UNIVERSITÉ FRANÇOIS - RABELAIS
CAMPUS BLOIS

Logique pour l'informatique

Jean-Yves Antoine

http://www.info.univ-tours.fr/~antoine

MATHEMATIQUES ET INFORMATIQUE

Travaux pionniers … toujours d'actualité

Machine de Türing (1930) : théorie logique des calculateurs

Machine de Von Neumann (1946) : architecture ordinateurs

Mathématiques et informatique

Conception systèmes informatiques	⇨ logique booléenne
Compilation : th. des langages et automates	⇨ logique formelle
Programmation	
- récursion/induction	⇨algèbre
- combinatoire	⇨ analyse (séries)
- structures de données (arbres, graphes)	⇨ maths. discrètes
- programmation logique	⇨ logique
Bases de données	⇨ algèbre relationnelle

Langage Prolog

- **Programmation logique** : application directe de la déduction en LP1
- **Approprié à l'Intelligence Artificielle** … pas au développement Web !
- Découverte ici au titre de la culture générale

TIOBE Programming Community Index (sept. 2012)

Position Sep 2012	Position Sep 2011	Delta in Position	Programming Language	Ratings Sep 2012	Delta Sep 2011	Status
1	2	↑	C	19.295%	+1.29%	A
2	1	↓	Java	16.267%	-2.49%	A
3	6	↑↑↑	Objective-C	9.770%	+3.61%	A
4	3	↓	C++	9.147%	+0.30%	A
5	4	↓	C#	6.596%	-0.22%	A
6	5	↓	PHP	5.614%	-0.98%	A
7	7	=	(Visual) Basic	5.528%	+1.11%	A

Position	Programming Language	Ratings
21	Bash	0.543%
22	Assembly	0.530%
23	SAS	0.528%
24	R	0.440%
25	COBOL	0.431%
26	Fortran	0.430%
27	ABAP	0.427%
28	RPG (OS/400)	0.410%
29	Scheme	0.401%
30	Logo	0.369%
31	Scratch	0.356%
32	Prolog	0.336%

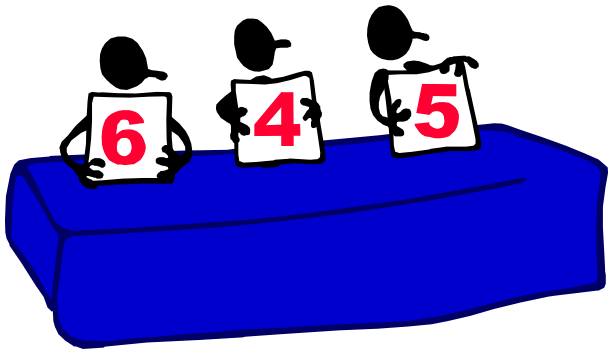
EVALUATION

Contrôle continu intégral

- Deux ou trois courtes interrogations … annoncées ou non
- Projet Prolog comptant comme un CC

Note finale F

- F = CC



Seconde session (si échec)

- Examen papier CT2
- F = CT2 (note de contrôle continu CC ne compte plus)

BIBLIOGRAPHIE

Logique

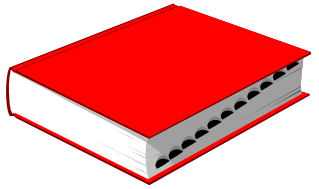
Ouvrages disponibles à la B.U. Blois

P. Lafourcade, S. Devisme, M. Levy (2012) Logique et démonstration automatique, Ellipses
S. Cerrito «Logique pour l'informatique : introduction à la déduction automatique», Vuibert.

Autres ouvrages

A. Aho et J. Ullman, « *Concepts fondamentaux de l'informatique* », Dunod (chap. 12-14)
G. Chazal, « *Elements de logique formelle* »
T. Lucas, I. Berlanger, I. De Greef, « *Initiation à la logique formelle* », De Boeck
R. Cori et D. Lascar, "*Logique mathématique*", Masson (vol. 1)
P. Gochet, P. Gribomont, "*Logique (méthodes pour l'informatique fondamentale)*" (vol. 1)
D. Hofstadter, " *Gödel Escher Bach les brins d'une guirlande éternelle* ", Dunod
C. Jacquemin, "*Logique et mathématiques pour l'informatique et l'IA"*, Masson

Approches logiques de la programmation



D. Gries, "*The science of programming"*, Springer Verlag
L. Sterling & E. Shapiro, "*L'art de Prolog*", Masson

Logique pour l'informatique

Chapitre I — Introduction

INTRODUCTION - Objectifs

1.1. Notions

1.1.1. Vérité et validité
1.1.2. Approches formelles et sémantiques : quelles relations (consistance, complétude)
1.1.3. Approche formelle : axiome, théorème
1.1.4. Approche sémantique : interprétation, modèle, tautologie, équivalence
1.1.5. Déduction : conséquence logique et validité d'un raisonnement
1.1.6. Consistance et complétude

1.2. Pratiques

VERITE ET VALIDITE

Logique : définition

"*Etudes des raisonnements valides*"

- **Philosophie** : raisonnement humain
- **Mathématiques** : validité des démonstrations
- **Informatique** : traitement automatique de l'information

Raisonnement logique

Articulation de plusieurs jugements, c'est-à-dire de propositions auxquelles on peut associer une valeur de vérité

- **Hypothèses ou prémisses** : propositions supposées ou jugées vraies au départ du raisonnement
- **Conclusion** : nouvelle proposition déduite des hypothèses

VERITE ET VALIDITE

- **Raisonnement 1**

✓ valide

Tous les chevaux ont une crinière
Les poneys sont des chevaux

Donc les poneys ont une crinière

- **Raisonnement 2**

✓ valide

Tous les oiseaux volent
L ’autruche est un oiseau

Donc l ’autruche vole

- **Raisonnement 3**

✓ invalide

Tout nombre pair est divisible par 2

Tout nombre divisible par 2 est pair

- **Validité ≠ Vérité**
- **CN validité : propagation de la vérité (fausseté)**

Deux approches ...

Approche syntaxique (formelle)

Étude du point de vue de la forme des énoncés logiques

systèmes formels

Approche sémantique

Étude du point de vue de la propagation de la vérité entre prémisses et conclusions

… une même réalité

LOGIQUE : APPROCHE FORMELLE

- **Axiome**

  Proposition primitive considérée comme non démontrable et admise a priori

  Exemple : *axiomes de la géométrie euclidienne*

- **Théorème**

  Proposition pouvant être démontrée à partir d'axiomes ou d'autres théorèmes à l'aide de raisonnement formels valides. Les axiomes sont considérés comme des théorèmes particuliers.

  Notation $\vdash T$

- **Règle d'inférence**

  Schéma minimal de raisonnement valide qui permet de produire de nouvelles propositions à partir de prémisses qui sont soient des théorèmes, soit des hypothèses.

  Exemple : *modus ponens*

  (P1) Si A alors B
  (P2) A

  (T) Donc B

  Notation $P1, P2 \vdash T$

LOGIQUE : APPROCHE SEMANTIQUE

- **Sens**

  Sens d'un énoncé logique = valeur de vérité { $\wp$,$\Im$ }

- **Interprétation**

  Toute application I attribuant une valeur de vérité aux propositions P d'un système logique : I(P) ∈ { $\wp$,$\Im$ }

- **Modèle**

  Une interprétation I est un modèle d'une proposition logique P ssi I(P)=*V*. On dit que P est **satisfaisable**.

- **Tautologie**

  Une proposition logique P est dite tautologique ssi elle est vraie pour toute interprétation I.

  Notation $\models T$

- **Contradiction**

  Une proposition logique est dite contradictoire, ou insatisfaisable, si elle n'admet aucun modèle.

- **Equivalence**

  Deux propositions logiques A et B sont équivalentes logiquement ssi elles ont la même valeur de vérité pour toute interprétation

  Notation $A \equiv B$

LOGIQUE : APPROCHE SEMANTIQUE (2)

- **Conséquence logique**

Une proposition logique B est la conséquence logique d'une proposition logique A ssi tout modèle de A est un modèle de B

Notation **A ⊨ B**

- **Raisonnement valide**

Un raisonnement est dit valide ssi sa conclusion est la conséquence logique de ses prémisses.

Notation **P1, …, Pn ⊨ B**

EQUIVALENCE ENTRE APPROCHES

- **Consistance** — Un système logique est **sain** (ou **consistant**) si tout théorème obtenu par démonstration formelle est une conséquence logique des axiomes
- **Complétude** — Un système logique est dit **complet** si, pour tout ensemble de formules du système logique, les conséquences logiques de ces formules peuvent être démontrées comme des théorèmes de ces dernières (i.e. en sont des conséquences formelles).
- **Système sain et complet** — Dans un système logique sain et complet, tout théorème est une tautologie et réciproquement

Exemple LP et LP1

Contre-exemple théorème d'incomplétude de Gödel

Logique pour l'informatique

Chapitre IIa — Logique des Propositions (LP) : modélisation et calcul booléen

LOGIQUE DES PROPOSITONS - Objectifs

2.1. Notions

2.1.1. Logique des propositions : que peut-on modéliser avec ?
2.1.2. Syntaxe et formule bien formée (fbf)
2.1.3. Connecteur logique et système complet de connecteurs
2.1.4. Table de vérité
2.1.5. Forme normale (conjonctive, disjonctive)
2.1.6. Fonction booléenne : tableau de Karnaugh

2.2. Pratiques

2.2.1. Représenter un énoncé ou un problème en logique des propositions et savoir quand cette logique est insuffisante
2.2.2. Calculer les interprétations d'une fbf de LP à l'aide d'une table de vérité. S'en servir pour montrer des propriétés de tautologie, contradiction, équivalence
2.2.3. Trouver le modèle d'une fbf par intuition ou calcul de ses interprétations
2.2.4. Construire le tableau de Karnaugh associé à une fbf
2.2.5. Mettre une formule sous sa forme normale minimale par des méthodes différentes (formules d'équivalence ou tableau de Karnaugh)
2.2.6 Trouver la formule logique qui réalise une fonction booléenne donnée (Karnaugh)

2.3. Approfondissement

2.3.1. Montrer qu'un système de connecteurs est complet

- **Exemple**

  Si *le drapeau est vert* ***et si*** *je suis raisonnable* ***alors*** *je* ***ne*** *me baigne* ***pas***
  Je peux me noyer ***ssi*** *je* ***ne*** *suis* ***pas*** *raisonnable* ***ou*** *je* ***ne*** *sais* ***pas*** *nager*
  Le drapeau est rouge ***et*** *je me baigne*

  Je risque la noyade

- **Atome** Jugement de base irréductible : *drapeau_vert*

- **Connecteurs logiques**

Négation	$\neg$	*Ne … pas*
Conjonction	$\wedge$	*Et*
Disjonction	$\vee$	*Ou*
Implication	$\Rightarrow$	*Alors*
Équivalence	$\Leftrightarrow$	*Ssi*

LP : SYNTAXE

- **Vocabulaire** Symboles représentant les atomes : chaînes de caractères
  Symboles des connecteurs logique : { $\wedge$, $\vee$, $\Leftrightarrow$, $\Rightarrow$, $\neg$, ..}
  Symboles de parenthésage : { (,) }

- **Règles de construction des formules bien formées (fbf)**
  - Tout atome P est une fbf de la LP
  - Si F est une fbf, alors (F) est une fbf
  - Si F est une fbf alors $\neg$F est une fbf
  - Si A et B sont deux fbd, alors les formules suivantes sont des fbf : A $\wedge$ B, A $\vee$ B, A $\Leftrightarrow$ B, A $\Rightarrow$B

Exemple et contre-exemples

$\neg\neg\neg$P
$\neg$(Q $\wedge$ P)
$\neg$Q $\wedge\wedge$ P
$\neg$Q $\wedge$ P
$\neg$QQ $\wedge$ PP
P $\neg$Q

- **Priorités d'application des connecteurs logiques**

$\neg$ prioritaire sur { $\wedge$, $\vee$ } prioritaire sur { $\Leftrightarrow$, $\Rightarrow$ }

- **Compositionnalité de la LP (Calul Booléen)**

  La valeur de vérité d'une fbf de la LP dépend uniquement :

  - de l'interprétation des atomes qui la composent
  - de l'emploi des connecteurs logiques entre ces atomes

- **Tables de vérité**

P	Q	P ∧ Q	P ∨ Q	P ⇔ Q	P ⇒ Q
V	F	F	V	F	F
V	V	V	V	V	V
F	F	F	F	V	V
F	V	F	V	F	V

P	¬P
V	F
F	V

- **Autres connecteurs** : ⊕ : Ou exclusif, ↑ : Non-Et, ↓ : Non-Ou, ...

FORMULES D'EQUIVALENCE DE LA LP

Équiv. connecteurs	$P \Rightarrow Q \equiv \neg P \vee Q$
	$P \Leftrightarrow Q \equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P) \equiv (P \wedge Q) \vee (\neg P \wedge \neg Q)$
Double négation	$\neg \neg P \equiv P$
Lois de Morgan	$\neg (P \wedge Q) \equiv \neg P \vee \neg Q$
	$\neg (P \vee Q) \equiv \neg P \wedge \neg Q$
Idempotence	$P \vee P \equiv P \wedge P \equiv P$
Commutativité	$P \wedge Q \equiv Q \wedge P$
	$P \vee Q \equiv Q \vee P$
Associativité	$(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R) \equiv P \wedge Q \wedge R$
	$(P \vee Q) \vee R \equiv P \vee (Q \vee R) \equiv P \vee Q \vee R$
Contradiction	$P \wedge \neg P \equiv$ **Faux**
Tiers-exclus	$P \vee \neg P \equiv$ **Vrai**
Distributivité	$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$
	$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$
Absorption	$P \vee (P \wedge Q) \equiv P$
	$P \wedge (P \vee Q) \equiv P$

FORMES NORMALES

Système complet de connecteur $\{ \neg , \wedge , \vee \}$

Littéral On appelle littéral tout atome ou sa négation

Exemples : Q ou $\neg$R

Forme normale conjonctive

Une fbf de LP est dite sous forme normale conjonctive (fnc) ssi elle se compose d'une conjonction de disjonctions de littéraux

Exemple : $(P \vee Q) \wedge (Q \vee \neg R) \wedge \neg P$ mais aussi …. $P \vee Q$

Forme normale disjonctive

Une fbf de LP est dite sous forme normale disjonctive (fnd) ssi elle se compose d'une disjonction de conjonctions de littéraux

Exemple : $(P \wedge Q) \vee (Q \wedge \neg R) \vee \neg P$ mais aussi …. $P \wedge Q$

FORMES NORMALES

Toute fbf de LP admet une fnc et une fnd (minimales) uniques (à une commutation prête) qui lui sont logiquement équivalentes.

Algorithme de mise sous forme normale

1) **Passage dans le système complet { ¬, ∨, ∧ }**

Remplacement des ⇒ et ⇔ par équivalence

2) **Réduction des négations**

Lois de De Morgan + double négation

3) **Distributivité + commutativité + absorption**

$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$ pour la fnc

$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$ pour la fnd.

FONCTIONS BOOLEENNES

On appelle fonction booléenne toute fbf de la LP exprimée uniquement au moyen des opérations de l'algèbre de Boole :

- addition (+) disjonction ($\vee$)
- multiplication (.) conjonction ($\wedge$)
- complémentation ($\bar{}$) négation ($\neg$)

Sir John Boole (1815-1864)

Exemple

$f(P,Q,R) = (P \wedge Q) \vee (Q \wedge \neg R)$ foncθ booléenne de P, Q et R

Problème classique

Retrouver l'expression d'une fonction booléenne à partir de sa table de vérité.

mintermes / maxtermes

tableaux de Karnaugh

Minterme / Maxterme

Minterme Toute conjonction de littéraux pour lesquels la fonction booléenne est vraie.

Maxterme Toute disjonction des littéraux t.q. la fonction est fausse quand le littéral est faux,

P	Q	F1(P,Q)	
V	V	V	minterme : $P \wedge Q$
V	F	V	minterme : $P \wedge \neg Q$
F	V	F	maxterme : $P \vee \neg Q$
F	F	V	minterme : $\neg P \wedge \neg Q$

Passage d'une fonction booléenne à son expression logique

fnd	disjonction des mintermes de la fonction
fnc	conjonction des maxtermes de la fonction

Exemple

fnd $f1(P,Q) \equiv (P \wedge Q) \vee (\neg P \wedge \neg Q) \vee (P \wedge \neg Q)$

fnc $f1(P,Q) \equiv P \vee \neg Q$

FONCTIONS BOOLEENNES : TABLES DE KARNAUGH

Impliquant

On appelle **impliquant** d'une fonction booléenne *f* toute conjonction *c* de littéraux telle que aucune des variable (atome) de la fonction *f* ne peut rendre en même temps *c* vraie et *f* fausse

Tables de Karnaugh

Fnd conjonction des impliquants issus des mintermes

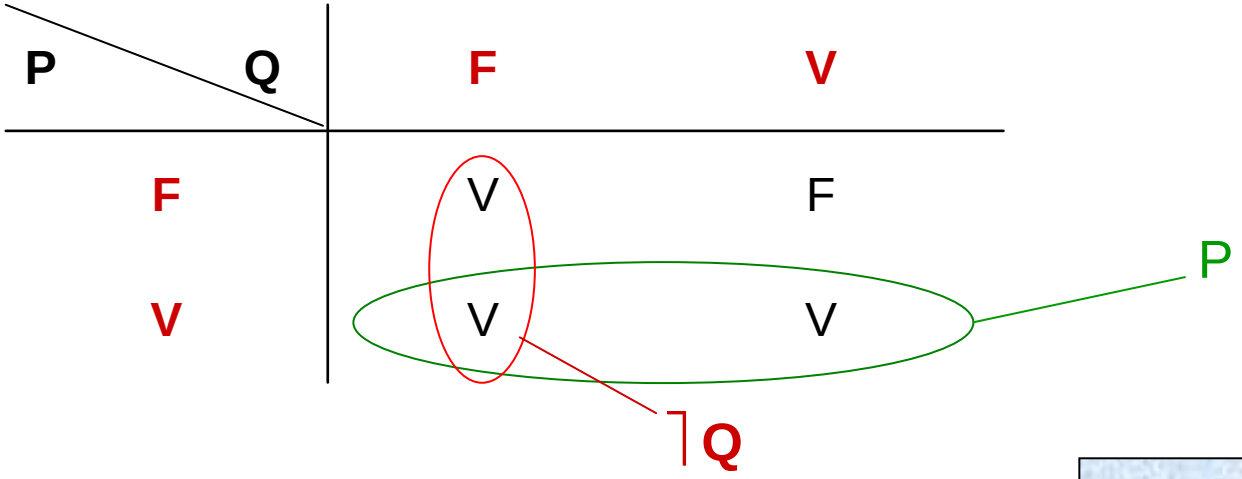
⑤ développement par les « V »

Fnc équivalence avec maxtermes

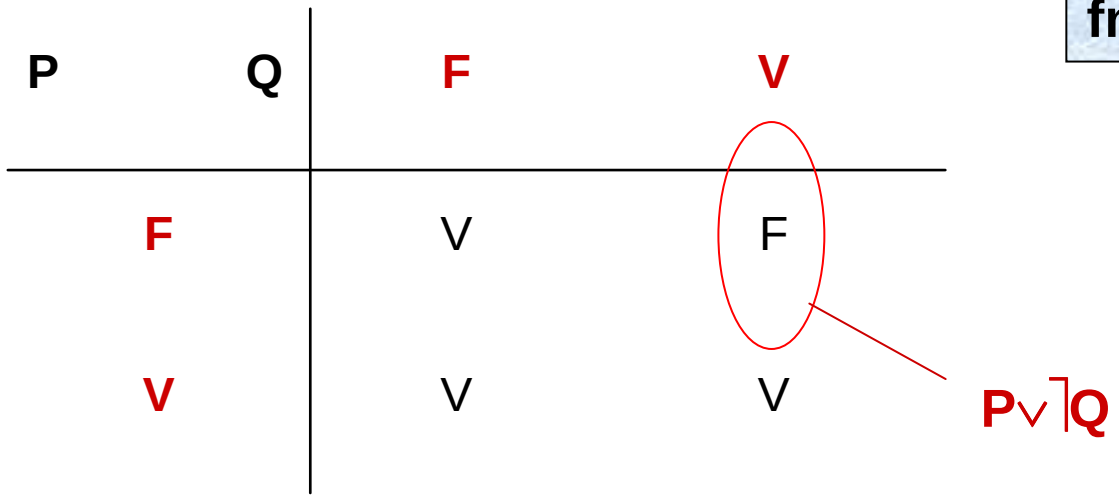
⑤ développement par les « F »

FONCTIONS BOOLEENNES : TABLES DE KARNAUGH

Table de Karnaugh à 2 variables



fnc ≡ fnd ≡ P∨¬Q



FONCTIONS BOOLEENNES : TABLES DE KARNAUGH

Table de Karnaugh à 3 variables

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

PQ \ R	F	V
FF	V	F
FV	F	F
VV	V	V
VF	V	V

PQ \ R	F	V
FF	V	F
FV	F	F
VV	V	V
VF	V	V

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

Logique pour l'informatique

Chapitre IIb — Déduction en Logique des Propositions : résolution

LOGIQUE DES PROPOSITIONS (Déduction) - Objectifs

2.1. Notions

2.1.7. Forme clausale
2.1.8. Principe de réfutation : que dit-il vraiment ? Formule de réfutation
2.1.9. Règle et principe de résolution : que montre-t-il vraiment ?

2.2. Pratiques

2.2.7. Montrer la validité d'un raisonnement en partant de la définition de la conséquence logique : tables de vérité
2.2.8. Montrer la validité d'un raisonnement par des méthodes plus complexes (transformation sur la formule de réfutation, méthode de résolution)
2.2.9. Appliquer ces méthodes à des cas particulier : validité ou contradiction

2.3. Approfondissement

2.3.2. Démontrer le principe de réfutation et la méthode de résolution en LP

DEDUCTION EN LOGIQUE DES PROPOSITIONS

Rappel Un raisonnement logique est valide ssi sa conclusion est la conséquence logique de ses prémisses. On note alors :

P1, ..., Pn |== C

Méthode des tables de vérité

Application directe de la définition de la conséquence logique : il suffit d'évaluer à l'aide d'une table de vérité l'ensemble des modèles de P1$\wedge$ ... $\wedge$ Pn constituent des modèles de C

Principe de déduction par réfutation (théorème)

Pour montrer que P1, ..., Pn |= C (raisonnement valide) il faut et il suffit de montrer que la **formule de réfutation** $\mathcal{F}r \equiv$ P1 $\wedge$...$\wedge$ Pn $\wedge$ $\neg$**C** est contradictoire

On dit alors qu'on a montré la validité par réfutation de la conclusion

LOGIQUE DES PROPOSITIONS : RESOLUTION

Clause On appelle **clause** toute disjonction d'atome ou de littéraux d'atome. Exemples : $P \vee Q$ ou $P \vee \neg Q$

Forme clausale Une formule est dite sous **forme clausale** si elle se présente comme une conjonction de clauses

Remarque — En LP, forme normale = fnc

Règle de résolution (Herbrand, Robinson)

Soient (S1) et (S2) deux clauses appartenant à une formule (S) mise sous forme clausale. S'il existe un atome L tq $L \in$ (S1) et $\neg L \in$ (S2) alors la clause : (R) $\equiv$ $(S1 \setminus \{L\}) \cup (S2 \setminus \{\neg L\})$ dite **résolvante** de (S1) et (S2) est une conséquence logique de (S1) et (S2)

Corollaire : (S) et (S) $\cup$ (R) sont logiquement équivalentes

Exemple — (S1) = $P \vee Q$, (S2) = $R \vee \neg Q$ alors res(S1,S2) = $P \vee R$

LOGIQUE DES PROPOSITIONS : RESOLUTION

Méthode de résolution de Robinson

Pour montrer qu'une formule (S) est contradictoire, il faut et il suffit de produire la clause vide par résolution de l'ensemble des clauses issues de (S) mise sous forme clausale

Soit un raisonnement dont on cherche à montrer la validité

1 — Construction de la formule de réfutation associée

2 — Mise sous forme clausale de la formule de réfutation ($\mathcal{F}r$)

3 — Tq la clause vide n'appartient pas à ($\mathcal{F}r$) ou bouclage

3.1. appliquer la résolution sur 2 clauses de ($\mathcal{F}r$)

3.2. ajouter la résolvante à ($\mathcal{F}r$)

4 — Si clause vide : raisonnement valide. Sinon, non valide.

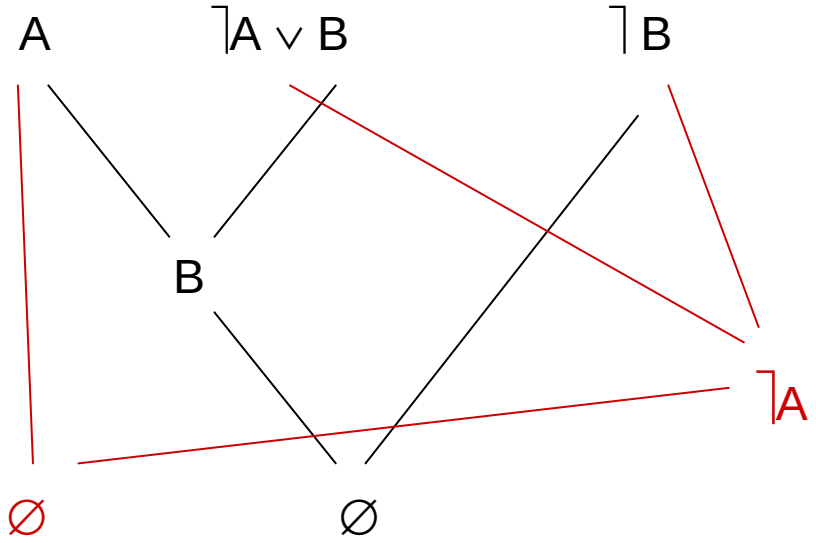
Méthode de résolution de Robinson : exemple

Modus ponens A, A $\Rightarrow$ B a-t-il pour conséquence logique B ?

Formule de réfutation $\mathcal{F}r \equiv A \wedge (A \Rightarrow B) \wedge \neg B$

Formule clausale $\mathcal{F}r \equiv A \wedge (\neg A \vee B) \wedge \neg B$ trois clauses

Résolution



2 démonstrations

$\mathcal{F}r$ contradictoire donc

Raisonnement valide