# INFO-F-302, Cours d'Informatique Fondamentale

Emmanuel Filiot
Département d'Informatique
Faculté des Sciences
Université Libre de Bruxelles

Année académique 2020-2021

# Logique Propositionnelle : Introduction

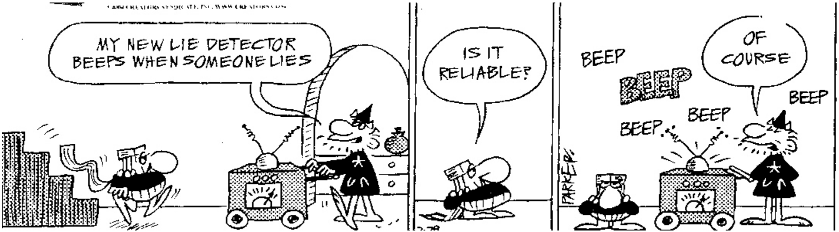
# 3 Qu'est-ce qu'un langage logique ?

- langage permettant de décrire avec précision et rigueur des énoncés et des raisonnements, basés notamment sur des connecteurs Booléens (et, ou, négation)
- langages équipés d'une sémantique claire et non-ambigüe
- les énoncés logiques seront appelés "formules"
- *but* : pallier à l'imprécision des langages naturels
- *but supplémentaire en informatique* : analyser de manière algorithmique les énoncés

# 4 Fausses vérités et paradoxes

- l'utilisation du langage naturel comme notation est imprécise et peut mener à des énoncés faux ...
  1. Tout ce qui est rare est cher
  2. *Or* une chose pas chère, c'est rare
  3. *Donc* une chose pas chère, c'est cher.
- ... ou à des paradoxes
  - Cette phrase est fausse. Problème de *l'auto référence*.
  - Un crocodile s'empare d'un bébé crocodile et propose à sa mère de récupérer son bébé si elle devine ce qu'il va en faire, ce à quoi elle répond "tu vas le dévorer". Que va donc faire le crocodile ?
  - Le barbier rase tous les hommes qui ne se rasent pas eux-mêmes et seulement ceux-là. Qui rase le barbier ?

# 5 Un petit dernier ...



# Exemples d'application de la logique en informatique

- design de circuits numériques
- vérification *hardware* et *software*, *model-checking*. Logiques temporelles.
- preuves de programmes
- bases de données : SQL.
- ...

# 7 Exemple de raisonnement

Considérons la situation décrite par les affirmations suivantes :

1. **Si** le train arrive en retard **et** il n'y a pas de taxis à la gare **alors** l'invité arrive en retard.
2. L'invité n'est pas en retard.
3. Le train est arrivé en retard.

Et la déduction suivante : il y avait des taxis à la gare.

# 7 Exemple de raisonnement

Considérons la situation décrite par les affirmations suivantes :

1. **Si** le train arrive en retard **et** il n'y a pas de taxis à la gare **alors** l'invité arrive en retard.
2. L'invité n'est pas en retard.
3. Le train est arrivé en retard.

Et la déduction suivante : il y avait des taxis à la gare.

## Question

Pourquoi peut-on déduire qu'il y avait des taxis à la gare ?

Premièrement, si on met l'affirmation 1 et l'affirmation 3 ensemble, on peut affirmer que s'il n'y avait pas eu de taxis à la gare, alors l'invité serait arrivé en retard. D'après l'affirmation 2, l'invité n'est pas arrivé en retard. Donc il y avait des taxis à la gare.

# 8 Autre Exemple

Considérons un autre exemple :

1. **Si** il pleut **et** l'invité a oublié son parapluie **alors** l'invité est trempé.
2. L'invité n'est pas trempé.
3. Il pleut.

Et la déduction suivante : l'invité n'a pas oublié son parapluie.

# 8 Autre Exemple

Considérons un autre exemple :

1. **Si** il pleut **et** l'invité a oublié son parapluie **alors** l'invité est trempé.
2. L'invité n'est pas trempé.
3. Il pleut.

Et la déduction suivante : l'invité n'a pas oublié son parapluie.

## Question

Pourquoi peut-on déduire que l'invité n'a pas oublié son parapluie ?

Premièrement, si on met l'affirmation 1 et l'affirmation 3 ensemble, on peut affirmer que si l'invité avait oublié son parapluie, alors il serait trempé. D'après l'affirmation 2, l'invité n'est pas trempé. Donc l'invité n'a pas oublié son parapluie.

# 9 Remarque

La deuxième démonstration suit la même **structure logique** que la première démonstration. Il suffit de remplacer les fragments de phrase comme suit :

| Exemple du train | Exemple du parapluie |
|---|---|
| Le train est arrivé en retard | Il pleut |
| Il y a des taxis à la gare | L'invité a son parapluie |
| L'invité est en retard | L'invité est mouillé. |

# 10 Formalisation Logique

| Exemple du train | Exemple du parapluie | Proposition |
|---|---|---|
| Le train est arrivé en retard | Il pleut | $p$ |
| Il y a des taxis à la gare | L'invité a son parapluie | $q$ |
| L'invité est en retard | L'invité est mouillé | $r$ |

## Démonstration

1. Hypothèse : si $p$ et non $q$, alors $r$
2. Hypothèse : $p$
3. Hypothèse : non $r$
4. Déduction : si non $q$ alors $r$
5. Déduction : comme non $r$, alors $q$.

# 11 Formalisation logique

Nous pouvons *formaliser* les trois hypothèses de chacun de ces deux premiers exemples par la *formule logique* suivante :

En LaTex

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

\models

Et nous pouvons *formaliser* la déduction par :

$$((p \wedge \neg q) \rightarrow r) \wedge \neg r \wedge p \models q$$

Où "$\models$" se lit : "$q$ est une conséquence logique de $((p \wedge \neg q) \rightarrow r) \wedge \neg r \wedge p$".

## 12 Conditions Booléennes dans les langages de programmation

Considérons le test suivant :

$$\textbf{if } (x \geq 3 \textbf{ and } z \leq 3) \textbf{ or } (y \leq 2 \textbf{ and } z \leq 3) \textbf{ then } \ldots$$

Il peut-être remplacé par

$$\textbf{if } (x \geq 3 \textbf{ or } \; y \leq 2) \textbf{ and } z \leq 3 \textbf{ then } \ldots$$

Car $(p \vee q) \wedge r$ est logiquement équivalent à $(p \wedge r) \vee (q \wedge r)$.

# Logique Propositionnelle

Définitions de base : Syntaxe, Sémantique et Exemples

## 14 Construction des formules

Le *vocabulaire du langage de la logique propositionnelle* est composé :

- d'un ensemble, fini ou dénombrable, de *propositions* notées $x, y, z, \ldots$
  Dans la suite, nous notons les ensembles de propositions par les lettres $X$, $Y$,... ;
- de deux constantes : vrai (notée $\top$) et faux (notée $\bot$), parfois notée $1$ et $0$ ;
- d'un ensemble de *connecteurs logiques* : et (noté $\wedge$), ou (noté $\vee$), non (noté $\neg$), implique (noté $\rightarrow$), équivalent (noté $\leftrightarrow$) ;
- les parenthèses : $($,$)$.

# 15 Construction des formules

Soit $X$ un ensemble de propositions. Les *formules de la logique propositionnelle* respectent la règle de formation BNF (Bachus-Naur Form) suivante :

$$\phi ::= \top \mid \bot \mid x \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid \phi_1 \leftrightarrow \phi_2 \mid \neg\phi_1 \mid (\phi_1)$$

Où $\top$, $\bot$ sont respectivement les constantes vrai et faux, $x \in X$ est une proposition et $\phi_1$, $\phi_2$ sont des formules propositionnelles bien formées.

La BNF se lit "une formule de la logique propositionnelle est soit la formule $\top$, soit la formule $\bot$, soit une proposition $x$, soit la conjonction de deux formules $\phi_1$ et $\phi_2$ (définition inductive), etc."

## 16 Quelques exemples de formules

Voici quelques exemples de formules et leur lecture intuitive :

- la formule propositionnelle "$x \rightarrow (y \wedge z)$", peut être lue de la façon suivante "$x$ implique $y$ et $z$", ou peut être également lue comme "si $x$ est vrai alors $y$ et $z$ doivent être vrais" ;
- la formule proxositionnelle "$\neg(x \wedge y)$", peut-être lue de la façon suivante "il est faux que $x$ et $y$ soient vrais (en même temps)".

## 17 Ambiguïtés

L'utilisation de la *notation infixe* s'accompagne de problèmes de lecture :

Comment lire $x \wedge y \vee z$ ? $x \rightarrow y \rightarrow z$ ?

# 17 Ambiguïtés

L'utilisation de la *notation infixe* s'accompagne de problèmes de lecture :

Comment lire $x \wedge y \vee z$ ? $x \rightarrow y \rightarrow z$ ?

Pour lever les ambiguïtés, on utilise les parenthèses ou des règles de priorité entre opérateurs :

- si $op_1$ a une plus grande précédence (priorité) que $op_2$ alors
  $e_1 \; op_1 \; e_2 \; op_2 \; e_3$ est équivalent à $((e_1 \; op_1 \; e_2) op_2 \; e_3)$
  - Par ex : $2 \times 3 + 5 = (2 \times 3) + 5 = 11 \neq 2 \times (3 + 5) = 16$.
- si $op_2$ a une plus grande précédence (priorité) que $op_1$ alors
  $e_1 \; op_1 \; e_2 \; op_2 \; e_3$ est équivalent à $(e_1 op_1 \; (e_2 op_2 \; e_3))$
  - Par ex : $2 + 3 \times 5 = 2 + (3 \times 5) = 17$.
- si $op$ est associatif à gauche alors
  $e_1 \; op \; e_2 \; op \; e_3$ est équivalent à $(e_1 \; op \; e_2) op \; e_3$
  - Par ex : $10/2/5 = (10/2)/5 = 1 \neq 10/(2/5) = 25$.
- associativité à droite : $e_1 \; op \; e_2 \; op \; e_3$ se lit $e_1 op(e_2 op \; e_3)$

On fixe des règles de priorité entre opérateurs afin d'avoir une lecture unique de la formule.

# 18 Règles de précédence en logique propositionnelle

Ordre de précédence $\prec$ sur les opérateurs :

$$\leftrightarrow \; \prec \; \rightarrow \; \prec \; \vee \; \prec \; \wedge \; \prec \; \neg$$

et associativité à gauche pour $\leftrightarrow, \vee, \wedge$ et à droite pour $\rightarrow$.

## Exemples

| | | |
|---|---|---|
| $x \vee y \wedge z$ | se lit | $x \vee (y \wedge z)$ |
| $x \rightarrow y \rightarrow x$ | se lit | $x \rightarrow (y \rightarrow x)$ |
| $x \vee y \rightarrow z$ | se lit | $(x \vee y) \rightarrow z$ |
| $\neg x \wedge y$ | se lit | $(\neg x) \wedge y$ |
| $x \rightarrow y \wedge z \rightarrow t$ | se lit | |

$x_1 \rightarrow (x_2 \rightarrow (x_3 \rightarrow x_4))$

A. $(x \rightarrow y) \wedge (z \rightarrow t)$ ✗

B. $x \rightarrow ((y \wedge z) \rightarrow t)$ ✓

C. $(x \rightarrow (y \wedge z)) \rightarrow t$ ✗

# 18 Règles de précédence en logique propositionnelle

Ordre de précédence $\prec$ sur les opérateurs :

$$\leftrightarrow \ \prec \ \rightarrow \ \prec \ \vee \ \prec \ \wedge \ \prec \neg$$

et associativité à gauche pour $\leftrightarrow, \vee, \wedge$ et à droite pour $\rightarrow$.

## Exemples

| | |
|---|---|
| $x \vee y \wedge z$ | se lit |
| $x \rightarrow y \rightarrow x$ | se lit |
| $x \vee y \rightarrow z$ | se lit |
| $\neg x \wedge y$ | se lit |
| $x \rightarrow y \wedge z \rightarrow t$ | se lit |

## Remarque

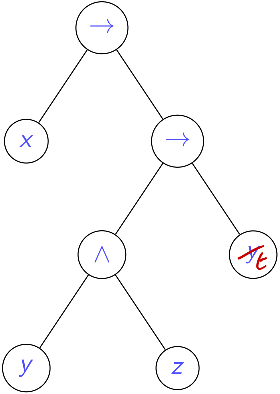
Les parenthèses permettent de contrecarrer ces règles, si elles ne conviennent pas. Elles permettent aussi de rendre une formule plus lisible, ou de ne pas devoir retenir les règles de précédence.

# 19 Arbre correspondant à une formule

La formule $x \to y \wedge z \to t$ est donc équivalente à

$$x \to ((y \wedge z) \to t)$$

et donc son arbre de lecture est :



# 20 La sémantique

- jusqu'ici, on a vu la syntaxe des formules, qui ne sont rien d'autre que des suites de caractères sans signification.
- la sémantique donne du sens aux formules, elle permet de les interpréter et de leur attribuer une *valeur de vérité* vrai ou faux.
- l'interprétation des symboles de proposition est cependant libre et c'est à nous de la fixer. On se donnera donc une une *fonction d'interprétation* $V$ pour l'ensemble $X$ de propositions considérées :

  $$V : X \to \{0, 1\}$$

  Cette fonction assigne à chaque proposition de $X$ la valeur vrai ou la valeur faux.
- Dans la suite, nous utiliserons parfois le terme *valuation* , souvent utilisé dans la littérature, au lieu de fonction d'interprétation.
- A partir d'une fonction d'interprétation $V$ des symboles de proposition, nous allons voir comment interpréter les formules.

# 21 La sémantique

## Définition

- La *valeur de vérité* d'une formule propositionnelle $\phi$ formée à partir des propositions d'un ensemble $X$, évaluée avec la fonction d'interprétation $V$, est notée $[\![\phi]\!]_V$. En particulier, on a $[\![\phi]\!]_V \in \{0, 1\}$.
- La fonction $[\![\phi]\!]_V$ est définie par induction sur la syntaxe de $\phi$ de la façon suivante :
  - $[\![\top]\!]_V = 1$ ; $[\![\bot]\!]_V = 0$ ; $[\![x]\!]_V = V(x)$.
  - $[\![\neg\phi]\!]_V = 1 - [\![\phi]\!]_V$
  - $[\![\phi_1 \vee \phi_2]\!]_V = \max([\![\phi_1]\!]_V, [\![\phi_2]\!]_V)$
  - $[\![\phi_1 \wedge \phi_2]\!]_V = \min([\![\phi_1]\!]_V, [\![\phi_2]\!]_V)$
  - $[\![\phi_1 \rightarrow \phi_2]\!]_V = \max(1 - [\![\phi_1]\!]_V, [\![\phi_2]\!]_V)$.
  - $[\![\phi_1 \leftrightarrow \phi_2]\!]_V = \min([\![\phi_1 \rightarrow \phi_2]\!]_V, [\![\phi_2 \rightarrow \phi_1]\!]_V)$.

Nous notons $V \models \phi$ si et seulement si $[\![\phi]\!]_V = 1$, qui se lit "$V$ satisfait $\phi$".

**QUESTION** Comment définir la sémantique de l'implication?

$$[\![\phi_1 \rightarrow \phi_2]\!]_V = ?$$

| $\phi_1$ | $\phi_2$ | $\phi_1 \rightarrow \phi_2$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

A. $\cancel{\min([\![\phi_1]\!]_V, 1-[\![\phi_2]\!]_V)}$

B. $\cancel{\max([\![\phi_1]\!]_V, 1-[\![\phi_2]\!]_V)}$

C. $\cancel{\min(1-[\![\phi_1]\!]_V, [\![\phi_2]\!]_V)}$

D. $\max(1-[\![\phi_1]\!]_V, [\![\phi_2]\!]_V)$

# 24 Exemples

- Prenons l'interprétation $V_1(x) = 0$, alors on a :

$$[\![x]\!]_{V_1} = 0 \quad [\![\neg x]\!]_{V_1} = 1 \quad [\![x \wedge x]\!]_{V_1} = 0 \quad [\![x \wedge \neg x]\!]_{V_1} = 0$$
$$[\![x \rightarrow x]\!]_{V_1} = 1 \quad [\![x \leftrightarrow x]\!]_{V_1} = 1 \quad [\![\neg x \rightarrow x]\!]_{V_1} = 0$$

- Prenons la formule $\phi = (x \vee y) \wedge (\neg y \vee z)$ et l'interprétation $V_2(x) = V_2(z) = 1$ et $V_2(y) = 0$. Pour calculer la valeur de vérité de $\phi$, on calcule d'abord les valeurs de vérité des sous-formules et on applique les tables de vérité.

$$\begin{array}{c} (x \vee y) \wedge (\neg y \vee z) \\ (1 \vee 0) \wedge (\neg 0 \vee 1) \\ (1 \vee 0) \wedge (1 \vee 1) \\ 1 \wedge (1 \vee 1) \\ 1 \wedge 1 \\ 1 \end{array}$$

Donc $[\![\phi]\!]_{V_2} = 1$.

# 25 Exemples

- Prenons la même formule $\phi = (x \vee y) \wedge (\neg y \vee z)$ et l'interprétation $V_3(z) = 1$ et $V_3(x) = V_3(y) = 0$.

$$
\begin{array}{c}
(x \vee y) \wedge (\neg y \vee z) \\
(0 \vee 0) \wedge (\neg 0 \vee 1) \\
(0 \vee 0) \wedge (1 \vee 1) \\
0 \wedge (1 \vee 1) \\
0 \wedge 1 \\
0
\end{array}
$$

Donc $[\![\phi]\!]_{V_3} = 0$.

## 22 La sémantique sous forme de tables de vérités

L'information contenue dans la définition précédente est souvent présentée sous forme de tables, appelées *tables de vérité* :

| $\top$ | $\bot$ |
|---|---|
| 1 | 0 |

| $\phi$ | $\neg\phi$ |
|---|---|
| 1 | 0 |
| 0 | 1 |

# 23 La sémantique sous forme de tables de vérités

| $\phi_1$ | $\phi_2$ | $\phi_1 \wedge \phi_2$ | $\phi_1 \vee \phi_2$ | $\phi_1 \rightarrow \phi_2$ | $\phi_1 \leftrightarrow \phi_2$ |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 |

# 26 Exemples

On peut utiliser le principe de décomposition en sous-formule et l'application des tables de vérité pour obtenir la valeur de vérité de $\phi$ pour toutes les interprétations possibles des variables

| $x$ | $y$ | $z$ | $(x \vee y)$ | $\neg y$ | $(\neg y \vee z)$ | $(x \vee y) \wedge (\neg y \vee z)$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | | | | |
| 1 | 1 | 0 | | | | |
| 1 | 0 | 1 | | | | |
| 1 | 0 | 0 | | | | |
| 0 | 1 | 1 | | | | |
| 0 | 1 | 0 | | | | |
| 0 | 0 | 1 | | | | |
| 0 | 0 | 0 | | | | |

## 26 Exemples

On peut utiliser le principe de décomposition en sous-formule et l'application des tables de vérité pour obtenir la valeur de vérité de $\phi$ pour toutes les interprétations possibles des variables

| $x$ | $y$ | $z$ | $(x \vee y)$ | $\neg y$ | $(\neg y \vee z)$ | $(x \vee y) \wedge (\neg y \vee z)$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | | | |
| 1 | 1 | 0 | 1 | | | |
| 1 | 0 | 1 | 1 | | | |
| 1 | 0 | 0 | 1 | | | |
| 0 | 1 | 1 | 1 | | | |
| 0 | 1 | 0 | 1 | | | |
| 0 | 0 | 1 | 0 | | | |
| 0 | 0 | 0 | 0 | | | |

# 26 Exemples

On peut utiliser le principe de décomposition en sous-formule et l'application des tables de vérité pour obtenir la valeur de vérité de $\phi$ pour toutes les interprétations possibles des variables

| $x$ | $y$ | $z$ | $(x \vee y)$ | $\neg y$ | $(\neg y \vee z)$ | $(x \vee y) \wedge (\neg y \vee z)$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | | |
| 1 | 1 | 0 | 1 | 0 | | |
| 1 | 0 | 1 | 1 | 1 | | |
| 1 | 0 | 0 | 1 | 1 | | |
| 0 | 1 | 1 | 1 | 0 | | |
| 0 | 1 | 0 | 1 | 0 | | |
| 0 | 0 | 1 | 0 | 1 | | |
| 0 | 0 | 0 | 0 | 1 | | |

# 26 Exemples

On peut utiliser le principe de décomposition en sous-formule et l'application des tables de vérité pour obtenir la valeur de vérité de $\phi$ pour toutes les interprétations possibles des variables

| $x$ | $y$ | $z$ | $(x \vee y)$ | $\neg y$ | $(\neg y \vee z)$ | $(x \vee y) \wedge (\neg y \vee z)$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 1 | |
| 1 | 1 | 0 | 1 | 0 | $\cancel{1}$ 0 | |
| 1 | 0 | 1 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 1 | 1 | 1 | |
| 0 | 1 | 1 | 1 | 0 | 1 | |
| 0 | 1 | 0 | 1 | 0 | 0 | |
| 0 | 0 | 1 | 0 | 1 | 1 | |
| 0 | 0 | 0 | 0 | 1 | 1 | |

# 26 Exemples

On peut utiliser le principe de décomposition en sous-formule et l'application des tables de vérité pour obtenir la valeur de vérité de $\phi$ pour toutes les interprétations possibles des variables

| $x$ | $y$ | $z$ | $(x \vee y)$ | $\neg y$ | $(\neg y \vee z)$ | $(x \vee y) \wedge (\neg y \vee z)$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | $\cancel{1}$ 0 | $\cancel{1}$ 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 |

# 27 Exemples

- $\neg x \vee y$ (qui se lit $(\neg x) \vee y$)

| $x$ | $y$ | $\neg x$ | $\neg x \vee y$ |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 |

- Remarque : c'est la même table de vérité que pour l'implication :

| $x$ | $y$ | $x \to y$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |
| 0 | 0 | 1 |

même colonne

- On dira que les deux formules $\neg x \vee y$ et $x \to y$ sont **équivalentes**.

# 28 Remarques sur l'implication

- l'implication $\phi_1 \rightarrow \phi_2$ modélise le raisonnement *si-alors* :

  **Si** $\phi_1$ est vraie, **ALORS** $\phi_2$ est vraie aussi.

- le cas où $\phi_1$ est faux ne nous intéresse pas dans l'implication.
- par conséquent, si $\phi_1$ est faux, alors peu importe la valeur de vérité de $\phi_2$, l'implication $\phi_1 \rightarrow \phi_2$ reste vraie.

# 29 Un dernier exemple : $\phi = \neg(x \wedge y) \leftrightarrow (\neg x \vee \neg y)$

- Calculons la table de vérité :

| $x$ | $y$ | $x \wedge y$ | $\neg(x \wedge y)$ | $\neg x$ | $\neg y$ | $\neg x \vee \neg y$ | $\phi$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

- pour toute les interprétations possibles des propositions, $\phi$ est vraie. On dira dans ce cas que $\phi$ est **valide**.
- En fait, on a montré ici que les deux formule $\neg(x \wedge y)$ et $\neg x \vee \neg y$ sont équivalentes. C'est une des lois de Morgan.

# 30 Exercices

Dire pour les interprétations $V$ et les formules $\phi$ suivantes si $V \models \phi$ :

1. $V(x) = 1$, $V(y) = 0$ et $\phi = (x \rightarrow y) \wedge (x \rightarrow \neg y)$
2. $V(x) = V(y) = 0$ et $\phi = ((\neg x \rightarrow y) \rightarrow (\neg y \rightarrow x)) \wedge (x \vee y)$
3. $V(x) = 0$ et $V(y) = 1$ et $\phi = ((\neg x \vee y) \rightarrow (y \wedge (x \leftrightarrow y))$
4. $V(x) = V(z) = 1$ et $V(y) = 0$ et
   $\phi = ((\neg z \rightarrow \neg x \wedge \neg y) \vee t) \leftrightarrow (x \vee y \rightarrow z \vee t)$
5. $V(x) = V(y) = 0$ et $V(z) = 1$ et
   $\phi = (x \wedge (y \rightarrow z)) \leftrightarrow ((\neg x \vee y) \rightarrow (x \wedge z))$.

# 31 Validité et Satisfaisabilité

Voici deux définitions importantes :

## Définition (Satisfaisabilité)

Une formule propositionnelle $\phi$ est *satisfaisable* si et seulement si il existe une fonction d'interprétation $V$ pour les propositions de $\phi$, telle que $V \models \phi$.

## Définition (Validité)

Une formule propositionnelle $\phi$ est *valide* si et seulement si pour toute fonction d'interprétation $V$ pour les propositions de $\phi$, on a $V \models \phi$.

## 32 Exemples

| $x$ | $y$ | $x \to y$ | $y \to x$ | $x \to y \vee y \to x$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

| | formule | satisfaisable | valide |
|---|---|---|---|
| 1 | $\top$ | oui | oui |
| 2 | $x$ | oui<br>$V(x) = 1$ | non<br>$V(x) = 0$ |
| 3 | $\neg x$ | oui<br>$V(x) = 0$ | non<br>$V(x) = 1$ |
| 4 | $(x \vee y) \wedge (\neg y \vee z)$ | oui<br>voir table précédente | non<br>voir table précédente |
| 5 | $x \wedge \neg x$ | non | non |
| 6 | $x \wedge \neg y \wedge (\neg y \to \neg x)$ | non | non |
| 7 | $x \vee \neg x$ | oui | oui |
| 8 | $(x \to y) \vee (y \to x)$ | oui | oui |

# 33 Notion de conséquence logique

## Définition

Soit $\phi_1, \ldots, \phi_n, \phi$ des formules. On dira que $\phi$ est une *conséquence logique* de $\phi_1, \ldots, \phi_n$, noté $\phi_1, \ldots, \phi_n \models \phi$, si $(\phi_1 \wedge \cdots \wedge \phi_n) \rightarrow \phi$ est valide.

Par exemple, $p, \neg p \models \bot$ et $\phi_1, \phi_1 \rightarrow \phi_2 \models \phi_2$.

# 34 Application importante de la notion de validité

## Définition

Deux formules $\phi$ et $\psi$ sont dites *équivalentes* si la formule $\phi \leftrightarrow \psi$ est valide. On notera $\phi \equiv \psi$ pour signifier que $\phi$ et $\psi$ sont équivalentes.

- dans une formule, on peut substituer une sous-formule par une autre équivalente sans changer la sémantique de la formule de départ
- par exemple, dans la formule

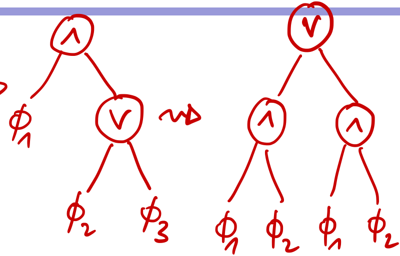
$$\gamma = (x \vee y) \wedge \neg(x \wedge z)$$

on peut remplacer la sous-formule $\neg(x \wedge z)$ par $(\neg x \vee \neg z)$ et on obtient la formule suivante, qui est équivalente à $\gamma$ :

$$\gamma' = (x \vee y) \wedge (\neg x \vee \neg z)$$

- l'application d'équivalences simples va nous permettre de simplifier des formules, et de les mettre sous des formes particulières

# 35 Quelques équivalences




Pour toutes formules $\phi_1, \phi_2, \phi_3$, on a :

- $\neg\neg\phi_1 \equiv \phi_1$ (double négation)
- $\neg(\phi_1 \wedge \phi_2) \equiv (\neg\phi_1 \vee \neg\phi_2)$ (loi de De Morgan pour le 'et')
- $\neg(\phi_1 \vee \phi_2) \equiv (\neg\phi_1 \wedge \neg\phi_2)$ (loi de De Morgan pour le 'ou')
- $\phi_1 \wedge (\phi_2 \vee \phi_3) \equiv (\phi_1 \wedge \phi_2) \vee (\phi_1 \wedge \phi_3)$ (distributivité du 'et')
- $\phi_1 \vee (\phi_2 \wedge \phi_3) \equiv (\phi_1 \vee \phi_2) \wedge (\phi_1 \vee \phi_3$ (distributivité du 'ou')
- $\phi_1 \rightarrow \phi_2 \equiv \neg\phi_2 \rightarrow \neg\phi_1$ (contraposition)

# 36 Lien entre satisfaisibilité et validité

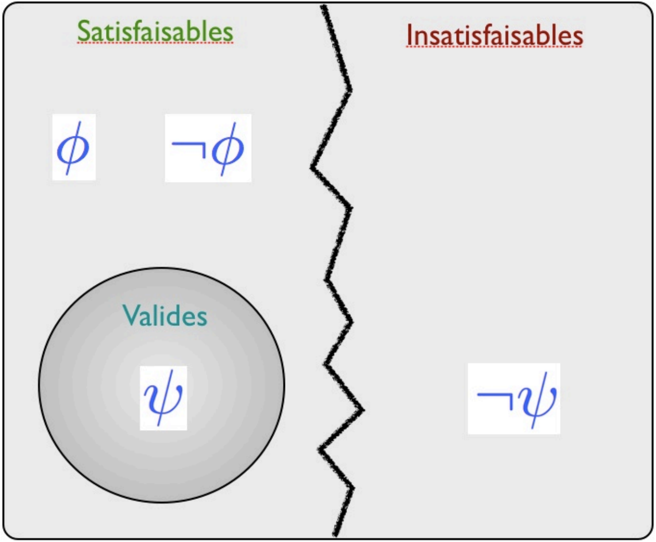
| $x$ | $y$ | $z$ | $\phi$ | $\neg\phi$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| ⋮ | ⋮ | ⋮ | 1 | 0 |
| | | | ⋮ | ⋮ |
| | | | 1 | 0 |

## Théorème

*Une formule propositionnelle $\phi$ est valide ssi sa négation $\neg\phi$ n'est pas satisfaisable.*

Par conséquent, si on dispose d'un algorithme qui décide si une formule est satisfaisable ou non, on obtient un algorithme qui décide si la formule est valide, il suffit de tester la (non-)satisfaisabilité de sa négation.

# 37 Diagramme



# 38 Comment tester la satisfaisabilité d'une formule ?

# 38 Comment tester la satisfaisabilité d'une formule ?

- on a vu la méthode des tables de vérité : tester toutes les interprétations de propositions jusqu'à ce qu'on en trouve une qui satisfait la formule
- **problème :** quelle est la complexité d'un tel algorithme ? combien d'interprétations faut-il tester si la formule contient $n$ propositions ?

# 38 Comment tester la satisfaisabilité d'une formule ?

- on a vu la méthode des tables de vérité : tester toutes les interprétations de propositions jusqu'à ce qu'on en trouve une qui satisfait la formule
- **problème :** quelle est la complexité d'un tel algorithme ? combien d'interprétations faut-il tester si la formule contient $n$ propositions ?
- il faut essayer, dans le pire des cas (c'est à dire le cas où la formule n'est pas satisfaisable), $2^n$ interprétations : cet algorithme a donc une complexité exponentielle dans le nombre de propositions
- ce n'est pas raisonnable pour les applications que nous allons aborder, car nous allons générer des formules contenant plusieurs centaines de propositions.
- nous allons maintenant étudier un algorithme "plus intelligent" : la méthode des tableaux sémantiques.

# QUIZZ

1. VRAI / FAUX Il existe une formule satisfaisable dont la négation est satisfaisable.

VRAI : par exemple la formule $x$

2. VRAI / FAUX Il existe deux formules $\phi$ et $\psi$ telles que $\phi \to \psi$ est valide et $\psi$ est insatisfaisable.

$\phi \to \psi \equiv \neg\phi \vee \psi$ VRAI : il suffit de prendre une formule $\neg\phi$ valide, par ex $\phi = \bot$ $\phi = p \wedge \neg p$

3. Indiquer les formules satisfaisables

A. $p \to (q \to p)$ B. $p \to \neg p$ C. $(p \vee q \vee r) \wedge \neg p \wedge \neg q \wedge (\neg r \vee p)$

D. $(p \to q) \to (\neg q \to \neg p)$

4. Indiquer les formules valides ci-dessus.