthèse fermante. □

Résultats intermédiaires (1/2)

Lemme 1.1.6

Toute formule a un nombre égal de parenthèses ouvrantes et de parenthèses fermantes.

Démonstration.

Toute parenthèse ouvrante est associée à une parenthèse fermante.

Lemme 1.1.7

Tout préfixe d'une formule a un nombre de parenthèses ouvrantes au moins égal à celui des parenthèses fermantes.

Démonstration.

Preuve par récurrence sur la longueur des formules (*cf.* Poly).

Taille d'une formule

Définition 1.1.8

La **taille d'une formule** A , notée $|A|$, est définie inductivement par :

- ▶ $|\top| = 0$ et $|\perp| = 0$.
- ▶ Si A est une variable $|A| = 0$.
- ▶ $|\neg A| = 1 + |A|$.
- ▶ $|(A \circ B)| = |A| + |B| + 1$.

Taille d'une formule

Définition 1.1.8

La **taille d'une formule** A , notée $|A|$, est définie inductivement par :

- ▶ $|\top| = 0$ et $|\perp| = 0$.
- ▶ Si A est une variable $|A| = 0$.
- ▶ $|\neg A| = 1 + |A|$.
- ▶ $|(A \circ B)| = |A| + |B| + 1$.

Exemple 1.1.9

$$|(a \vee (\neg b \wedge c))| =$$

Taille d'une formule

Définition 1.1.8

La **taille d'une formule** A , notée $|A|$, est définie inductivement par :

- ▶ $|\top| = 0$ et $|\perp| = 0$.
- ▶ Si A est une variable $|A| = 0$.
- ▶ $|\neg A| = 1 + |A|$.
- ▶ $|(A \circ B)| = |A| + |B| + 1$.

Exemple 1.1.9

$|(a \vee (\neg b \wedge c))| =$

3.

Résultats intermédiaires (2/2) + exemple récurrence

Lemme 1.1.10

Tout préfixe strict d'une formule n'est pas une formule.

Démonstration.

Induction sur la taille de la formule. □

Cas de base

Soit A une formule de taille 0, A est donc soit une variable soit une constante. Le seul préfixe strict de A est le mot vide qui n'est pas une formule.

Hypothèse d'induction

Supposons le lemme vrai pour toute formule de longueur inférieure à n avec $n > 0$.

Pas d'induction

Soit A une formule de longueur n .

Pas d'induction

Soit A une formule de longueur n .

- ▶ Soit $A = \neg B$, où B est une formule.

- ▶ Soit $A = (B \circ C)$ où B et C sont des formules.

Pas d'induction

Soit A une formule de longueur n .

- Soit $A = \neg B$, où B est une formule.

Supposons que A ait un préfixe strict qui soit une formule. Ce préfixe doit s'écrire $\neg B'$ où B' est une formule et un préfixe strict de B . C'est impossible d'après l'hypothèse de récurrence.

- Soit $A = (B \circ C)$ où B et C sont des formules.

Supposons que A ait un préfixe strict qui soit une formule. Ce préfixe doit s'écrire $(B' \circ C')$ où B' et C' sont des formules. La formule B est un préfixe de la formule B' ou la formule B' est un préfixe de la formule B . Il est impossible, d'après l'hypothèse de récurrence, que ces préfixes soient stricts. Donc $B = B'$. D'où, C' est un préfixe strict de C . D'après le lemme 1.1.6, C' a autant de parenthèses ouvrantes que de fermantes. Donc C' est un préfixe de la formule C , qui a plus de parenthèses fermantes que de parenthèses ouvrantes, ce qui contredit le lemme 1.1.7. Donc A n'a pas de préfixe strict qui est une formule.

Pas d'induction

Soit A une formule de longueur n .

- ▶ Soit $A = \neg B$, où B est une formule.

Supposons que A ait un préfixe strict qui soit une formule. Ce préfixe doit s'écrire $\neg B'$ où B' est une formule et un préfixe strict de B . C'est impossible d'après l'hypothèse de récurrence.

- ▶ Soit $A = (B \circ C)$ où B et C sont des formules.

Supposons que A ait un préfixe strict qui soit une formule. Ce préfixe doit s'écrire $(B' \circ C')$ où B' et C' sont des formules. La formule B est un préfixe de la formule B' ou la formule B' est un préfixe de la formule B . Il est impossible, d'après l'hypothèse de récurrence, que ces préfixes soient stricts. Donc $B = B'$. D'où, C' est un préfixe strict de C . D'après le lemme 1.1.6, C' a autant de parenthèses ouvrantes que de fermantes. Donc C' est un préfixe de la formule C , qui a plus de parenthèses fermantes que de parenthèses ouvrantes, ce qui contredit le lemme 1.1.7. Donc A n'a pas de préfixe strict qui est une formule.

Ainsi par récurrence, tout préfixe strict d'une formule n'est pas une formule.

Premier résultat

Theoreme 1.1.11

Pour toute formule A , un et un seul de ces cas se présente :

- ▶ A est une variable,
- ▶ A est une constante,
- ▶ A est s'écrit d'une unique façon sous la forme $\neg B$ où B est une formule,
- ▶ A s'écrit d'une unique façon sous la forme $(B \circ C)$ où B et C sont des formules.

Démonstration.

Puisque les vocabulaires constantes, variables, négation, parenthèses sont disjoints, toute formule A est d'une et d'une seule des formes variable, constante, $\neg B$ où B est une formule, $(B \circ C)$ où B et C sont des formules.

Premier résultat

Theoreme 1.1.11

Pour toute formule A , un et un seul de ces cas se présente :

- ▶ A est une variable,
- ▶ A est une constante,
- ▶ A est s'écrit d'une unique façon sous la forme $\neg B$ où B est une formule,
- ▶ A s'écrit d'une unique façon sous la forme $(B \circ C)$ où B et C sont des formules.

Démonstration.

Puisque les vocabulaires constantes, variables, négation, parenthèses sont disjoints, toute formule A est d'une et d'une seule des formes variable, constante, $\neg B$ où B est une formule, $(B \circ C)$ où B et C sont des formules.

Il suffit de montrer l'unicité de la dernière décomposition.

Premier résultat

Theoreme 1.1.11

Pour toute formule A , un et un seul de ces cas se présente :

- ▶ A est une variable,
- ▶ A est une constante,
- ▶ A est s'écrit d'une unique façon sous la forme $\neg B$ où B est une formule,
- ▶ A s'écrit d'une unique façon sous la forme $(B \circ C)$ où B et C sont des formules.

Démonstration.

Puisque les vocabulaires constantes, variables, négation, parenthèses sont disjoints, toute formule A est d'une et d'une seule des formes variable, constante, $\neg B$ où B est une formule, $(B \circ C)$ où B et C sont des formules.

Il suffit de montrer l'unicité de la dernière décomposition.

Supposons que $A = (B \circ C) = (B' \circ C')$ où B, B', C, C' sont des formules.

Premier résultat

Theoreme 1.1.11

Pour toute formule A , un et un seul de ces cas se présente :

- ▶ A est une variable,
- ▶ A est une constante,
- ▶ A est s'écrit d'une unique façon sous la forme $\neg B$ où B est une formule,
- ▶ A s'écrit d'une unique façon sous la forme $(B \circ C)$ où B et C sont des formules.

Démonstration.

Puisque les vocabulaires constantes, variables, négation, parenthèses sont disjoints, toute formule A est d'une et d'une seule des formes variable, constante, $\neg B$ où B est une formule, $(B \circ C)$ où B et C sont des formules.

Il suffit de montrer l'unicité de la dernière décomposition.

Supposons que $A = (B \circ C) = (B' \circ C')$ où B, B', C, C' sont des formules.

La formule B est préfixe de B' ou la formule B' est préfixe de B .

Premier résultat

Theoreme 1.1.11

Pour toute formule A , un et un seul de ces cas se présente :

- ▶ A est une variable,
- ▶ A est une constante,
- ▶ A est s'écrit d'une unique façon sous la forme $\neg B$ où B est une formule,
- ▶ A s'écrit d'une unique façon sous la forme $(B \circ C)$ où B et C sont des formules.

Démonstration.

Puisque les vocabulaires constantes, variables, négation, parenthèses sont disjoints, toute formule A est d'une et d'une seule des formes variable, constante, $\neg B$ où B est une formule, $(B \circ C)$ où B et C sont des formules.

Il suffit de montrer l'unicité de la dernière décomposition.

Supposons que $A = (B \circ C) = (B' \circ C')$ où B, B', C, C' sont des formules.

La formule B est préfixe de B' ou la formule B' est préfixe de B .

D'après le lemme 1.1.10, il est impossible que ce soit des préfixes stricts, donc $B = B'$ et par suite $C = C'$. □

Formule à priorité

Définition 1.1.12

Une **formule à priorité** est définie inductivement par :

- ▶ \top et \perp sont des formules à priorité,
- ▶ une variable est une formule à priorité,
- ▶ si A est une formule à priorité alors $\neg A$ est une formule à priorité,
- ▶ si A et B sont des formules à priorité alors $A \circ B$ est une formule à priorité,
- ▶ si A est une formule à priorité alors (A) est une formule à priorité.

Exemple

$a \vee \neg b \wedge c$ est une formule à priorité mais pas une formule.

Règles de priorité

Définition 1.1.14

dans l'ordre des priorités décroissantes : \neg , \wedge , \vee , \Rightarrow et \Leftrightarrow .

À priorité égale, le connecteur gauche est prioritaire, **sauf pour l'implication.**

Règles de priorité

Définition 1.1.14

dans l'ordre des priorités décroissantes : \neg , \wedge , \vee , \Rightarrow et \Leftrightarrow .

À priorité égale, le connecteur gauche est prioritaire, **sauf pour l'implication.**

Une formule à priorité est **l'abréviation** de la formule (stricte) associé.

Exemple de formule à priorité

Exemple 1.1.15

- ▶ $a \wedge b \wedge c$ est l'abréviation de
- ▶ $a \wedge b \vee c$ est l'abréviation de
- ▶ $a \vee b \wedge c$ est l'abréviation de
- ▶ $a \Rightarrow b \Rightarrow c$ est l'abréviation de

Exemple de formule à priorité

Exemple 1.1.15

- ▶ $a \wedge b \wedge c$ est l'abréviation de

$$((a \wedge b) \wedge c)$$

- ▶ $a \wedge b \vee c$ est l'abréviation de

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

- ▶ $a \vee b \wedge c$ est l'abréviation de

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

- ▶ $a \Rightarrow b \Rightarrow c$ est l'abréviation de

$$(a \Rightarrow (b \Rightarrow c))$$

Plan

Préambule

Introduction à la Logique

Logique propositionnelle

Syntaxe

Sens des formules

Equivalences remarquables

Conclusion

Tables de base

0 désigne le faux et 1 le vrai.

La constante \top vaut 1 et la constante \perp vaut 0

Table 1.1 (table de vérité des connecteurs)

x	y	$\neg x$	$x \vee y$	$x \wedge y$	$x \Rightarrow y$	$x \Leftrightarrow y$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

Valeur d'une formule

Définition 1.2.1

Une **assignation** est une application de l'ensemble de toutes les variables d'une formule dans l'ensemble $\{0, 1\}$. Soit A une formule et v une assignation, $[A]_v$ dénote la valeur de la formule A dans l'assignation v .

Valeur d'une formule

Définition 1.2.1

Une **assignation** est une application de l'ensemble de toutes les variables d'une formule dans l'ensemble $\{0, 1\}$. Soit A une formule et v une assignation, $[A]_v$ dénote la valeur de la formule A dans l'assignation v .

Exemple : Soit v une assignation telle que $v(x) = 0$ et $v(y) = 1$

Valeur d'une formule

Définition 1.2.1

Une **assignation** est une application de l'ensemble de toutes les variables d'une formule dans l'ensemble $\{0, 1\}$. Soit A une formule et v une assignation, $[A]_v$ dénote la valeur de la formule A dans l'assignation v .

Exemple : Soit v une assignation telle que $v(x) = 0$ et $v(y) = 1$
Appliquer l'assignation v à $x \vee y$ s'écrit $[x \vee y]_v$

Valeur d'une formule

Définition 1.2.1

Une **assignation** est une application de l'ensemble de toutes les variables d'une formule dans l'ensemble $\{0, 1\}$. Soit A une formule et v une assignation, $[A]_v$ dénote la valeur de la formule A dans l'assignation v .

Exemple : Soit v une assignation telle que $v(x) = 0$ et $v(y) = 1$
Appliquer l'assignation v à $x \vee y$ s'écrit $[x \vee y]_v$
Cela vaut $0 \vee 1 = 1$

Valeur d'une formule

Définition 1.2.1

Une **assignation** est une application de l'ensemble de toutes les variables d'une formule dans l'ensemble $\{0, 1\}$. Soit A une formule et v une assignation, $[A]_v$ dénote la valeur de la formule A dans l'assignation v .

Exemple : Soit v une assignation telle que $v(x) = 0$ et $v(y) = 1$

Appliquer l'assignation v à $x \vee y$ s'écrit $[x \vee y]_v$

Cela vaut $0 \vee 1 = 1$

Conclusion : $x \vee y$ est vrai pour l'assignation v

Valeur de $[A]_v$

Soient A, B des formules, x une variable et v une assignation.

- ▶ $[x]_v =$
- ▶ $[\top]_v =$, $[\perp]_v =$
- ▶ $[\neg A]_v =$
- ▶ $[(A \vee B)]_v =$
- ▶ $[(A \wedge B)]_v =$
- ▶ $[(A \Rightarrow B)]_v =$
- ▶ $[(A \Leftrightarrow B)]_v =$

Valeur de $[A]_v$

Soient A, B des formules, x une variable et v une assignation.

- ▶ $[x]_v = v(x)$
- ▶ $[\top]_v = \text{vrai}$, $[\perp]_v = \text{faux}$
- ▶ $[\neg A]_v = \text{non } [A]_v$
- ▶ $[(A \vee B)]_v = [A]_v \vee [B]_v$
- ▶ $[(A \wedge B)]_v = [A]_v \wedge [B]_v$
- ▶ $[(A \Rightarrow B)]_v = [A]_v \Rightarrow [B]_v$
- ▶ $[(A \Leftrightarrow B)]_v = [A]_v \Leftrightarrow [B]_v$

Valeur de $[A]_v$

Soient A, B des formules, x une variable et v une assignation.

- ▶ $[x]_v = v(x)$
- ▶ $[\top]_v = 1, [\perp]_v =$
- ▶ $[\neg A]_v =$
- ▶ $[(A \vee B)]_v =$
- ▶ $[(A \wedge B)]_v =$
- ▶ $[(A \Rightarrow B)]_v =$
- ▶ $[(A \Leftrightarrow B)]_v =$

Valeur de $[A]_v$

Soient A, B des formules, x une variable et v une assignation.

- ▶ $[x]_v = v(x)$
- ▶ $[\top]_v = 1, [\perp]_v = 0$
- ▶ $[\neg A]_v =$
- ▶ $[(A \vee B)]_v =$
- ▶ $[(A \wedge B)]_v =$
- ▶ $[(A \Rightarrow B)]_v =$
- ▶ $[(A \Leftrightarrow B)]_v =$

Valeur de $[A]_v$

Soient A, B des formules, x une variable et v une assignation.

- ▶ $[x]_v = v(x)$
- ▶ $[\top]_v = 1, [\perp]_v = 0$
- ▶ $[\neg A]_v = 1 - [A]_v$
- ▶ $[(A \vee B)]_v =$
- ▶ $[(A \wedge B)]_v =$
- ▶ $[(A \Rightarrow B)]_v =$
- ▶ $[(A \Leftrightarrow B)]_v =$

Valeur de $[A]_v$

Soient A, B des formules, x une variable et v une assignation.

- ▶ $[x]_v = v(x)$
- ▶ $[\top]_v = 1, [\perp]_v = 0$
- ▶ $[\neg A]_v = 1 - [A]_v$
- ▶ $[(A \vee B)]_v = \max\{[A]_v, [B]_v\}$
- ▶ $[(A \wedge B)]_v =$
- ▶ $[(A \Rightarrow B)]_v =$
- ▶ $[(A \Leftrightarrow B)]_v =$

Valeur de $[A]_v$

Soient A, B des formules, x une variable et v une assignation.

- ▶ $[x]_v = v(x)$
- ▶ $[\top]_v = 1, [\perp]_v = 0$
- ▶ $[\neg A]_v = 1 - [A]_v$
- ▶ $[(A \vee B)]_v = \max\{[A]_v, [B]_v\}$
- ▶ $[(A \wedge B)]_v = \min\{[A]_v, [B]_v\}$
- ▶ $[(A \Rightarrow B)]_v =$
- ▶ $[(A \Leftrightarrow B)]_v =$

Valeur de $[A]_v$

Soient A, B des formules, x une variable et v une assignation.

- ▶ $[x]_v = v(x)$
- ▶ $[\top]_v = 1, [\perp]_v = 0$
- ▶ $[\neg A]_v = 1 - [A]_v$
- ▶ $[(A \vee B)]_v = \max\{[A]_v, [B]_v\}$
- ▶ $[(A \wedge B)]_v = \min\{[A]_v, [B]_v\}$
- ▶ $[(A \Rightarrow B)]_v = \text{si } [A]_v = 0 \text{ alors } 1 \text{ sinon } [B]_v$
- ▶ $[(A \Leftrightarrow B)]_v =$

Valeur de $[A]_v$

Soient A, B des formules, x une variable et v une assignation.

- ▶ $[x]_v = v(x)$
- ▶ $[\top]_v = 1, [\perp]_v = 0$
- ▶ $[\neg A]_v = 1 - [A]_v$
- ▶ $[(A \vee B)]_v = \max\{[A]_v, [B]_v\}$
- ▶ $[(A \wedge B)]_v = \min\{[A]_v, [B]_v\}$
- ▶ $[(A \Rightarrow B)]_v =$ si $[A]_v = 0$ alors 1 sinon $[B]_v$
- ▶ $[(A \Leftrightarrow B)]_v =$ si $[A]_v = [B]_v$ alors 1 sinon 0

Table de vérité

Définition 1.2.2

Une **table de vérité** d'une formule A est un tableau qui représente la valeur de A pour toutes les valeurs possibles des variables de A .

- ▶ ligne de la table de vérité = une assignation
- ▶ colonne = valeur d'une formule.

Exemple :

Exemple 1.2.3

Donner la table de vérité des formules suivantes.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0					
0	1					
1	0					
1	1					

Exemple :

Exemple 1.2.3

Donner la table de vérité des formules suivantes.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1				
0	1	1				
1	0	0				
1	1	1				

Exemple :

Exemple 1.2.3

Donner la table de vérité des formules suivantes.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1	1			
0	1	1	1			
1	0	0	0			
1	1	1	0			

Exemple :

Exemple 1.2.3

Donner la table de vérité des formules suivantes.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1	1	1		
0	1	1	1	1		
1	0	0	0	0		
1	1	1	0	1		

Exemple :

Exemple 1.2.3

Donner la table de vérité des formules suivantes.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1	1	1	1	
0	1	1	1	1	1	
1	0	0	0	0	1	
1	1	1	0	1	1	

Exemple :

Exemple 1.2.3

Donner la table de vérité des formules suivantes.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1	1	1	1	1
0	1	1	1	1	1	0
1	0	0	0	0	1	1
1	1	1	0	1	1	1

Autre exemple :

Donnez la table de vérité de

a	b	c	$\neg b$	$(\neg b \wedge c)$	$(a \vee (\neg b \wedge c))$
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Autre exemple :

Donnez la table de vérité de

a	b	c	$\neg b$	$(\neg b \wedge c)$	$(a \vee (\neg b \wedge c))$
0	0	0	1		
0	0	1	1		
0	1	0	0		
0	1	1	0		
1	0	0	1		
1	0	1	1		
1	1	0	0		
1	1	1	0		

Autre exemple :

Donnez la table de vérité de

a	b	c	$\neg b$	$(\neg b \wedge c)$	$(a \vee (\neg b \wedge c))$
0	0	0	1	0	
0	0	1	1	1	
0	1	0	0	0	
0	1	1	0	0	
1	0	0	1	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	1	0	0	

Autre exemple :

Donnez la table de vérité de

a	b	c	$\neg b$	$(\neg b \wedge c)$	$(a \vee (\neg b \wedge c))$
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

Equivalence

Définition 1.2.4

Deux formules A et B sont **équivalentes** (noté $A \equiv B$ ou simplement $A = B$) si, pour toute assignation, elles ont la même valeur.

Equivalence

Définition 1.2.4

Deux formules A et B sont **équivalentes** (noté $A \equiv B$ ou simplement $A = B$) si, pour toute assignation, elles ont la même valeur.

Exemple 1.2.5

$$x \Rightarrow y = \neg x \vee y$$

Equivalence

Définition 1.2.4

Deux formules A et B sont **équivalentes** (noté $A \equiv B$ ou simplement $A = B$) si, pour toute assignation, elles ont la même valeur.

Exemple 1.2.5

$$x \Rightarrow y = \neg x \vee y$$

Remarque :

Le connecteur logique \Leftrightarrow ne signifie pas $A \equiv B$.

Valide, tautologie (1/2)

Définition 1.2.6

- ▶ Une formule est **valide** si elle a la valeur 1 pour toute assignation.
- ▶ Une formule valide est aussi appelée une **tautologie**.
- ▶ Nous notons le fait que A soit valide par $\models A$.

Valide, tautologie (1/2)

Définition 1.2.6

- ▶ Une formule est **valide** si elle a la valeur 1 pour toute assignation.
- ▶ Une formule valide est aussi appelée une **tautologie**.
- ▶ Nous notons le fait que A soit valide par $\models A$.

Exemple 1.2.7

- ▶ $x \Rightarrow y$ n'est pas valide car $x = 1$ et $y = 0$.
- ▶ La formule $(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$ est valide.
- ▶ Nous pouvons écrire $\models x \vee \neg x$, car $x \vee \neg x$ est une tautologie.

Valide, tautologie (2/2)

Propriété 1.2.8

Les formules A et B sont équivalentes si et seulement si la formule $A \Leftrightarrow B$ est valide.

Démonstration.

La propriété est une conséquence immédiate de la table 1.1.

Modèle d'une formule

Définition 1.2.9

Un **modèle** de A est une assignation v telle que $[A]_v = 1$

v **satisfait** A ou v rend A **vraie**.

Modèle d'une formule

Définition 1.2.9

Un **modèle** de A est une assignation v telle que $[A]_v = 1$

v **satisfait** A ou v rend A **vraie**.

Exemple 1.2.10

Un modèle de $x \Rightarrow y$ est :

Modèle d'une formule

Définition 1.2.9

Un **modèle** de A est une assignation v telle que $[A]_v = 1$

v **satisfait** A ou v rend A **vraie**.

Exemple 1.2.10

Un modèle de $x \Rightarrow y$ est :

$x = 1, y = 1$ ou l'assignation $x = 0$ et y quelconque.

Modèle d'une formule

Définition 1.2.9

Un **modèle** de A est une assignation v telle que $[A]_v = 1$

v **satisfait** A ou v rend A **vraie**.

Exemple 1.2.10

Un modèle de $x \Rightarrow y$ est :

$x = 1, y = 1$ ou l'assignation $x = 0$ et y quelconque.

Par contre l'assignation $x = 1$ et $y = 0$ n'est pas un modèle de $x \Rightarrow y$.

Modèle d'un ensemble de formules

Définition 1.2.11

Une assignation est **un modèle d'un ensemble de formules** si et seulement si elle est un modèle de chaque formule de l'ensemble.

Modèle d'un ensemble de formules

Définition 1.2.11

Une assignation est **un modèle d'un ensemble de formules** si et seulement si elle est un modèle de chaque formule de l'ensemble.

Exemple 1.2.12

Un modèle de $\{a \Rightarrow b, b \Rightarrow c\}$ est :

Modèle d'un ensemble de formules

Définition 1.2.11

Une assignation est **un modèle d'un ensemble de formules** si et seulement si elle est un modèle de chaque formule de l'ensemble.

Exemple 1.2.12

Un modèle de $\{a \Rightarrow b, b \Rightarrow c\}$ est :

$a = 0, b = 0$ (et c quelconque).

Satisfiable

Définition 1.2.13

Une formule (respectivement un ensemble de formules) est **satisfaisable** s'il y a une assignation qui en est un modèle.

Définition 1.2.14

Une formule (respectivement un ensemble de formules) est **contradictoire** (aussi dite **insatisfiable**) si elle n'est pas satisfiable.

Propriété de la satisfiabilité

Propriété 1.2.16

Une assignation est un modèle d'un ensemble de formules si et seulement si elle est un modèle de la conjonction des formules de l'ensemble.

La preuve est demandée dans l'exercice 12.

Propriété de la satisfiabilité

Propriété 1.2.16

Une assignation est un modèle d'un ensemble de formules si et seulement si elle est un modèle de la conjonction des formules de l'ensemble.

La preuve est demandée dans l'exercice 12.

Exemple 1.2.17

L'ensemble de formules $\{a \Rightarrow b, b \Rightarrow c\}$ et la formule $(a \Rightarrow b) \wedge (b \Rightarrow c)$ ont les mêmes modèles.

Contre-modèle

Définition 1.2.18

Un **contre-modèle** de A est une assignation v telle que $[A]_v = 0$.

v ne satisfait pas A ou v rend A **fausse**.

Contre-modèle

Définition 1.2.18

Un **contre-modèle** de A est une assignation v telle que $[A]_v = 0$.

v ne satisfait pas A ou v rend A **fausse**.

Exemple 1.2.19

Un contre-modèle de $x \Rightarrow y$ est :

Contre-modèle

Définition 1.2.18

Un **contre-modèle** de A est une assignation v telle que $[A]_v = 0$.

v ne satisfait pas A ou v rend A **fausse**.

Exemple 1.2.19

Un contre-modèle de $x \Rightarrow y$ est :

l'assignation $x = 1, y = 0$.

Contre-modèle

Définition 1.2.18

Un **contre-modèle** de A est une assignation v telle que $[A]_v = 0$.

v ne satisfait pas A ou v rend A **fausse**.

Exemple 1.2.19

Un contre-modèle de $x \Rightarrow y$ est :

l'assignation $x = 1, y = 0$.

Remarque 1.2.20

Les logiciens utilisent le mot **consistant** comme synonyme de satisfaisable et **contradictoire** comme synonyme d'insatisfaisable.

Conséquence

Définition 1.2.21

A est **conséquence** de l'ensemble Γ d'hypothèses ($\Gamma \models A$) si tout modèle de Γ est modèle de A .

Remarque 1.2.23

Nous notons A est valide par $\models A$, car A est valide si et seulement si A est conséquence de l'ensemble vide.

Exemple de Conséquence

Exemple 1.2.28

$$a \Rightarrow b, b \Rightarrow c \models a \Rightarrow c.$$

Exemple de Conséquence

Exemple 1.2.28

$$a \Rightarrow b, b \Rightarrow c \models a \Rightarrow c.$$

a	b	c	$a \Rightarrow b$	$b \Rightarrow c$	$a \Rightarrow c$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	1

Propriété INCONTOURNABLE

Constamment utilisé dans les exercices, EXAMENS.

Propriété 1.2.27

Soient $n + 1$ formules A_1, \dots, A_n, B et soit H_n la *conjonction* des formules A_1, \dots, A_n . Les trois formulations suivantes sont équivalentes :

1. B est conséquence des hypothèses A_1, \dots, A_n (c'est-à-dire $A_1, \dots, A_n \models B$).
2. La formule $H_n \Rightarrow B$ est valide.
3. $H_n \wedge \neg B$ est insatisfaisable.

Démonstration.

La propriété est une conséquence de la table 1.1



Preuve (1/3)

- ▶ $1 \Rightarrow 2$: supposons que $A_1, \dots, A_n \models B$: tout modèle de A_1, \dots, A_n est aussi modèle de B .
 - ▶ Soit une assignation v non modèle de $A_1, \dots, A_n : \exists i, [A_i]_v = 0$, donc $[H_n]_v = 0$. Ainsi $[H_n \Rightarrow B]_v = 1$.
 - ▶ Soit une assignation v modèle de A_1, \dots, A_n : $[A_i]_v = 1$ pour $i = 1, \dots, n$, donc $[H_n]_v = 1$. Or v est un modèle de B donc $[B]_v = 1$. Ainsi $[H_n \Rightarrow B]_v = 1$.

Donc $H_n \Rightarrow B$ est valide.

Preuve (2/3)

► $2 \Rightarrow 3$: supposons que $H_n \Rightarrow B$ est valide : $\forall v$ assignation, $[H_n \Rightarrow B]_v = 1$.

► soit $[H_n]_v = 0$,

► soit $[H_n]_v = 1$ et $[B]_v = 1$.

Or $[H_n \wedge \neg B]_v = \max([H_n]_v, [\neg B]_v) = \max([H_n]_v, 1 - [B]_v)$. Dans les deux cas, nous obtenons $[H_n \wedge \neg B]_v = 0$. Donc $H_n \wedge \neg B$ est insatisfaisable.

Preuve (3/3)

- ▶ $3 \Rightarrow 1$: supposons que $H_n \wedge \neg B$ est insatisfaisable : \forall assignation la formule $H_n \wedge \neg B$ est contradictoire. Montrons que les modèles de A_1, \dots, A_n sont aussi des modèles de B .
Soit v une assignation modèle de A_1, \dots, A_n : $[A_i]_v = 1$ pour $i = 1, \dots, n$ donc $[H_n]_v = [A_1 \wedge \dots \wedge A_n]_v = 1$.
D'après notre hypothèse $[\neg B]_v = 0$. D'où, $1 - [B]_v = 0$. Donc $[B]_v = 1$, cad v est un modèle de B .

En utilisant le résultat de l'exercice 8 nous concluons.

Illustration de la propriété

Exemple 1.2.28

a	b	c	$a \Rightarrow b$	$b \Rightarrow c$	$a \Rightarrow c$	$(a \Rightarrow b) \wedge (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$	$(a \Rightarrow b) \wedge (b \Rightarrow c) \wedge \neg(a \Rightarrow c)$
0	0	0	1	1	1	1	0
0	0	1	1	1	1	1	0
0	1	0	1	0	1	1	0
0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	0
1	0	1	0	1	1	1	0
1	1	0	1	0	0	1	0
1	1	1	1	1	1	1	0

Compacité

Théorème 1.2.25 Compacité propositionnelle

Un ensemble de formules **propositionnelles** a un modèle si et seulement si tous ses sous-ensembles finis ont un modèle.

Plan

Préambule

Introduction à la Logique

Logique propositionnelle

Syntaxe

Sens des formules

Equivalences remarquables

Conclusion

La disjonction

La disjonction

- ▶ **associative** $x \vee (y \vee z) \equiv (x \vee y) \vee z$

La disjonction

- ▶ **associative** $x \vee (y \vee z) \equiv (x \vee y) \vee z$
- ▶ **commutative** $x \vee y \equiv y \vee x$

La disjonction

- ▶ **associative** $x \vee (y \vee z) \equiv (x \vee y) \vee z$
- ▶ **commutative** $x \vee y \equiv y \vee x$
- ▶ **idempotente** $x \vee x \equiv x$

La conjonction

La conjonction

- ▶ **associative** $x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$

La conjonction

- ▶ **associative** $x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$
- ▶ **commutative** $x \wedge y \equiv y \wedge x$

La conjonction

- ▶ **associative** $x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$
- ▶ **commutative** $x \wedge y \equiv y \wedge x$
- ▶ **idempotente** $x \wedge x \equiv x$

Distributivité

Distributivité

- ▶ Le produit est distributif sur la somme
$$x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$$

Distributivité

- ▶ Le produit est distributif sur la somme
$$x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$$
- ▶ La somme est distributive sur le produit
$$x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$$

Neutralité et Absorption

Neutralité et Absorption

- ▶ 0 est l'élément neutre de la disjonction $0 \vee x \equiv x$
- ▶ 1 est l'élément neutre de la conjonction $1 \wedge x \equiv x$

Neutralité et Absorption

- ▶ 0 est l'élément neutre de la disjonction $0 \vee x \equiv x$
- ▶ 1 est l'élément neutre de la conjonction $1 \wedge x \equiv x$
- ▶ 1 est l'élément absorbant de la disjonction $1 \vee x \equiv 1$
- ▶ 0 est l'élément absorbant de la conjonction $0 \wedge x \equiv 0$

Les lois de la négation

Les lois de la négation

► $\neg\neg X \equiv X$

Les lois de la négation

▶ $\neg\neg X \equiv X$

▶ $\neg 0 \equiv 1$

Les lois de la négation

▶ $\neg\neg X \equiv X$

▶ $\neg 0 \equiv 1$

▶ $\neg 1 \equiv 0$

Les lois de la négation

▶ $\neg\neg x \equiv x$

▶ $\neg 0 \equiv 1$

▶ $\neg 1 \equiv 0$

▶ $x \wedge \neg x \equiv 0$

Les lois de la négation

▶ $\neg\neg x \equiv x$

▶ $\neg 0 \equiv 1$

▶ $\neg 1 \equiv 0$

▶ $x \wedge \neg x \equiv 0$

▶ $x \vee \neg x \equiv 1$

Les lois de la négation

- ▶ $\neg\neg x \equiv x$
- ▶ $\neg 0 \equiv 1$
- ▶ $\neg 1 \equiv 0$
- ▶ $x \wedge \neg x \equiv 0$
- ▶ $x \vee \neg x \equiv 1$
- ▶ Les lois de De Morgan :
 - ▶ $\neg(x \wedge y) \equiv \neg x \vee \neg y,$
 - ▶ $\neg(x \vee y) \equiv \neg x \wedge \neg y$

Lois de simplification

Propriété 1.2.26

Pour tout x, y nous avons :

- ▶ $x \vee (x \wedge y) \equiv x$
- ▶ $x \wedge (x \vee y) \equiv x$
- ▶ $x \vee (\neg x \wedge y) \equiv x \vee y$

Démonstration.

La preuve est demandée en exercice 13. □

Plan

Préambule

Introduction à la Logique

Logique propositionnelle

Syntaxe

Sens des formules

Equivalences remarquables

Conclusion

Conclusion : Aujourd'hui

- ▶ Introduction et Historique
- ▶ Logique propositionnelle
- ▶ Syntaxe
- ▶ Sens des formules
- ▶ Equivalence remarquables

Conclusion : La prochaine fois

- ▶ Substitutions et remplacements
- ▶ Formes normales
- ▶ Algèbre de Boole
- ▶ Fonctions booléennes
- ▶ Circuit

Conclusion

Merci de votre attention.

Questions ?

Devise d'Oxford

The more I study, the more I know
The more I know, the more I forget
The more I forget, the less I know