logique mathématique

Université Mohamed Seddik Ben Yahia -Jijel-
2ième année Licence informatique

F. BOUDJERIDA

chapitre 2

logique propositionnelle

Le langage propositionnel L
L'alphabet

L'élément de base de la logique propositionnel est la ... proposition. Il s'agit d'un énoncé **qui peut être vrai ou faux.**

un ensemble, fini ou dénombrable, de variables p*ropositionnelles* notées *p , q , r , . . .*

Le langage propositionnel L
L'alphabet

exemple de propositions

$2 + 2 = 4$

$1 + 1 = 0$

Le soleil brille

Il a les yeux rouges

un carr´e est un polygone

tout homme est mortel $\cdots$

Le langage propositionnel L

L'alphabet

Par exemple

Jeber mange une pomme

Ali pense qu'il pleuvra demain

Tous les nombres impairs sont des nombres premiers

Si le moteur n'est pas en marche et les phares sont allumés et la porte avant gauche est ouverte alors l'alarme sonne

Remarquons que certaines phrases de la langue courante ne sont pas de propositions, elles ne sont ni vraies ni fausses : Sortez immédiatement !

Demain il pleuvra.

Ce livre est-il bon ?

Le langage propositionnel L
L'alphabet

Vocabulaire

- un ensemble infini dénombrable de variables propositionnelles ou **propositions** $\mathcal{P}$
- les constantes : 0 (*Faux*, F ou $\perp$) et 1 (*Vrai*, V, ou $\top$)
- les connecteurs : $\neg \quad \wedge \quad \vee \quad \rightarrow \quad \leftrightarrow$
- les parenthèses

Le langage propositionnel L

Exemples de formules propositionnelles

- $(\neg P \land Q) \lor (P \land \neg Q)$
- $P \lor (Q \land R)$
- $(P \to (Q \to P))$
- $(A \to B) \to ((A \to (B \to C)) \to (A \to C))$
- $\neg\neg A \to A$

Le langage propositionnel L
Aspects déductifs et sémantiques de la logique

- On peut appréhender un logique soit sous son aspect déductif (ou syntaxique) soit sous son aspect sémantique.

Dans l’approche syntaxique

on s’intéresse à la notion de *preuve formelle*. À partir d’axiomes et d’hypothèses on déduit de nouveaux faits en appliquant un ensemble de règles d’inférence.
Par exemple, la règle dite du *modus ponens* nous permet de déduire à partir des deux énoncés

S’il pleut alors il y a des nuages.
Il pleut.

l’énoncé Il y a des nuages.

Le langage propositionnel L
Aspects déductifs et sémantiques de la logique

Dans l'approche sémantique

on s'intéresse à évaluer la vérité ou fausseté de certains énoncés dans certaines situations, à voir si un énoncé est une conséquence logique d'un autre, à vérifier la consistance d'un ensemble d'énoncés. Par exemple, est-il possible que les deux énoncés

Paul ne mange pas de chocolat
Paul mange du pain ou du chocolat

soient vrais simultanément ?

Approche Sémantique

validité, satisfiabilité

Valuation

La sémantique associé une fonction de valuation ϑ telle que

$$\vartheta: \quad Vp \rightarrow \{0; 1\} \qquad \begin{array}{l} 0 - F \\ 1 - V \end{array}$$

Soit on dit valuation ou interprétation

- Grace à cela on pourra calculer la valeur de vérité de la formule par rapport à cette interprétation.

Approche Sémantique
validité, satisfiabilité

- **tables de vérité des connecteurs**

Le rôle des tables de vérité des connecteurs est de fixer formellement le comportement de chacun des connecteurs.

p	q	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
vrai	vrai	vrai	vrai	vrai	vrai
vrai	faux	faux	vrai	faux	faux
faux	vrai	faux	vrai	vrai	faux
faux	faux	faux	faux	vrai	vrai

Approche Sémantique
validité, satisfiabilité

Ces fonctions sont définies de la façon suivante :

- $T_{\wedge}$ (x; y) = vrai ssi x = vrai et y = vrai
- $T_{\vee}$ (x; y) = vrai ssi x = vrai ou y = vrai
- $T_{\rightarrow}$ (x; y) = faux ssi x = vrai et y = faux
- $T_{\neg}$ (x) = vrai ssi x = faux

Selon la table de vérité ; deux propositions p et q équivalentes sont des propositions ayant les mêmes valeurs de vérités.

Approche Sémantique
validité, satisfiabilité

- **Valeur de vérité d'une formule**

Les notions d'interprétation et de table de vérité des connecteurs permettent de calculer la valeur de vérité pour une interprétation donnée de la formule toute entière.

- **exemple** : Chaque ligne correspond à une interprétation.

i_1	s	t	$(s \vee t)$	$(t \wedge s)$	$(s \vee t) \rightarrow (t \wedge s)$
i_1	v	v	v	v	v
i_2	v	f	v	f	f
i_3	f	v	v	f	f
i_4	f	f	f	f	v

Remarque

si F contient n variables propositionnelles alors il existe exactement 2^n interprétations de F.

Définition

- Une formule α est **valide** est on note $\models \alpha$ si elle est vraie pour toute interprétation (on dit aussi que c'est une **tautologie**).
- Elle est **satisfiable** s'il existe en moins une interprétation pour laquelle elle est vraie
- et elle est **contradictoire** ou **insatisfiable** sinon, c'est-à-dire quand la formule est fausse quelle que soit l'interprétation.

Définition

- On étend ces notions à un ensemble E fini ou infini de formules:
- L'ensemble E est valide si pour toute interprétation I et toute formule $\alpha \in$ E on a

  $V_I(\alpha)=v$.
- L'ensemble E est satisfiable s'il existe une interprétation I telle que pour toute formule $\alpha \in$ E on a $V_I(\alpha)=v$.
- L'ensemble E est insatisfiable si pour toute interprétation I, il existe une formule $\alpha \in$ E telle que $V_I(\alpha)=f$.

Remarques

- Dire que deux formules F et G sont **valides** est équivalent à dire que l'ensemble {F,G} est valide. Par contre si deux formules sont satisfiables, cela ne veut pas dire que l'ensemble {F,G} est satisfiable.
  - Il suffit de prendre {x,¬ x} qui est insatisfiable car une interprétation I ne peut pas rendre vrai à la fois x et ¬ x alors que, pour chaque formule prise séparément, on peut trouver une interprétation qui la rend vraie.

Remarques

- De manière duale si F et G sont **insatisfiables**, il en est de même de l'ensemble {F,G} mais le contraire n'est pas vrai, un ensemble de formules peut être insatisfiable alors que chaque formule individuellement est satisfiable.
- Un ensemble de formules **valide** est a fortiori satisfiable. Comme la valeur d'une formule est vraie (resp. fausse) si et seulement si la valeur de sa négation est fausse (resp. vraie), on en déduit une correspondance entre validité et insatisfiabilité.

Théorème

- La formule F est **valide** (resp. **insatisfiable**) ssi la formule ¬ F est insatisfiable (resp. valide).
- L'ensemble fini de formules $\{A_1,\ldots,A_n\}$ est **valide** (resp. **satisfiable**, resp. **insatisfiable**) ssi la formule $A_1 \wedge \ldots \wedge A_n$ est valide (resp. satisfiable, resp. insatisfiable)

Définition(Modèle d'une formule)

- Un **modèle** d'une formule F est une interprétation I des variables de cette formule qui rend la formule vraie c'est-à-dire telle que $V_I(F)=1$.

Exemple *Soit la formule P définie comme* $\neg(x \wedge y) \rightarrow \neg y$

- *L'interprétation* $\{x \mapsto 1; y \mapsto 0\}$ *est un modèle, par contre* $\{x \mapsto 0; y \mapsto 1\}$ *n'en est pas un.*

Définition (Conséquence logique $\models$)

- *On dira que P est une* **conséquence logique** *d'un ensemble de formules* $\{A_1,\ldots,A_n\}$ *et on écrira* $\boldsymbol{A_1,\ldots,A_n \models P}$ *si tout modèle de* $\{A_1,\ldots,A_n\}$ *est aussi un modèle de P.*

Remarque

- Si A est une tautologie, on peut écrire $\vDash A$.
- Si $A \rightarrow B$ est une tautologie($\vDash A \rightarrow B$), on dit de A qu'elle implique logiquement B.
- Si $A \leftrightarrow B$ est une tautologie($\vDash A \leftrightarrow B$), alors A et B sont dites logiquement équivalentes et on note $A \equiv B$.

Proposition

Pour toutes formules $A_1,\ldots,A_n$ et P, on a les propriétés suivantes:

- ***$A_1,\ldots,A_n \models P$ ssi $A_1 \wedge \ldots \wedge A_n \rightarrow P$ est valide***
- ***si $\{A_1,\ldots,A_n\}$ est insatisfiable alors pour toute formule P, on a $A_1,\ldots,A_n \models P$***
- ***$A_1,\ldots,A_n \models P$ ssi l'ensemble de formules $\{A_1,\ldots,A_n,\neg P\}$ est insatisfiable.***
-

Théorème

Si $\models A$ et $\models A \rightarrow B$ alors $\models B$

Vérité et substitution

- Soit β une formule où figure la variable propositionnelle p, et soit β' la formule obtenue à partir de β en substituant à p en toutes ses occurrences une formule α.
- La validité et l'insatisfiabilité qui sont des propriétés de toutes les interprétations ne changent pas lorsque l'on effectue une substitution dans une formule, un ensemble de formules et ne change pas la notion de conséquence logique.

Vérité et substitution

- Si une formule β est valide (resp. insatisfiable) alors β' est valide (resp. insatisfiable).
- Si un ensemble de formules E est valide (resp. insatisfiable) alors

$E[p\leftarrow \alpha] =^{def} \{\beta' [p\leftarrow \alpha] | \beta' \in E\}$ est valide (resp. insatisfiable).

- Si $A_1,\ldots,A_n \vDash B$ alors

$A_1[x\leftarrow \alpha],\ldots,A_n[x\leftarrow \alpha] \vDash B[x\leftarrow \alpha]$.

Équivalences booléennes

- On utilise plusieurs connecteurs booléens mais il y a des équivalences entre ces connecteurs.
- On écrira $P \equiv Q$ si $P \models Q$ et $Q \models P$ ce qui revient aussi à dire que les valeurs de vérité de *P* et *Q* coïncident pour toute interprétation
- **Lois algébriques :l**'ensemble des booléens avec les opérations de conjonction et de disjonction forme ce que l'on appelle une *Algèbre de Boole*.

Équivalences booléennes

La conjonction et la disjonction sont associatifs et commutatifs :

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

T et ⊥ sont des éléments neutres ou absorbants :

- $P \wedge \mathrm{T} \equiv P \quad P \vee \mathrm{T} \equiv \mathrm{T} \quad P \vee \perp \equiv P \quad P \wedge \perp \equiv \perp$

Les lois de Morgan établissent le comportement de la négation par rapport aux autres connecteurs :

- $\neg \perp \equiv \mathrm{T} \quad \neg \mathrm{T} \equiv \perp \quad \neg\neg P \equiv P$
- $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$
- $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$

Équivalence entre connecteurs.

Certains connecteurs peuvent se définir en fonction d'autres :

- $\neg P \equiv P \rightarrow \perp$
- $P \rightarrow Q \equiv \neg P \vee Q$
- $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$

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

Formes normales

- Les équivalences mises en évidence ci-dessus montrent que la même propriété peut s'exprimer de plusieurs manières différentes.
- On a par exemple déjà établi dans le paragraphe précédent que l'on pouvait se passer des connecteurs $\rightarrow$ et $\leftrightarrow$ en utilisant à la place les connecteurs $\neg$, $\vee$ et $\wedge$. On constate que dans le cas de l'équivalence, la formule transformée est deux fois plus grosse que la formule initiale, donc l'élimination du connecteur peut avoir un coût important.

Forme normale de négation(FNN)

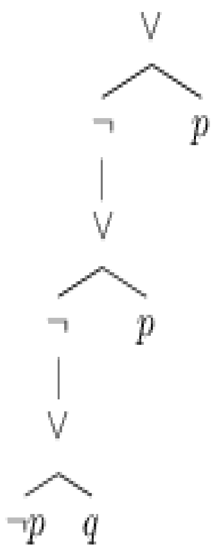
Les lois de Morgan rappelées dans la section permettent de distribuer la négation vers les variables propositionnelles.

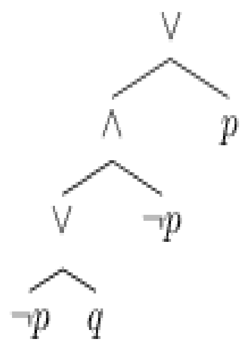
- **Définition 1 (Littéral)** On appelle littéral une formule qui est soit une formule atomique(variables propositionnelles x), soit la négation d'une formule atomique (variables propositionnelles ¬ x).
- **Proposition 1** Toute formule propositionnelle est équivalente à une formule propositionnelle qui ne comporte pas de connecteur ¬ sauf dans un littéral.
- Une telle formule est dite en forme normale de négation.

Forme normale de négation(FNN)

Exemple Soit la formule $((p \to q) \to p) \to p$ dans laquelle on élimine les implications, cette formule devient : $\neg (\neg (\neg p \vee q) \vee p) \vee p$.

- On peut ensuite repousser les négations en commençant par l'extérieur:





On obtient la formule :

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

qui peut se simplifier en

$\neg p \vee p$ puis en 1.

Forme normale conjonctive (FNC)

- **Définition 2 (Clause)** On appelle clause une formule propositionnelle qui n'utilise que des littéraux et la disjonction.
- Comme la disjonction est associative et commutative, la clause peut toujours s'écrire de la forme $l_1 \vee l_2 \vee \ldots \vee l_n$ avec l_i un littéral et on peut réarranger les littéraux dans l'ordre que l'on veut (par exemple en ordonnant les variables propositionnelles, puis pour chaque variable en mettant d'abord les littéraux positifs puis les négatifs). Par ailleurs, on a $P \vee P \equiv P$ et $P \vee \neg P \equiv 1$.
- On peut se ramener à une clause dans laquelle chaque variable propositionnelle apparaît au plus une fois sous forme positive ou négative.
- Par convention, la clause associée à un ensemble vide de littéraux est interprétée comme la formule fausse 0.

Forme normale conjonctive (FNC)

Proposition 3

Soient L_1 et L_2 des ensembles de littéraux, si $L_1 \subseteq L_2$ alors la clause correspondant à L_2 est une conséquence logique de la clause correspondant à L_1.

Définition 3 (FNC)

Une formule est dite en forme normale conjonctive si elle s'écrit comme une conjonction de clauses.

Proposition 4

Toute formule P admet une forme normale conjonctive, c'est-à-dire qu'il existe une formule Q équivalente à P et qui est en forme normale conjonctive.

Forme normale conjonctive (FNC)

Exemple 2 Soit la formule P : $(p \rightarrow q) \rightarrow (\neg p \wedge q)$, on commence par écrire la table de vérité :

- Les lignes qui nous intéressent sont la première et la dernière et on doit exprimer une formule qui dit exactement que l'on n'est pas dans un de ces deux cas.
- La formule qui dit que l'on n'est pas dans le cas de la ligne 1 est $\neg p \vee \neg q$ et la formule qui dit que l'on n'est pas dans le cas de la ligne 4 est $p \vee q$.

p	q	$\neg p \wedge q$	$p \rightarrow q$	$(p \rightarrow q) \rightarrow (\neg p \wedge q)$
1	1	0	1	0
1	0	0	0	1
0	1	1	1	1
0	0	0	1	0

La forme normale conjonctive est donc $(\neg p \vee \neg q) \wedge (p \vee q)$.

Forme normale disjonctive(FND)

Définition 4 (FND)

Une conjonction élémentaire est une formule uniquement composée de littéraux et de conjonctions.

Une formule P est en forme normale disjonctive si elle s'écrit comme une disjonction de conjonctions élémentaires.

Proposition 6 Pour toute formule propositionnelle P, il existe une formule Q équivalente en forme normale disjonctive.

Preuve: Comme précédemment, on peut construire la table de vérité de la formule P par rapport aux variables propositionnelles $\{p_1,\ldots,p_n\}$. On s'intéresse à l'ensemble des n-uplets $(b_1,\ldots,b_n)$, pour lesquels la table de vérité donne la valeur 1. Pour chaque ligne on construit une conjonction élémentaire $l_1 \wedge l_2 \wedge \ldots \wedge l_n$ avec $l_i=p_i$ si $b_i=1$ et $l_i=\neg\, p_i$ si $b_i=0$. Il suffit ensuite de prendre la disjonction de ces formules pour trouver une formule équivalente à P.

Forme normale disjonctive(FND)

Exemple 3 En reprenant la formule P de l'exemple 2, on obtient la forme normale disjonctive : $(p \wedge \neg q) \vee (\neg p \wedge q)$

- Si la formule est en forme normale disjonctive alors il est facile de trouver un modèle. On suppose que les branches de la disjonction sont en forme "simplifiée" c'est-à-dire qu'une variable propositionnelle apparaît dans un seul littéral. Il suffit de choisir une des branches de la disjonction. Pour chaque variable propositionnelle x qui apparaît dans un littéral *l* on prend dans le modèle $x \mapsto 1$ si $l=x$ et $x \mapsto 0$ si $l=\neg x$. Il peut arriver qu'il n'y ait aucune composante en forme simplifiée: c'est le cas où la formule initiale est équivalente à 0 et donc il n'y a pas de modèle.
- Cependant, la mise en forme normale n'est pas forcément la meilleure manière de trouver un modèle.
- En effet, cette forme normale peut être exponentiellement plus grosse que la formule initiale.

Principe de résolution de Robinson

- Cette règle d'inférence a été inventée par Robinson en 1965 . Elle est surtout utilisée dans les systèmes de preuve automatique (démonstrateurs de théorème, systèmes de programmation logique). C'est l'unique règle du système , donc il n'y a pas de problème de choix de la bonne règle à appliquer. Par contre on peut l'appliquer de multiples manières.
- Cette règle travaille uniquement sur des clauses. Il faut donc commencer par tout mettre en forme FNC $C1 \wedge C2 \wedge \ldots \wedge Cn$ où chaque clause Ci est de la forme $L1 \vee \cdots \vee Lm$. Les Lj sont des littéraux de la forme V ou $\neg V$ où V est une variable.

La règle elle-même s'énonce comme suit :

- Si on a deux clauses

  C1 = (p ∨ L1 ∨ L2 ∨ ... ∨ Lm),

  C2 = (¬p ∨ M1 ∨ M2∨..∨ Mn)

- on peut déduire le résolvant de C1 et C2 qui est la clause L 1 ∨ L 2 ∨ ... ∨ L m ∨ M1 ∨ M2 ∨ ... ∨ Mn.
- En d'autre termes, si un littéral et son complément apparaissent dans deux clauses on peut déduire une nouvelle clause à partir de celles-ci.

Exemple

. Application du principe de résolution

$C1 = (p \vee q \vee \neg r \vee s)$

$C2 = (q \vee \neg p \vee t)$

Résolution sur p et ¬p.

$Rés(C1,C2)= (q \vee \neg r \vee s \vee q \vee t)$,

Exemple2

. On veut prouver

$\{(p \vee t) \rightarrow q \, , \, r \rightarrow (t \vee s) \, , \, (q \wedge t) \rightarrow u \, , \, p \, , \, \neg s \, , \, r\} \vDash u$

On met les formules sous forme de clauses :

1. $(p \vee t) \rightarrow q \equiv \neg (p \vee t) \vee q \equiv (\neg p \wedge \neg t) \vee q$

$$\equiv (\neg p \vee q) \wedge (\neg t \vee q)$$

ce qui donne deux clauses : $(\neg p \vee q)$ et $(\neg t \vee q)$

2. $r \rightarrow (t \vee s) \equiv \neg r \vee (t \vee s) \equiv \neg r \vee t \vee s$
3. $q \wedge t) \rightarrow u \equiv \neg(q \wedge t) \vee u \equiv (\neg q \vee \neg t) \vee u \equiv \neg q \vee \neg t \vee u$

Exemple2

On applique ensuite le principe de résolution à partir des six clauses obtenues :

1. ¬p ∨ q
2. ¬t ∨ q
3. ¬r ∨ t ∨ s
4. ¬q ∨ ¬t ∨ u
5. p
6. ¬s
7. r
8. q (rés(1,5) sur p)
9. t ∨ s (rés(3,7) sur r)
10. t (rés(6,9) sur s)
11. ¬t ∨ u (rés(4,8) sur q)
12. u (rés(10,11) sur t)

Exemple2

On ne peut pas appliquer le principe sur deux variables en même temps. Si on a

C1 = (p ∨ ¬q ∨ ¬r ∨ s)

C2 = (q ∨ ¬p ∨ t)

on peut déduire

¬q ∨ ¬r ∨ s ∨ q ∨ t (rés sur p) ou p ∨ ¬r ∨ s ∨ ¬p ∨ t (rés sur q)

mais pas ¬r ∨ s ∨ t ! ("rés " sur p et q)

L'utilisation pratique du principe de résolution se base sur le théorème suivant :

Théorème

Le résolvant C est satisfiable ssi les clauses parentes C1 et C2 le sont (simultanément)

À partir de ce théorème on peut créer une procédure de résolution pour vérifier si un ensemble *S* de clauses est satisfiable.

- On définit $S0 := S$
- On calcule une séquence $S1$, $S2$, . . . avec la règle

(a) soient $C1$ et $C2$ deux clauses de Si dont le résolvant est C
(b) on définit $S_{i+1} := Si \cup \{C\}$
(c) si au cours du processus on trouve un C = [] alors $S0$ était inconsistant
(d) si $S_{i+1} = Si$ pour tout choix de $C1$ et $C2$, alors $S0$ était satisfiable

Théorème

- Une dérivation de [] à partir de S s'appelle une **réfutation** de *S*.
- La technique de ***preuve par*** **réfutation** consiste à prouver que *A* est une conséquence logique de *S* en réfutant

  $S0 = S \cup \{\neg A\}$.

Exemple

Soit $S = \{p, \neg p \vee q, \neg p \vee \neg q \vee r\}$. On veut montrer que $S \vDash r$.

1. $S0 = \{p, \neg p \vee q, \neg p \vee \neg q \vee r, \neg r\}$
2. $S1 = \{p, \neg p \vee q, \neg p \vee \neg q \vee r, \neg r, \neg p \vee \neg q\}$
3. $S2 = \{p, \neg p \vee q, \neg p \vee \neg q \vee r, \neg r, \neg p \vee \neg q, \neg p\}$
4. $S3 = \{p, \neg p \vee q, \neg p \vee \neg q \vee r, \neg r, \neg p \vee \neg q, \neg p, [\,]\}$

donc $S \vDash r$

Propriétés de consistance et de complétude :

Dans le calcul propositionnel, il y a deux aspects :

- **Aspect sémantique (théorie des modèles) :**Qui introduit les notions de satisfiabilité, de tautologie et de conséquences logiques.
- **Aspect syntaxique (théorie de démonstration):** On s’intéresse à montrer qu’une formule est un théorie ou qu’une formule déduite à partir d’un ensemble de formules**.**
- Existe-t-il un rapport entres les théorèmes et les tautologies ?
- Existe-t-il un rapport entres la notion de conséquence logique et celle de déduction ?

Théorème de consistance

- Toutes formules démontrable est une tautologie Si $\vdash \beta$ alors $\models \beta$
- Un système d'inférence est **consistant** si tous les théorèmes sont des conséquences logiques des formules des axiomes (tout ce qui est démontrable est « vrai »).
- Ou de manière équivalence, si on peut produire *G* à partir de l'ensemble de formules *F*, noté $F \vdash G$, alors G est une conséquence de *F* ($F \models G$)

Théorème de complétude

- Si $\models \beta$ alors $\vdash \beta$
- Un système d'inférence est **complet** s'il permet de produire *toutes* les conséquence logiques des axiomes, c'est-à-dire si toute conséquence des axiomes est un théorème (tout ce qui est vrai est démontrable).
- Ou, de manière équivalente, $F \models G$ entraîne

  $F \vdash G$

Le système d'inférence naturel est consistant et complet.

Principe de résolution de Robinson

- Cette règle d'inférence a été inventée par Robinson en 1965 . Elle est surtout utilisée dans les systèmes de preuve automatique (démonstrateurs de théorème, systèmes de programmation logique). C'est l'unique règle du système , donc il n'y a pas de problème de choix de la bonne règle à appliquer. Par contre on peut l'appliquer de multiples manières.
- Cette règle travaille uniquement sur des clauses. Il faut donc commencer par tout mettre en forme FNC $C1 \wedge C2 \wedge \ldots \wedge Cn$ où chaque clause Ci est de la forme $L1 \vee \cdots \vee Lm$. Les Lj sont des littéraux de la forme V ou ¬V où V est une variable.

La règle elle-même s’énonce comme suit :

- Si on a deux clauses

  C1 = ($p \vee L1 \vee L2 \vee \dots \vee Lm$),

  C2 = ($\neg p \vee M1 \vee M2 \vee .. \vee Mn$)

- on peut déduire le résolvant de C1 et C2 qui est la clause $L\,1 \vee L\,2 \vee \dots \vee L\,m \vee M1 \vee M2 \vee \dots \vee Mn$.
- En d’autre termes, si un littéral et son complément apparaissent dans deux clauses on peut déduire une nouvelle clause à partir de celles-ci.

Exemple

. Application du principe de résolution

$C1 = (p \vee q \vee \neg r \vee s)$

$C2 = (q \vee \neg p \vee t)$

Résolution sur p et $\neg p$.

$Rés(C1,C2)= (q \vee \neg r \vee s \vee t)$,

Exemple2

. On veut prouver

$\{(p \vee t) \rightarrow q\ ,\ r \rightarrow (t \vee s)\ ,\ (q \wedge t) \rightarrow u\ ,\ p\ ,\ \neg s\ ,\ r\ \} \vDash u$

On met les formules sous forme de clauses :

1. $(p \vee t) \rightarrow q \equiv \neg (p \vee t) \vee q \equiv (\neg p \wedge \neg t) \vee q$

   $\equiv (\neg p \vee q) \wedge (\neg t \vee q)$

 ce qui donne deux clauses : $(\neg p \vee q)$ et $(\neg t \vee q)$

2. $r \rightarrow (t \vee s) \equiv \neg r \vee (t \vee s) \equiv \neg r \vee t \vee s$
3. $q \wedge t) \rightarrow u \equiv \neg(q \wedge t) \vee u \equiv (\neg q \vee \neg t) \vee u \equiv \neg q \vee \neg t \vee u$

Exemple2

On applique ensuite le principe de résolution à partir des six clauses obtenues :

1. ¬p ∨ q
2. ¬t ∨ q
3. ¬r ∨ t ∨ s
4. ¬q ∨ ¬t ∨ u
5. p
6. ¬s
7. r
8. q (rés(1,5) sur p)
9. t ∨ s (rés(3,7) sur r)
10. t (rés(6,9) sur s)
11. ¬t ∨ u (rés(4,8) sur q)
12. u (rés(10,11) sur t)

Exemple2

On ne peut pas appliquer le principe sur deux variables en même temps. Si on a

$C1 = (p \vee \neg q \vee \neg r \vee s)$

$C2 = (q \vee \neg p \vee t)$

on peut déduire

$\neg q \vee \neg r \vee s \vee q \vee t$ (rés sur p) ou $p \vee \neg r \vee s \vee \neg p \vee t$ (rés sur q)

mais pas $\neg r \vee s \vee t$! ("rés " sur p et q)

L'utilisation pratique du principe de résolution se base sur le théorème suivant :

Théorème

Le résolvant C est satisfiable ssi les clauses parentes C1 et C2 le sont (simultanément)

À partir de ce théorème on peut créer une procédure de résolution pour vérifier si un ensemble S de clauses est satisfiable.

- On définit $S0 := S$
- On calcule une séquence $S1$, $S2$, . . . avec la règle

  (a) soient $C1$ et $C2$ deux clauses de Si dont le résolvant est C
  (b) on définit $S_{i+1} := Si \cup \{C\}$
  (c) si au cours du processus on trouve un C = [] alors $S0$ était inconsistant
  (d) si $S_{i+1} = Si$ pour tout choix de $C1$ et $C2$, alors $S0$ était satisfiable

Théorème

- Une dérivation de [] à partir de S s'appelle une **réfutation** de *S*.
- La technique de ***preuve par* réfutation** consiste à prouver que *A* est une conséquence logique de *S* en réfutant

$S0 = S \cup \{\neg A\}$.

Exemple

Soit $S = \{p, \neg p \vee q, \neg p \vee \neg q \vee r\}$. On veut montrer que $S \models r$.

1. $S0 = \{p, \neg p \vee q, \neg p \vee \neg q \vee r, \neg r\}$
2. $S1 = \{p, \neg p \vee q, \neg p \vee \neg q \vee r, \neg r, \neg p \vee \neg q\}$
3. $S2 = \{p, \neg p \vee q, \neg p \vee \neg q \vee r, \neg r, \neg p \vee \neg q, \neg p\}$
4. $S3 = \{p, \neg p \vee q, \neg p \vee \neg q \vee r, \neg r, \neg p \vee \neg q, \neg p, [\,]\}$

donc $S \models r$

Propriétés de consistance et de complétude :

Dans le calcul propositionnel, il y a deux aspects :

- **Aspect sémantique (théorie des modèles) :**Qui introduit les notions de satisfiabilité, de tautologie et de conséquences logiques.
- **Aspect syntaxique (théorie de démonstration):** On s'intéresse à montrer qu'une formule est un théorie ou qu'une formule déduite à partir d'un ensemble de formules**.**
- Existe-t-il un rapport entres les théorèmes et les tautologies ?
- Existe-t-il un rapport entres la notion de conséquence logique et celle de déduction ?

Théorème de consistance

- Toutes formules démontrable est une tautologie Si $\vdash \beta$ alors $\models \beta$
- Un système d'inférence est **consistant** si tous les théorèmes sont des conséquences logiques des formules des axiomes   (tout ce qui est démontrable est « vrai »).
- Ou de manière équivalence, si on peut produire *G* à partir de l'ensemble de formules *F*, noté $F \vdash G$, alors G est une conséquence de *F* ($F \models G$)

Théorème de complétude

- Si $\models \beta$ alors $\vdash \beta$
- Un système d'inférence est **complet** s'il permet de produire *toutes* les conséquence logiques des axiomes, c'est-à-dire si toute conséquence des axiomes est un théorème (tout ce qui est vrai est démontrable).
- Ou, de manière équivalente, $F \models G$ entraîne $F \vdash G$

Le système d'inférence naturel est consistant et complet.