Outils mathématiques pour l'informatique

HAUTE ÉCOLE DE NAMUR-LIÈGE-LUXEMBOURG

Informatique – bloc 1

Module 3 : La logique des propositions

La logique mathématique (du moins, celle qui sera abordée dans ce cours) se fonde sur un principe appelé le ***principe du tiers exclu***. Celui-ci signifie qu'on ne considère que deux valeurs possibles (pas de troisième — d'où le nom de « tiers » ou « troisième » exclu) : ***vrai*** et ***faux***. Les éléments manipulés en logique mathématique sont des phrases qui ne peuvent être que vraies ou fausses, sans demi-mesure, sans niveau de gris.

Cette théorie mathématique sert de base à de nombreux domaines d'étude tels que la conception de circuits électriques, la construction de conditions dans les algorithmes, ou encore le stockage d'informations dans une mémoire informatique. C'est pour cela qu'il existe plusieurs manières de noter ces deux valeurs. En voici quelques-unes.

vrai	V	true	T	1
faux	F	false	F	0

Dans ce module, on s'intéresse tout d'abord à ces phrases qui ne peuvent être que vraies ou fausses puis on examine comment les assembler, au moyen de diverses opérations, pour formuler des composites plus complexes. On observe ensuite comment déduire la valeur de vérité (vrai ou faux) de ces composites.

Plan du module

1 Propositions et assertions

Les éléments de base de la logique sont les propositions. Une ***proposition*** est une phrase, une déclaration, un fait, qui est soit vrai soit faux. Voici quelques exemples de propositions (la 1^re^ et la 3^e^ étant vraies et les deux autres, fausses).

- Bruxelles est la capitale de la Belgique.
- Namur est située sur la côte belge.
- 1 + 1 = 2
- 3 * 5 = 35

À l'inverse, il existe en français toute une série de phrases qui ne sont pas des propositions, soit parce qu'elles ne décrivent pas un fait vrai ou faux, soit parce que leur valeur de vérité dépend d'éléments inconnus. En voici quelques exemples :

- À quelle heure commence le cours ?
- Relisez chaque fois vos notes avant de venir au cours !
- Il pleuvait.
- $x + 2 = 17$

La troisième phrase ci-dessus n'est pas une proposition parce que sa valeur de vérité dépend du moment auquel on fait référence. Par contre, « Hier, à 8h, il pleuvait sur Namur. » est bel et bien une proposition. De même la quatrième phrase n'est pas une proposition parce que sa valeur de vérité dépend de x qui, a priori, n'est pas connu. Par contre, « $\pi + 2 = 17$ » est bien une proposition parce que la valeur de π est déterminée.

Note. Au lieu de « proposition », on emploie parfois le mot « ***assertion*** ». On peut y voir une légère nuance : une assertion est une proposition qui est présentée comme vraie (mais qui peut toutefois être fausse). Certains langages informatiques possèdent une instruction appelée `assert`, qui permet de formuler une condition qui devrait être vraie si le programme fonctionne correctement. Par exemple, la ligne `assert (x >= 0)` indique que, si tout est correct, la variable x devrait être positive à ce moment-là de l'exécution. Si ce n'est pas le cas, le compilateur affiche un message d'erreur afin d'aider le programmeur à dénicher les éventuels bugs.

NOTATIONS

Quand on manipule des propositions, il peut être pratique de leur donner un nom. On utilise généralement des lettres minuscules telles que $p, q, r, \ldots$ Dans le cadre de ce syllabus, on utilisera généralement **V** et **F** pour dénoter la valeur de vérité (respectivement vrai et faux) d'une proposition.

2 Négation

Dans la suite de ce module, on va aborder diverses « opérations » permettant d'utiliser des propositions existantes pour en créer de nouvelles. La plus simple de ces opérations est la négation.

Si p est une proposition, sa ***négation***, qu'on note $\neg p$ (ou parfois $\bar{p}$), est la proposition signifiant « Il n'est pas vrai que p. » Les symboles $\neg p$ se lisent souvent « non p »

Par exemple, si p et q sont les propositions

$p \equiv$ Ce PC possède 8 GB de mémoire.

$q \equiv$ Je consulte mes mails au moins 10 fois par jour.

alors leurs négations sont

$\neg p \equiv$ Il n'est pas vrai que ce PC possède 8 GB de mémoire.

$\neg q \equiv$ Il n'est pas vrai que je consulte mes mails au moins 10 fois par jour.

ce qu'on peut réécrire en

$\neg p \equiv$ Ce PC ne possède pas 8 GB de mémoire.

$\neg q \equiv$ Je consulte mes mails moins de 10 fois par jour.

TABLE DE VÉRITÉ

On peut déduire la valeur de vérité de la négation $\neg p$ à partir de la valeur de vérité de p en utilisant le tableau suivant. Tout naturellement, $\neg p$ est vrai quand p est faux et inversement. Ce type de tableau est ce qu'on appelle la ***table de vérité*** de la proposition $\neg p$.

Négation	
p	$\neg p$
F	**V**
V	**F**

EXERCICE

1) Niez chacune des propositions suivantes.
   a) Geralt de Rivia est un chasseur de monstres.
   b) La Meuse n'est pas polluée.
   c) $2 \times 2 = 4$
   d) Nathan Drake a découvert plus de trésors qu'Indiana Jones.
   e) Une pochette contient 2 barrettes de mémoire.
   f) L'équipage du commandant Shépard comporte au moins 7 membres.
   g) Un entier est codé sur 8 bytes ou moins.
   h) Je viendrai avec des saucisses ou des brochettes.
   i) Dans cette classe, on peut trouver deux étudiants nés à la même date.
   j) Tous les schtroumpfs sont bleus.
   k) J'ai effacé le mail et le tweet qu'il m'a envoyés.

3 Propositions composées et valuations

La négation est une « opération » qui se pratique sur une proposition. Il existe d'autres « opérations » permettant de combiner plusieurs propositions (comme « et » et « ou »). Dans ce cas-là, on obtient au final une ***proposition composée*** (ou encore ***formule propositionnelle***) dans laquelle figurent plusieurs propositions simples p, q, …

Si une proposition n'est pas composée, on dit qu'il s'agit d'une ***proposition simple*** ou encore ***élémentaire***.

TABLES DE VÉRITÉ POUR DES PROPOSITIONS COMPOSÉES

La table de vérité de la négation (voir ci-dessus) ne comporte que deux lignes car il n'existe que deux cas possibles : soit p est vraie, soit p est fausse. Dans le cas d'une proposition composée utilisant plusieurs propositions simples, le nombre de lignes sera plus grand.

Par exemple, si la proposition composée utilise deux propositions simples p et q, la table de vérité comportera 4 lignes.

Table de vérité		
p	q	…
V	**V**	…
V	**F**	…
F	**V**	…
F	**F**	…

VALUATIONS

Chacune de ces lignes correspond à une valuation. Une ***valuation*** consiste à attribuer une valeur de vérité à chacune des propositions simples. Par exemple, la valuation correspondant à la 3^{e} ligne de cette table de vérité associe la valeur **V** à p et **F** à q.

EXERCICE

2) Combien existe-t-il de valuations possibles pour une proposition qui comporte…
   a) 3 variables (ou propositions élémentaires) p, q et r ?
   b) 4 variables (ou propositions élémentaires) p, q, r et s ?
   c) n variables (avec $n \geq 1$) ?

4 Conjonction

Une première option pour combiner deux conditions consiste à demander qu'elles soient toutes les deux vérifiées, ce qui se traduit généralement en français par « et ». L'opération correspondante en mathématique s'appelle la ***conjonction***.

Si p et q sont deux propositions, leur ***conjonction***, qu'on note $p \wedge q$ (ou parfois $p \cdot q$ ou simplement pq), est la proposition signifiant « p et q. » Les symboles $p \wedge q$ se lisent souvent « p et q ».

Par exemple, si p et q sont les propositions

$p \equiv$ Ce PC possède 8 GB de mémoire.
$q \equiv$ Ce PC est un portable.

alors leur conjonction est

$p \wedge q \equiv$ Ce PC possède 8 GB de mémoire et est un portable.

Cette proposition composée n'est vraie que si le PC en question remplit les deux conditions, c'est-à-dire s'il s'agit d'un portable possédant 8 GB de mémoire.

TABLE DE VÉRITÉ

La table de vérité de la conjonction se construit assez facilement à partir de son interprétation : la seule valuation pour laquelle la conjonction est vraie est celle qui rend les deux propositions élémentaires vraies.

Conjonction		
p	q	$p \wedge q$
F	**F**	**F**
F	**V**	**F**
V	**F**	**F**
V	**V**	**V**

5 Disjonction

Les symboles qui permettent de combiner deux propositions (comme $\wedge$ par exemple) sont souvent appelés des ***connecteurs logiques***. La conjonction est loin d'être le seul connecteur logique : à la place d'un « et », qui impose que les deux conditions soient vérifiées, on peut utiliser un « ou », pour lequel il suffit qu'une des deux conditions soient vérifiées.

Si p et q sont deux propositions, leur ***disjonction***, qu'on note $p \vee q$ (ou parfois $p + q$), est la proposition signifiant « p ou q. » Les symboles $p \vee q$ se lisent souvent « p ou q ».

Par exemple, si p et q sont les propositions

$p \equiv$ Ce PC possède 8 GB de mémoire.
$q \equiv$ Ce PC est un portable.

alors leur disjonction est

$p \vee q \equiv$ Ce PC possède 8 GB de mémoire ou est un portable.

Cette proposition composée est vraie dès que le PC est un portable ou qu'il possède 8 GB de mémoire. Trois situations la rendent donc vraie : (1) si c'est un portable avec 8 GB de mémoire, (2) si c'est un portable qui n'a pas 8 GB de mémoire, (3) si c'est PC qui n'est pas un portable mais qui a 8 GB de mémoire. Le seul cas où la proposition composée est fausse est celui où le PC n'est pas un portable et qu'il ne possède pas non plus 8 GB de mémoire (c'est-à-dire quand les propositions simples sont toutes les deux fausses).

TABLE DE VÉRITÉ

La valeur de vérité d'une proposition construite à partir de ce connecteur logique se déduit de la table de vérité suivante. La seule valuation qui rend une disjonction fausse est celle pour laquelle les deux propositions élémentaires sont fausses.

Disjonction		
p	q	$p \vee q$
F	**F**	**F**
F	**V**	**V**
V	**F**	**V**
V	**V**	**V**

POUR NE PAS CONFONDRE LE « ET » ET LE « OU »

Les symboles pour ces deux connecteurs sont fort proches : ∧ pour le « et » et ∨ pour le « ou ». Une astuce mnémotechnique pour retenir cette association : dans le mot « ou », on trouve un « u », dont la forme ressemble à « ∨ ».

6 Disjonction exclusive

Le mot « ou » en français peut cacher deux significations légèrement différentes. Considérez les deux exemples suivants.

(1) À la fin du repas, on vous servira du fromage ou un dessert.
(2) Tu auras ton diplôme si tu réussis tes examens ou si tu me verses 50 000 €.

La phrase (1) indique qu'à la fin du repas, vous pourrez demander à avoir du fromage ou du dessert. Toutefois, vous ne pourrez sans doute pas recevoir à la fois du fromage <u>et</u> du dessert : il s'agit d'un choix à faire, un choix dit « exclusif » (parce que choisir une des options exclut l'autre).

À l'inverse, dans la phrase (2), si la personne concernée verse 50 000 € <u>et</u> réussit ses examens, elle recevra tout de même son diplôme. Il ne s'agit pas ici d'un choix exclusif : les deux options pourraient être vraies.

Le cas (2) correspond à la disjonction introduite dans la section précédente. Le cas (1), par contre, correspond à un autre connecteur appelé la ***disjonction exclusive***. Celle-ci signifie qu'une des deux propositions doit être vraie et que l'autre doit être fausse.

Si p et q sont deux propositions, leur ***disjonction exclusive***, qu'on note $p \dot{\vee} q$ (ou parfois $p \oplus q$), est la proposition signifiant « p ou q mais pas les deux. » Les symboles $p \dot{\vee} q$ se lisent souvent « p ou exclusif q ».

TABLE DE VÉRITÉ

La table de vérité de la disjonction exclusive ressemble à celle de la disjonction, avec une seule différence dans le cas où les deux conditions sont vraies en même temps. On remarque que la disjonction exclusive est vraie pour toutes les valuations qui associent des valeurs de vérité <u>différentes</u> aux deux propositions élémentaires et fausse pour les valuations qui associent la même valeur de vérité aux deux proposition élémentaires.

Disjonction exclusive		
p	q	$p \dot{\vee} q$
F	**F**	**F**
F	**V**	**V**
V	**F**	**V**
V	**V**	**F**

EXERCICES

3) On considère les propositions $p \equiv$ « Roméo aime Juliette » et $q \equiv$ « Juliette aime Roméo ». Traduisez par une phrase chacune des propositions suivantes.
   a) $\neg p$
   b) $p \vee q$
   c) $p \wedge q$
   d) $\neg(p \wedge q)$
   e) $(\neg p) \wedge (\neg q)$

4) Modélisez ces phrases en utilisant $p \equiv$ L'eau est chaude » et $q \equiv$ « L'eau est bleue ».
   a) L'eau est chaude et bleue.
   b) L'eau n'est ni chaude ni bleue.
   c) Il est faux que l'eau soit chaude ou bleue.
   d) L'eau est chaude ou froide et bleue.

5) Dans les phrases suivantes, précisez s'il s'agit d'une disjonction ou d'une disjonction exclusive.
   a) Si tu as plus de 18 ans ou que tu es accompagné d'un adulte, tu peux entrer.
   b) Ce ticket vous permet d'assister à la séance du matin ou à celle de l'après-midi.
   c) Tu prendras un café ou un thé après ton repas ?
   d) Le candidat à ce poste devra connaître le C ou le Java.
   e) Le prix du repas couvre également la soupe ou l'entrée chaude.
   f) La bourse ou la vie !
   g) Votre mot de passe doit contenir au moins 3 chiffres ou comporter au moins 8 caractères.
   h) À l'achat de 10 bouteilles, le magasin vous offre un bon de 5€ ou un tire-bouchon.
   i) Ce site de vente en ligne vous permet de payer en Euros ou en dollars.
   j) Pour 2€, tu peux avoir quatre sucettes ou une crème glacée.

7 Implication

« S'il pleut demain matin, j'emmènerai mon parapluie. » Cette phrase est construite à partir de deux propositions simples.

$p \equiv$ il pleuvra demain matin
$q \equiv$ j'emmènerai mon parapluie

La phrase en question signifie que, si p est vraie, alors q est vraie : si, demain matin, je vois qu'il pleut, alors, je prendrai mon parapluie. On peut noter dès à présent que la phrase n'impose rien du tout dans le cas où p serait faux. Ainsi, s'il ne pleut pas, rien ne m'empêche de prendre tout de même mon parapluie !

À partir de deux propositions p et q, on peut construire l'***implication*** $p \Rightarrow q$, c'est-à-dire la proposition signifiant « si p alors q. » La proposition p est la ***condition*** (ou la ***prémisse*** ou l'***hypothèse*** ou l'**antécédent**) de l'implication ; la proposition q, elle, est la ***conclusion*** (ou la ***conséquence*** ou la ***thèse***) de l'implication.

Les symboles $p \Rightarrow q$ peuvent se lire de nombreuses manières ; entre autres :

- si p alors q
- p implique q
- p est une condition suffisante pour q
- q est une condition nécessaire pour p
- chaque fois que p, on a q
- de p, on peut déduire q

CONDITIONS NÉCESSAIRES ET CONDITIONS SUFFISANTES

Reprenons la phrase « S'il pleut demain matin, j'emmènerai mon parapluie. »

Il pourrait y avoir toute une série de raisons pour lesquelles je choisis d'emmener mon parapluie : parce qu'il est déchiré et a besoin d'être réparé, pour l'amener à un ami qui désire l'emprunter, parce qu'il s'agit d'un nouvel achat et que j'ai envie de frimer… Mais, même si ces conditions sont fausses, voir qu'il pleut demain matin suffi à me décider à emmener mon parapluie. On dit que la condition « il pleuvra demain matin » est une ***condition suffisante*** pour que j'emmène mon parapluie.

Dans $p \Rightarrow q$, on dit que p est une condition suffisante pour q.

Si demain, vous me voyez arriver chez vous sans parapluie, même sans regarder par la fenêtre, vous pouvez en déduire qu'il ne pleut pas. C'est seulement si vous constatez que j'ai emmené mon parapluie que vous pouvez vous dire « Ah, tiens, il pleut peut-être. » Le fait que j'ai emmené mon parapluie avec moi est une ***condition nécessaire*** pour qu'il soit envisageable qu'il pleuve.

Dans $p \Rightarrow q$, on dit que q est une condition nécessaire pour p.

TABLE DE VÉRITÉ

La table de vérité de l'implication se construit en notant que le seul cas dans lequel l'implication $p \Rightarrow q$ est fausse (ou mise en défaut) est celui où la condition p est vraie mais la conséquence q ne l'est pas.

Implication		
p	q	$p \Rightarrow q$
F	**F**	**V**
F	**V**	**V**

V	**F**	**F**
V	**V**	**V**

On peut observer que, dans tous les cas où la condition p est fausse, l'implication $p \Rightarrow q$ est vraie. Ce choix s'explique par le fait que, si la condition n'est pas remplie, on ne peut rien dire à propos de l'implication : on ne peut pas la mettre en défaut et, par conséquence, on la suppose vraie.

Pour comprendre ce choix, vous pouvez vous placer dans la peau d'un scientifique qui doit vérifier l'implication « Si tu es un étudiant en DA, alors, tu es intelligent. » Si le scientifique en question ne rencontre <u>que</u> des étudiants des autres sections, il ne peut rien dire au sujet de l'implication. En cas de doute, il la considérera vraie. Après tout, « tous les étudiants d'IA qu'il a rencontrés » étaient bel et bien intelligents !

EXERCICES

6) Considérons la phrase « Pour programmer correctement, tu dois respecter les règles du Clean Code. ».

   - Quelle sont les propositions élémentaires utilisées ?
   - Par quelle implication peut-on traduire cette phrase ?
   - Quelle est la condition suffisante ?
   - Quelle est la condition nécessaire ?

7) Repérez dans les phrases suivantes les antécédents (ou conditions) et les conséquents (ou conséquences).
   a) Quand elle parle, tout le monde écoute.
   b) Tout le monde écoute si elle parle.
   c) Elle ne parle que si tout le monde écoute.
   d) Un entier est positif si son cube est positif.
   e) Je ne viendrai que si tu y vas.
   f) Un entier n'est positif que si son cube est positif.

8) Déterminez si chacune des implications suivantes est vraie ou fausse.
   a) Si la température descend sous zéro, alors l'eau gèle.
   b) Si les poules ont des dents, alors il pleut.
   c) Chaque fois qu'il a plu en 2014, Bruxelles était la capitale de la Belgique.
   d) Si 1 + 1 = 2, alors 2 + 2 = 5.
   e) Si 1 + 1 = 3, alors 2 + 2 = 4.
   f) Si 2 + 2 = 5, alors 1 + 2 = 6.
   g) 2 est pair implique que π vaut à peu près 3,14.
   h) 3 est pair implique que π vaut à peu près 3,14.

9) Vous êtes surveillant dans une salle équipée de PC accessibles à tous. Une règle est en vigueur dans cette salle : seuls les étudiants peuvent utiliser les PC pour accéder à internet (les autres ne pouvant travailler qu'en local).
   a) Modélisez la règle citée sous la forme d'une implication qui reflète ce que vous devez vérifier en tant que surveillant chargé de l'application de cette règle.
   b) Quatre personnes se trouvent dans la salle. Vous savez que la 1^{re} est un étudiant et que la 2^{e} n'en est pas un. Vous savez que la 3^{e} personne utilise actuellement un navigateur pour accéder à internet et que la 4^{e} personne travaille en local. Parmi ces 4 personnes, quelles

sont celles que vous devez interroger / surveiller de plus près pour faire appliquer la règle citée ci-dessus ?

c) Établissez un lien entre la réponse au point b et la table de vérité de l'implication du point a.

10) [Énigme tirée du journal Le Soir] Quatre cartes sont déposées sur une table, chacune comportant un chiffre sur une face et une lettre sur l'autre. Les faces visibles indiquent « E », « K », « 4 » et « 7 ». Quelle(s) carte(s) devez-vous retourner au minimum afin de pouvoir conclure que, si une carte a un « E » sur une face, alors, elle présente un « 4 » sur l'autre face ?

8 Inverse, converse et contraposée

À partir des propositions d'une implication $p \Rightarrow q$, on peut construire plusieurs autres implications. En voici trois, la converse, l'inverse et la contraposée, qui seront présentées à l'aide de l'exemple suivant.

$p \equiv$ Tu programmes correctement.

$q \equiv$ Tu respectes le Clean Code.

(Implication de départ)
$p \Rightarrow q \equiv$ Si tu programmes correctement, tu respectes le Clean Code.

L'***inverse*** est obtenue en niant la condition et la conclusion : $\neg p \Rightarrow \neg q$.

(Inverse)
$\neg p \Rightarrow \neg q \equiv$ Si tu ne programmes pas correctement, tu ne respectes pas le Clean Code.

On peut noter que l'inverse n'a pas le même sens que l'implication initiale. Ainsi, l'implication de l'exemple indique que, pour programmer correctement, il faut respecter le Clean Code (ce qui, a priori, est vrai). L'inverse, quant à elle, dit que, si on ne programme pas correctement, c'est qu'on ne respecte pas le Clean Code… mais ce n'est pas forcément le cas : on pourrait respecter le Clean Code et programmer comme un pied en écrivant des algorithmes incorrects par exemple. L'inverse a donc une signification différente de l'implication de départ.

L'inverse n'est pas non plus la négation de l'implication originale. On peut le constater en remarquant que, dans le cas où p et q sont tous les deux faux, l'implication de départ est vraie (**F** ⇒ **F**) et l'inverse est également vraie (**V** ⇒ **V**).

La ***converse*** s'obtient en permutant la condition et la conséquence : $q \Rightarrow p$.

(Converse)
$q \Rightarrow p \equiv$ Si tu respectes le Clean Code, tu programmes correctement.

Ici encore, on obtient une nouvelle proposition composée qui n'a pas le même sens que l'implication de départ. En effet, l'implication de départ indiquait que, pour programmer correctement, il fallait respecter le Clean Code et la converse dit qu'il <u>suffit</u> de respecter le Clean Code pour programmer correctement.

La différence entre l'implication de départ et la converse est du même ordre que celle qui sépare les notions de condition suffisante et de condition nécessaire.

Finalement, la ***contraposée*** est l'implication obtenue en niant et en permutant la condition et la conséquence : $\neg q \Rightarrow \neg p$.

> (Contraposée)
> $\neg q \Rightarrow \neg p \equiv$ Si tu ne respectes pas le Clean Code, tu ne programmes pas correctement.

Contrairement aux deux autres propositions considérées ci-dessus, la contraposée, elle, a le même sens que l'implication de départ. L'implication disait que, pour programmer correctement, il fallait respecter le Clean Code ; la contraposée dit que, si on ne respecte pas le Clean Code, on ne programme pas correctement. Il s'agit bien là de deux manières différentes de dire la même chose.

9 Double-implication (ou équivalence)

Si on parle des polygones (les figures géométriques à plusieurs côtés), on peut dire que « Si la figure a 3 côtés, c'est un triangle. » La converse est également vraie : « Si la figure est un triangle, elle a 3 côtés. » Dans ce cas-ci, les deux propositions de base

> $p \equiv$ La figure a 3 côtés.
> $q \equiv$ La figure est un triangle.

sont en quelque sorte « équivalentes » ; elles reviennent au même. Cela correspond à la notion de définition : pour éviter de répéter « une figure à 3 côtés », on introduit le mot « triangle ». Ces deux propositions sont donc soit toutes les deux vraies en même temps soit toutes les deux fausses.

On pourrait traduire le lien entre p et q par deux implications $p \Rightarrow q$ et $q \Rightarrow p$ mais il existe un connecteur spécifique permettant d'écrire tout cela en une seule formule.

À partir de deux propositions p et q, on peut construire l'***équivalence*** ou la ***double-implication*** $p \Leftrightarrow q$, c'est-à-dire la proposition signifiant « p si et seulement si q » (ce qu'on abrège souvent en « p ssi q »).

Les symboles $p \Leftrightarrow q$ peuvent se lire de nombreuses manières ; entre autres :
• p si et seulement si q
• si p alors q et inversement
• p est une condition nécessaire et suffisante pour q

CONDITIONS NÉCESSAIRES ET SUFFISANTES

Si $p \Leftrightarrow q$ est vraie, cela signifie que p implique q (c'est-à-dire que p est une condition suffisante pour q) et que q implique p (c'est-à-dire que p est une condition nécessaire pour q). On peut donc dire qu'au final, p est une condition nécessaire et suffisante pour q (on parle parfois aussi de ***caractérisation*** de q). Concrètement, cela signifie que p et q sont des

« synonymes », des propositions équivalentes et dire que l'une est vraie (ou fausse) revient à dire que l'autre est vraie (ou fausse).

TABLE DE VÉRITÉ

La table de vérité de l'équivalence se construit facilement si on note que les deux propositions doivent avoir la même valeur de vérité.

Équivalence		
p	q	$p \Leftrightarrow q$
F	**F**	**V**
F	**V**	**F**
V	**F**	**F**
V	**V**	**V**

ENCORE UNE IMPRÉCISION DU FRANÇAIS

Dans les langages courants (qu'on appelle langages informels), la double-implication est souvent implicite. Par exemple, quand un parent dit à son enfant « Si tu finis ton repas, tu pourras avoir ton dessert », il indique non seulement une implication mais bel et bien une équivalence. Ainsi, si l'enfant termine son repas, il aura son dessert (c'est l'implication)… et, s'il veut recevoir un dessert, il doit terminer son repas (c'est la converse, l'implication dans l'autre sens). Une formule plus précise (mais un peu lourde) serait « Tu auras ton dessert si et seulement si tu finis ton repas. »

EXERCICES

11) Modélisez les phrases suivantes sous forme d'implications entre les assertions données.
   a) Si tu dépasses le 120 km/h, tu auras une contravention.
      $p \equiv$ Tu dépasses le 120 km/h
      $q \equiv$ Tu as une contravention
   b) Je viens au cours chaque fois qu'il y a une interrogation.
      $p \equiv$ Je viens au cours
      $q \equiv$ Il y a une interrogation
   c) Je ne viens au cours que s'il y a une interrogation.
      $p \equiv$ Je viens au cours
      $q \equiv$ Il y a une interrogation
   d) Elle a le mal de mer chaque fois qu'elle monte à bord d'un bateau.
      $p \equiv$ Elle a le mal de mer
      $q \equiv$ Elle monte à bord d'un bateau
   e) Pour pouvoir faire jouer la garantie, il suffit que l'achat du PC date de moins d'un an.
      $p \equiv$ On peut faire jouer la garantie
      $q \equiv$ L'achat date de moins d'un an
   f) Il est nécessaire de soudoyer le professeur pour réussir ce cours.
      $p \equiv$ Tu soudoies le professeur
      $q \equiv$ Tu réussis ce cours
   g) La garantie n'est valable que si le CD est encore emballé.
      $p \equiv$ La garantie est valable
      $q \equiv$ Le CD est encore emballé

12) Ecrivez les propositions suivantes sous la forme d'une implication puis construisez leur inverse, leur converse et leur contraposée.
    a) S'il pleut, je prendrai mon parapluie.
    b) Si le train n'a pas de retard, j'arriverai à l'heure prévue.
    c) L'appli ne se lance pas si le champ *nomUtilisateur* n'est pas complété.

10 Formules bien formées

On peut combiner les éléments présentés dans les sections précédentes pour construire des propositions de plus en plus complexes. Cependant, n'importe quelle suite de symboles ne convient pas. Ainsi, l'écriture « $p \vee\Rightarrow q\neg$ » n'a par exemple pas de sens. Alors, qu'est-ce qu'une écriture correcte, qu'est-ce qu'une ***formule bien formée*** ?

La réponse mathématique à cette question prend la forme de ce qu'on appelle une ***définition inductive***. C'est une notion sur laquelle on reviendra plus tard mais, en première approche, on peut se contenter de dire qu'on donne différentes règles qui peuvent être utilisées plusieurs fois pour former des éléments qui seront considérés comme corrects. Parmi ces règles, on distingue les ***règles de base*** qui présentent les propositions élémentaires (construites directement) et les ***règles inductives*** qui, elles, indiquent comment combiner des propositions pour en former de nouvelles, plus complexes.

Règle de base. Quelle que soit la proposition élémentaire p, l'écriture « p » est une formule bien formée. (Autrement dit, une proposition élémentaire constitue déjà, à elle seule, une formule bien formée.)

Règle inductive 1. Si « α » est une formule bien formée, alors « $\neg\alpha$ » en est également une. (Autrement dit, on peut ajouter un symbole $\neg$ devant une formule bien formée.)

Règle inductive 2. Si « α » et « β » sont deux formules bien formées, alors toutes les écritures suivantes en sont également : « $(\alpha \vee \beta)$ », « $(\alpha \wedge \beta)$ », « $(\alpha \Rightarrow \beta)$ », « $(\alpha \Leftrightarrow \beta)$ », « $(\alpha \dot{\vee} \beta)$ ». (Autrement dit, on peut combiner deux formules bien formées en utilisant l'un des connecteurs logiques présentés plus haut.)

Seules les écritures qui peuvent être construites à partir de ces règles (en supprimant éventuellement des parenthèses inutiles) sont des formules bien formées.

Ainsi, à partir des propositions élémentaires p, q et r, les règles présentées ci-dessus permettent par exemple de construire (a) la négation $\neg q$ puis (b) les conjonctions $p \wedge q$ et $\neg q \wedge r$ puis ensuite de (c) les combiner en utilisant une disjonction afin d'obtenir $(p \wedge q) \vee (\neg q \wedge r)$. On pourrait encore reprendre cette dernière proposition et la composer avec une autre pour obtenir un résultat plus long, et ainsi de suite.

DES DÉFINITIONS INDUCTIVES EN INFORMATIQUE ?

On en trouve quasiment partout dans la définition des langages informatiques, et plus particulièrement de leur syntaxe. C'est par exemple comme ça qu'on définit précisément ce qu'est une instruction valable ou encore une expression ou une fonction ou une procédure. Ces définitions inductives sont utilisées à la fois par les programmeurs, pour savoir comment écrire un programme et par les compilateurs, pour savoir comment décomposer ce que les programmeurs ont écrits.

11 Tables de vérité, tautologies et inepties

La définition inductive présentée ci-dessus indique comment il est possible d'obtenir des formules logiques en les composant partie par partie, morceau par morceau. À l'inverse, on peut également utiliser cette définition pour savoir comment une formule logique a été composée.

Considérons par exemple la formule $(p \vee \neg q) \Rightarrow (p \wedge q)$. Elle a été créée en combinant via le connecteur ⇒ les deux composantes $p \vee \neg q$ et $p \wedge q$. Le premier de ces composants est lui-même composé : il a été créé en utilisant le connecteur ∨ sur les composantes p et $\neg q$. La composante p est élémentaire… on ne peut pas la décomposer plus. Par contre, la composante $\neg q$ provient de l'utilisation de ¬ sur q. En procédant ainsi pour toutes les parties de la formule de départ, on se ramène finalement aux propositions élémentaires p et q, qui sont les deux « variables » qui y apparaissent.

Pour déterminer la valeur de vérité d'une telle formule logique (en fonction des diverses valuations possibles), on peut dresser sa table de vérité. Pour ce faire, on procède comme suit.

1. On recherche les propositions élémentaires qui ont été utilisées et on construit une colonne pour chacune d'elles.
2. On détermine toutes les valuations possibles sur ces propositions élémentaires.
3. On ajoute une colonne pour chacune des composantes de la formule, en commençant par les composantes les plus simples et en terminant par la formule entière.
4. On complète ces colonnes en indiquant les valeurs de vérité de chacune des composantes.

Voici par exemple une table de vérité pour la proposition $(p \vee \neg q) \Rightarrow (p \wedge q)$.

Table de vérité					
p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \Rightarrow (p \wedge q)$
F	**F**	**V**	**V**	**F**	**F**
F	**V**	**F**	**F**	**F**	**V**
V	**F**	**V**	**V**	**F**	**F**
V	**V**	**F**	**V**	**V**	**V**
(1) déterminer les valuations à considérer		*(2) calculer petit à petit les valeurs de vérité des composantes*			*(3) en déduire les valeurs de vérité pour la proposition entière*

TAUTOLOGIE, INEPTIE ET PROPOSITIONS SATISFAISABLES

Certaines propositions ont une table de vérité un peu particulière. Une proposition dont la valeur de vérité est toujours **V** (quelle que soit la valuation) est une ***tautologie***. Une proposition dont la valeur de vérité est **V** pour au moins une valuation est une ***proposition satisfaisable*** (on peut la rendre vraie / la satisfaire en faisant les bons choix). Une proposition dont la valeur de vérité est toujours **F** (quelle que soit la valuation) est une ***proposition insatisfaisable*** ou encore une ***ineptie*** ou une ***contradiction***.

Si une proposition α est une tautologie, on l'indique parfois par la notation $\vDash \alpha$.

PROPOSITIONS ÉQUIVALENTES

Si deux propositions (simples ou composées) ont la même valeur de vérité sous chacune des valuations possibles, on dit qu'elles sont ***équivalentes***. Concrètement, sur la table de vérité, deux propositions équivalentes correspondent à des colonnes identiques (**V** et **F** aux mêmes endroits).

Cela devient intéressant de trouver des propositions équivalentes à partir du moment où l'une des deux est plus simple que l'autre. Par exemple, en observant la table de vérité ci-dessus, on remarque que les colonnes correspondant aux propositions $(p \vee \neg q) \Rightarrow (p \wedge q)$ et q sont identiques. Cela signifie que, plutôt que de réaliser des calculs en utilisant la proposition complexe, on peut se contenter de regarder la valeur de q. Ces équivalences et diverses méthodes de simplification seront abordées dans la suite du cours.

12 Précédence des opérateurs logiques

Tout comme les opérateurs arithmétiques en C (et dans les autres langages de programmation), les connecteurs logiques sont classés selon un ordre de précédence qui indique lequel sera calculé en premier en l'absence de parenthèses. Cela permet de simplifier les écritures obtenues grâce aux règles inductives présentées plus haut en supprimant certaines parenthèses inutiles.

Précédence	**Opérateur**	
grande	$\neg$	négation
	$\wedge$	conjonction
	$\vee$, $\dot{\vee}$	disjonction (exclusive)
	$\Rightarrow$	implication
petite	$\Leftrightarrow$	double-implication/équivalence

Concrètement, plus la précédence d'un opérateur est grande et plus cet opérateur sera considéré tôt. Ainsi, la formule $p \vee q \Rightarrow r$ signifie $(p \vee q) \Rightarrow r$ vu que la disjonction est résolue avant l'implication. Et, si on veut nier l'implication $p \Rightarrow q$, il ne suffit pas d'écrire $\neg p \Rightarrow q$, qui signifie $(\neg p) \Rightarrow q$, mais il faut obligatoirement ajouter des parenthèses : $\neg(p \Rightarrow q)$.

EXERCICES (LECTURE ET TRADUCTIONS)

13) Sachant que a et b sont des propositions vraies et que c et d sont fausses, donnez la valeur de vérité des propositions composées suivantes.
    a) $a \wedge (\neg b \vee c)$
    b) $\neg(\neg a \vee d) \Rightarrow \neg b$
    c) $\neg\big(\neg d \wedge (b \Rightarrow \neg a)\big)$
    d) $(c \Rightarrow a) \Rightarrow d$

14) En supposant que p et q sont les propositions décrites ci-dessous, traduisez en français les formules mathématiques suivantes.

    $p \equiv$ on peut nager dans ce fleuve
    $q \equiv$ ce fleuve est infesté de piranhas

    a) $\neg p$
    b) $p \vee q$
    c) $\neg p \wedge q$
    d) $\neg p \vee (p \wedge q)$
    e) $p \Rightarrow \neg q$
    f) $\neg q \Rightarrow p$
    g) $\neg p \Leftrightarrow q$

15) En supposant que p et q sont les propositions décrites ci-dessous, modélisez les phrases suivantes sous la forme de formules logiques.

    $p \equiv$ il gèle
    $q \equiv$ il neige

    a) Il gèle et il neige.
    b) Il gèle mais il ne neige pas.
    c) Il ne gèle pas et il ne neige pas non plus.
    d) Il gèle ou il neige (ou les deux ensemble).
    e) Il gèle ou il neige (mais pas les deux ensemble).
    f) S'il gèle, alors il neige également.
    g) S'il ne gèle pas, alors, il ne neige pas non plus.
    h) Il ne neige pas, à moins qu'il ne gèle.
    i) Il est nécessaire qu'il gèle pour qu'il neige.

16) En supposant que p, q et r sont les propositions décrites ci-dessous, modélisez en propositions les phrases suivantes.

    $p \equiv$ L'utilisateur rentre un mot de passe correct.
    $q \equiv$ L'utilisateur a accès au site.
    $r \equiv$ L'utilisateur a payé son abonnement.

    a) L'utilisateur a payé son abonnement mais n'a pas rentré un mot de passe correct.
    b) L'utilisateur a accès au site à partir du moment où il a payé son abonnement et il a rentré un mot de passe correct.
    c) L'utilisateur qui n'a pas payé son abonnement n'a pas accès au site.
    d) Si l'utilisateur ne rentre pas un mot de passe valide mais qu'il a payé son abonnement, il a accès au site.
    e) Pour avoir accès au site, il est suffisant d'avoir payé son abonnement.
    f) Pour avoir accès au site, il est nécessaire de rentrer un mot de passe correct.
    g) Un utilisateur qui a payé son abonnement a accès au site si et seulement s'il rentre un mot de passe correct.

17) Dans un programme, on stocke dans la variable `nombreLu` un nombre (entier) rentré par l'utilisateur. Exprimez les conditions suivantes en utilisant le langage mathématique. Si nécessaire, vous pouvez utiliser les propositions élémentaires ci-dessous ainsi que celles que vous définirez en résolvant cet exercice.

    $p \equiv$ `nombreLu` $\in 2\mathbb{N} \equiv$ `nombreLu` est pair
    $q \equiv$ `nombreLu` $\in 3\mathbb{N} \equiv$ `nombreLu` est divisible par 3

    a) $r \equiv$ Le nombre doit être supérieur ou égal à 10. (proposition élémentaire)
    b) $s \equiv$ Le nombre doit être (strictement) supérieur à 20. (proposition élémentaire)
    c) Le nombre doit être supérieur ou égal à 10 et impair.
    d) Le nombre doit vérifier au moins l'une des deux conditions suivantes : être un multiple de 3, être supérieur à 20.

e) Le nombre doit vérifier les deux conditions suivantes : être pair, être supérieur ou égal à 10.
f) Le nombre doit vérifier une seule des conditions suivantes : être pair, être multiple de 3.
g) Si le nombre est pair, il doit être (strictement) inférieur à 10.
h) Le nombre doit être compris (au sens large) entre 10 et 20.
i) Si le nombre n'est divisible ni par 2 ni par 3, alors il doit être (strictement) supérieur à 20.
j) Le nombre doit être pair et divisible par 3, ou bien impair et non divisible par 3.
k) Si le nombre est inférieur à 10, il doit être pair. Sinon, il doit être divisible par 3.

EXERCICE (TABLES DE VÉRITÉ)

18) Construisez les tables de vérité des formules suivantes. Repérez les utopies et les tautologies.
   a) $p \vee (q \wedge \neg p)$
   b) $p \wedge (p \Rightarrow q) \wedge \neg q$
   c) $(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$
   d) $(p \Rightarrow q) \wedge (\neg p \Rightarrow q)$
   e) [optionnel] $[p \vee (\neg q)] \Rightarrow [(p \vee r) \Leftrightarrow (q \vee r)]$

13 Équivalences logiques

Pour déterminer quand une proposition complexe est vraie, on peut construire sa table de vérité. Cependant, si la proposition comporte de très nombreuses composantes, cette méthode peut s'avérer assez longue… Une autre approche consiste à analyser la proposition de départ et à la transformer en une autre proposition, plus courte, mais possédant les mêmes valeurs de vérité.

Si p et q sont des propositions logiques qui possèdent la même valeur de vérité pour chaque valuation possible, alors, on dit que p et q sont des ***propositions équivalentes***, ce qu'on note $p \equiv q$. Cela revient à dire que les deux propositions ont la même table de vérité.

Cette nouvelle notion permet de revenir sur les concepts de tautologies et d'utopies. Une tautologie est une proposition p qui est toujours vraie, c'est-à-dire une proposition p telle que $p \equiv \mathbf{V}$. Et une utopie est une proposition q qui est toujours fausse, c'est-à-dire une proposition q telle que $q \equiv \mathbf{F}$.

On peut aussi remarquer que deux propositions p et q sont équivalentes si et seulement si la proposition $p \Leftrightarrow q$ est une tautologie.

Pour transformer une proposition p en une proposition équivalente plus simple, on peut appliquer diverses règles de réécriture. Les sections qui suivent présentent quelques-unes de ces règles.

14 Premières équivalences

Certaines équivalences se déduisent directement de la définition des opérateurs logiques.

DOUBLE NÉGATION

Nier une proposition deux fois (donc nier la négation) revient à garder la proposition de départ. Mathématiquement, cela se note $\neg(\neg p) \equiv p$: plutôt que d'étudier $\neg(\neg p)$, on peut analyser tout simplement p et obtenir les mêmes valeurs de vérité au final. Et cela reste vrai que p soit une proposition élémentaire ou une proposition complexe avec de nombreuses composantes.

COMMUTATIVITÉ

La plupart des opérateurs logiques sont commutatifs, ce qui signifie qu'on peut changer l'ordre des sous-propositions sans modifier le résultat final. On a donc par exemple $p \vee q \equiv q \vee p$ et $p \wedge q \equiv q \wedge p$. Attention… ce n'est cependant pas le cas pour l'implication, car $p \Rightarrow q$ et $q \Rightarrow p$ ne sont pas équivalents !

RÉPÉTITION (EN TERMES TECHNIQUES : IDEMPOTENCE)

Quand une formule combine deux fois la même proposition, il y a généralement moyen de la simplifier. Par exemple, la proposition $p \vee p$ aura toujours la même valeur de vérité que p. On a $p \vee p \equiv p$ et $p \wedge p \equiv p$.

EXERCICES

19) Peut-on également simplifier $p \Rightarrow p$ et $p \Leftrightarrow p$ en p ?

20) Pourquoi ne peut-on pas dire que $p \Rightarrow q$ et $q \Rightarrow p$ sont des propositions équivalentes ?

21) [Écritures équivalentes à l'implication] Modélisez les phrases suivantes sous la forme de formules logiques puis dressez-en la table de vérité afin de repérer celles qui sont équivalentes. Notez ensuite les règles d'équivalence trouvées.
    a) Si tu travailles, tu réussiras.
    b) Tu travailles ou tu ne réussiras pas.
    c) Si tu réussis, c'est que tu as travaillé. [converse de a]
    d) Si tu ne travailles pas, tu ne réussiras pas. [inverse de a]
    e) Si tu ne réussis pas, (c'est que) tu n'as pas travaillé. [Contraposée de a]

22) Réécrivez les propositions suivantes en des propositions équivalentes plus simples.
    a) $p \vee$ **V**
    b) $p \vee$ **F**
    c) $p \wedge$ **V**
    d) $p \wedge$ **F**
    e) $p \wedge p$
    f) $p \vee p$
    g) $p \Rightarrow$ **V**
    h) $p \Rightarrow$ **F**
    i) $p \Rightarrow p$

15 Les lois de De Morgan

On a vu plus haut que nier la négation d'une proposition revenait à considérer la proposition de départ. Mais que se passe-t-il si on nie une conjonction ou une disjonction ?

Soit p et q les propositions suivantes.

$p \equiv$ ce programme est écrit en C.
$q \equiv$ ce programme est écrit en Java.

La disjonction $p \vee q$ signifie « Ce programme est écrit en C ou en Java. » Sa négation est donc « Ce programme n'est pas écrit en C ni en Java. », ce qui revient à dire que la programme n'est pas écrit en C et qu'il n'est pas écrit en Java. En niant un « ou », on obtient un « et ».

C'est ce que traduit la ***loi de De Morgan*** (non, ce n'est pas une erreur typographique ; l'inventeur était un certain Augustus De Morgan, avec un petit « de »).

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

L'autre loi de De Morgan permet de nier une conjonction. Pour ce faire, on nie chacune des composantes et on remplace le « et » par un « ou ».

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

On peut noter que ce procédé permettant de nier une conjonction ou une disjonction peut s'étendre aux cas où on combine plus que deux sous-propositions. Par exemple :

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

Pratiquement, pour nier une disjonction, on nie chacune des composantes et on remplace le « ou » par un « et ». Et, pour nier une conjonction, on nie chacune de ses composantes et on remplace le « et » par un « ou ». Dans les deux cas, on nie les composantes et on « inverse » le connecteur.

EXERCICES

23) Utilisez les lois de De Morgan pour nier les phrases suivantes.
    a) Bill Gates est riche et content.
    b) Demain, je viendrai à pied ou en voiture.
    c) Cet étudiant est intelligent et il travaille dur.
    d) Tu auras 19 ou 20 sur 20.
    e) Tu trouveras un job ou tu poursuivras tes études.
    f) Ce candidat connaît l'algèbre et le langage Java.

24) En utilisant les lois de De Morgan, simplifiez les conditions ci-dessous.
    g) $\neg(x > 5 \wedge x \leq 10)$
    h) $\neg(x \geq 0 \vee y^2 < 7)$

25) Écrivez la négation de ces propositions de manière aussi simple que possible.
    a) $\neg p$
    b) $p \vee q$
    c) $p \wedge q$

d) $p \dot{\vee} q$
e) $p \Rightarrow q$
f) $p \Leftrightarrow q$

26) Après l'examen, les étudiants Albert, Barnabé et Charles demandent au professeur s'ils ont réussi. Il répond: « au moins un de vous trois a réussi et si Albert a réussi, Barnabé a réussi aussi. Comme je suis certain que Albert et Charles ont collaboré, ils ont réussi ou échoué tous les deux. Enfin, Barnabé et Charles n'ont pas réussi tous les deux ensemble ».
Qui a (ont) réussi l'examen ?

27) Un directeur des RH doute de l'honnêteté de son personnel, dont les personnages a, b et c.

a. Au moins un d'entre eux a, b ou c est malhonnête,
b. Si a est malhonnête, b l'est aussi
c. Si a est malhonnête, c l'est aussi
d. b et c ne sont pas malhonnêtes tous les deux en même temps
e. Si a n'est pas malhonnête, c ne l'est pas non plus

Qui est (sont) malhonnête(s)?

Pour aller plus loin…

La logique des propositions sert de base à de nombreux domaines liés à l'informatique. C'est par exemple le cas de toute une série de langages de programmation dits « langages logiques », où on décrit certains faits ou certaines règles et où le compilateur se charge d'effectuer toutes les déductions possibles à partir de ces informations.

On a vu que la logique des propositions était également étroitement liée à la construction des conditions en programmation (pour des instructions conditionnelles ou encore des boucles). Pour de nombreux systèmes décisionnels, une telle logique à deux valeurs (vrai et faux) suffit. Par contre, quand on entre dans le monde l'intelligence artificielle, on se rend rapidement compte que ces deux extrêmes ne suffisent pas pour modéliser des raisonnements plus complexes ou plus fins.

Dans ce cas-là, on remplace souvent cette « logique à deux valeurs » par ce qu'on appelle la « logique floue » (fuzzy logic en anglais). En logique floue, on ne considère plus seulement deux valeurs de vérité (vrai et faux) mais toute une gamme de « gris » entre ces deux extrêmes. Bien souvent, on utilise l'intervalle des réels allant de 0 à 1, 0 signifiant « faux à coup sûr » et 1 signifiant « vrai à coup sûr ». En logique floue, la valeur 0,5 représente l'incertitude : c'est la valeur de vérité d'une phrase dont on ne sait pas si elle est vraie ou fausse, alors qu'une valeur telle que 0,75 indique une certitude relative (il y a une certitude de 75% que la phrase soit vraie).

La logique floue, qui est assez similaire à l'étude des probabilités sous certains aspects, possède des applications dans toute une série de domaines tels que la médecine (aide au diagnostic), l'étude de l'environnement ou la robotique (reconnaissance vocale, reconnaissance visuelle…) par exemple.