Chapitre 14 - Logique

Objectif : formaliser le discours, le raisonnement, la démonstration.

La logique définit formellement :

- le langage qu'on utilise : aspects syntaxiques
- son interprétation : aspects sémantiques

Définition :

La logique propositionnelle manipule des propositions qui peuvent être soit vraies soit fausses.

Exemples :

p_1 = “Il pleut”, p_2 = “Je prends mon parapluie”.

La logique propositionnelle combine ces faits élémentaires à l’aide de connecteurs logiques.

Exemple :

“Il pleut et je prends mon parapluie” : $p_1 \wedge p_2$

“Puisqu’il, pleut, alors je ne prends pas mon parapluie” : $p_1 \rightarrow \neg p_2$

Logique du premier ordre :

Exemple : $2 \leq x < 10$

La logique du premier ordre ajoute des prédicats.

Exemple : $I(a, b)$: prédicat indiquant si $a < b$

On peut ensuite exprimer des phrases plus complexes.

Reprise du premier exemple : $2 \leq x < 10 = I(x, 10) \wedge \neg I(x, 2)$

On ajoute à cela les quantificateurs pour pouvoir formaliser tout raisonnement.

Exemple :

“Tous les hommes sont mortels. Socrate est un homme. Donc Socrate est mortel.”

Prédicats :

- $H(x)$ signifie que x est un homme.
- $M(x)$ signifie que x est mortel.

On ne peut pas écrire M(tous les hommes) car “tous les hommes” n’est pas élémentaire. On utilise donc le quantificateur $\forall$.

Formule correspondant à ce discours :

$$(\forall x(H(x) \to M(x)) \land H(Socrate)) \to M(Socrate)$$

I. Syntaxe des formules

1. Définition des formules propositionnelles

Définition :

Une variable propositionnelle est une proposition élémentaire, qui ne peut prendre que deux états, appelés valeurs de vérité.

L’ensemble des variables propositionnelles est noté $\mathcal{V}$.

Définition :

Deux symboles $\top$ (top) et $\bot$ (bottom) représentent des propositions élémentaires toujours vraie (pour $\top$) et toujours fausse (pour $\bot$).

Définition inductive : l'ensemble des formules propositionnelles $\mathcal{P}$:

- Les constantes logiques sont les formules propositionnelles $\top \in \mathcal{P}$ et $\bot \in \mathcal{P}$.
- Les variables propositionnelles sont des formules propositionnelles $\mathcal{V} \subset \mathcal{P}$.
- Si $\varphi \in \mathcal{P}$, alors $\neg\varphi \in \mathcal{P}$ avec $\neg$ le connecteur logique de négation d’arité 1.
- Si φ_1 et $\varphi_2 \in \mathcal{P}$ alors :

$(\varphi_1 \wedge \varphi_2) \in P$ avec $\wedge$ le connecteur de conjonction (“et”).

$(\varphi_1 \vee \varphi_2) \in P$ avec $\vee$ le connecteur de disjonction (“ou”).

$(\varphi_1 \to \varphi_2) \in P$ avec $\to$ le connecteur de d’implication.

$(\varphi_1 \leftrightarrow \varphi_2) \in P$ avec $\leftrightarrow$ le connecteur de d’équivalence.

Cette définition est non ambigüe :

- $((\varphi_1 \wedge \varphi_2) \wedge \varphi_3)$
- $(\varphi_1 \wedge (\varphi_2 \wedge \varphi_3))$

On autorise le fait d’écrire une formule sans les $()$ extérieures.

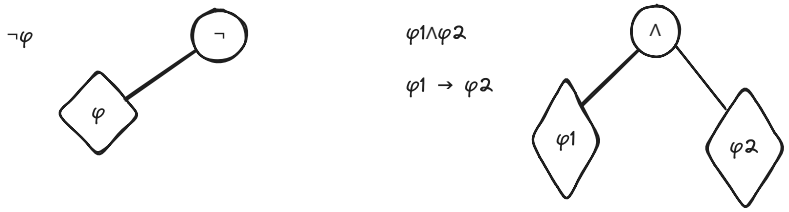
Exemple : $\varphi_1 \wedge \varphi_2$ est considéré comme syntaxiquement correcte.

Remarque : D’autres définitions inductives existent.

2. Représentation arborescente d’une formule

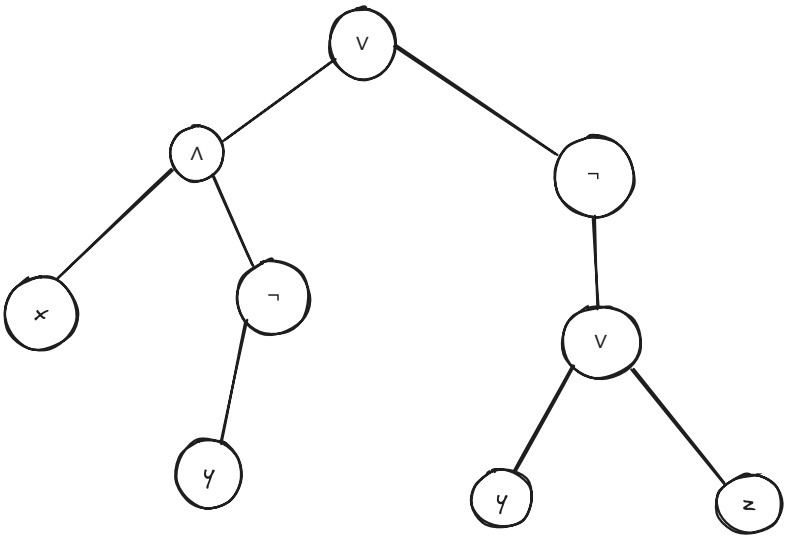
Toute formule propositionnelle admet une représentation arborescente.

Les variables propositionnelles et $\top$ et $\bot$ sont représentés par des feuilles.



Exemple : $x, y, z \in \mathcal{V}$
formule : $((x \wedge \neg y) \vee \neg(y \vee z))$

Pour retrouver la formule propositionnelle depuis sa représentation arborescente, on effectue un parcours infixe(φ à droite des noeuds $\neg$)



Remarque :

L'arité des connecteurs correspond au nombre de fils des noeuds.

Implémentation des formules propositionnelles :

```
type 'a formule =
        |Top
        |Bottom
        |Var_prop of 'a
        |Non of formule
        |Et of formule * formule
        |Ou of formule * formule
        |Impl of formule * formule
        |Equiv of formule * formule
```

En C, on utilise une définition classique d'arbre.

```
struct formule_s {
        char* etiq;
        struct formule_s* gauche;
        struct formule_s* droite;
};
```

On doit vérifier à la main que la formule implémentée est syntaxiquement correcte.

3. Fonctions inductives sur l'ensemble des formules propositionnelles

Taille d'une formule φ, notée $|\varphi|$:

$|\top| = |\bot| = 1$

$|x| = 1$ avec $x \in \mathcal{V}$

$|\neg\varphi| = 1 + |\varphi|$ avec $\varphi \in \mathcal{P}$

$|\varphi_1 \diamond \varphi_2| = 1 + |\varphi_1| + |\varphi_2|$ avec $\diamond \in \{1, \vee, \rightarrow, \leftrightarrow\}$ et $\varphi_1, \varphi_2 \in \mathcal{P}$

Hauteur d'une formule $\varphi, h(\varphi)$:

$h(\bot) = h(\top) = 0$

$h(x) = 0$ avec $x \in \mathcal{V}$

$h(\neg\varphi) = 1 + h(\varphi)$ avec $\varphi \in \mathcal{P}$

$h(\varphi_1 \diamond \varphi_2) = 1 + max(h(\varphi_1), h(\varphi_2))$ avec $\varphi_1, \varphi_2 \in \mathcal{P}$ et $\diamond \in \{1, \vee, \rightarrow, \leftrightarrow\}$

Sous-formules d'une formule propositionnelle φ, notée $SF(\varphi)$:

$SF(\top) = \top, SF(\bot) = \bot$

$SF(x) = \{x\}$ avec $x \in \mathcal{V}$

$SF(\neg\varphi) = \neg\varphi \cup SF(\varphi)$ avec $\varphi \in \mathcal{P}$

$SF((\varphi_1 \diamond \varphi_2)) = (\varphi_1 \diamond \varphi_2) \cup SF(\varphi_1) \cup SF(\varphi_2)$ avec $\varphi_1, \varphi_2 \in \mathcal{P}$ et $\diamond \in \{1, \vee, \rightarrow, \leftrightarrow\}$

Substitution d'une variable propositionnelle x par une formule ψ dans une formule φ, notée $\varphi[\psi/x]$.

$\psi[\varphi/x] = \bot, \top[\psi/x] = \top$

$x[\psi/x] = \psi$

$x'[\psi/x] = x'$ avec $x' \in \mathcal{V}$ et $x' \neq x$

$\neg\varphi[\psi/x] = \neg(\varphi[\psi/x])$ avec $\varphi \in \mathcal{P}$

$(\varphi_1 \diamond \varphi_2)[\psi/x] = (\varphi_1[\psi/x]) \diamond (\varphi_2[\psi/x])$ avec $\diamond \in \{1, \vee, \rightarrow, \leftrightarrow\}$ et $\varphi_1, \varphi_2 \in \mathcal{P}$

4. Logique du premier ordre

Permet de manipuler des objets qui ne sont pas de nature logique(termes) sur lesquels on appliquera des prédicats.

Définition :

Un domaine est constitué de :

- un ensemble de symboles de variables X.
- un ensemble de symboles de fonctions, d'arité a, S_f^a.
- un ensemble de symboles de prédicats, d'arité a, S_p^a.

Définition inductive des termes :

- Toute variable de X est un terme.
- Si $t_0, t_1, \ldots, t_a$ sont des termes, et $f \in S_f^a$., alors $f(t_1, t_2, \ldots, t_a)$ est un terme.

Définition :

Un atome est le résultat de $p(t_1, \ldots, t_a)$ avec $p \in S_p^a$ et $t_1, .., t_a$ des termes.

Définition :

- Le quantificateur existentiel $\exists$ exprime qu'une propriété est vraie pour au moins un élément du domaine.
- Le quantificateur universel $\forall$ exprime qu'une propriété est vraie pour tous les éléments du domaine.

Définition inductive d'une formule du premier ordre :

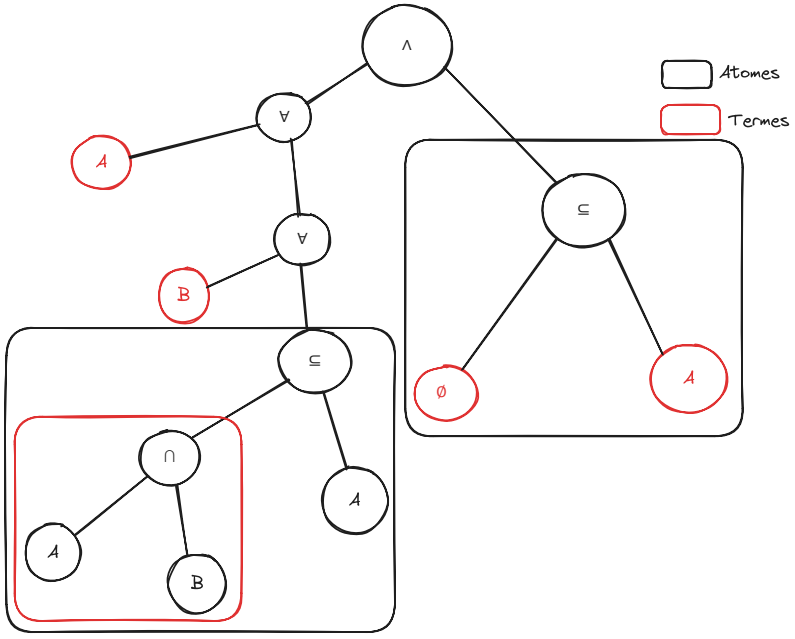
- Tout atome est une formule du premier ordre.
- si φ est une formule du premier ordre, alors $\neg\varphi$ aussi
- Si φ_1, φ_2 formules du premier ordre, alors $(\varphi_1 \diamond \varphi_2)$ aussi avec $\diamond \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.
- Si x est une variable du domaine, φ une formule du premier ordre, alors $(\forall x.\, \varphi)$ et $(\exists x.\, \varphi)$ formules du premier ordre.

Exemple :

$(\forall A, \forall B, A \cap B \subseteq A) \wedge (\varnothing \subseteq A)$ est une formule du premier ordre sur le domaine suivant :

$X = \{A, B\}$, $S_f = \{\varnothing$ d'arité $0, \cap$ d'arité $2\}$, $S_p = \{\subseteq$ d'arité $2\}$.

Représentation arborescente de la formule :



Vocabulaire :

- Dans $\forall x. \varphi$ et $\exists x\ .\ \varphi$, la portée de x est φ.
- Une variable x qui appartient à la suite d'un $\forall x.$ ou $\exists x.$ est dite liée, sinon elle est libre. Une même variable peut avoir des occurrences libres et liées.

II. Sémantique du calcul propositionnel

Objectif : Interpréter les formules propositionnelles, leur donner un sens.

1. Valeurs de vérité d'une formule propositionnelle

Définition :

L'ensemble $\{V, F\}$ est appelé valeurs de vérité.

Définition :

On appelle valuation toute fonction $\mathcal{V} \mapsto \{V, F\}$ qui à chaque variable propositionnelle lui associe une valeur de vérité.

Propriété :

Il existe 2^n valuations pour une formule à n variables propositionnelles.

Définition :

On appelle fonction booléenne d'arité n, toute fonction de $\{V, F\}^n \mapsto \{V, F\}$. On peut associer une fonction booléenne a chaque connecteur ($\neg$,$\vee$,$\wedge$,$\rightarrow$,$\leftrightarrow$).

Fonctions booléennes des connecteurs :

Soit $f_\neg$ la fonction booléenne associée au connecteur de négation $\neg$.

$$\begin{cases} f_\neg(V) = F \\ f_\neg(F) = V \end{cases}$$

Soit $f_\wedge$ la fonction booléenne associée à $\wedge$.

$$\begin{cases} f_\wedge(F, F) = F \\ f_\wedge(F, V) = F \\ f_\wedge(V, F) = F \\ f_\wedge(V, V) = V \end{cases}$$

On définit de même f_V, $f_\rightarrow$ et $f_\leftrightarrow$.

On peut représenter le résultat d'une fonction booléenne par une table de vérité. Chaque ligne de cette table correspond à une valuation possible.

Définition inductive de l'évaluation d'une formule propositionnelle par une valuation :

Pour toute formule propositionnelle v, l'évaluation de φ par v consiste à déterminer la valeur de vérité de φ en appliquant les fonctions booléennes associées aux connecteurs. On note $[\![\varphi]\!]_v$

Définition inductive de $[\![\varphi]\!]_v$:

- $[\![\top]\!]_v = V$
- $[\![\bot]\!]_v = F$
- $[\![x]\!]_v = v(x)$
- $[\![\neg\varphi]\!]_v = f_\neg([\![\varphi]\!]_v)$
- $[\![\varphi_1 \diamond \varphi_2]\!]_v = f_\diamond([\![\varphi_1]\!]_v, [\![\varphi_2]\!]_v)$ avec $\diamond \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

Exemple :

$\varphi = ((x \to y) \vee (\neg x \wedge y)) \wedge (x \vee \neg y)$ valuation v telle que $v(x) = F, v(y) = F$

$[\![\varphi]\!]_v = f_\wedge(f_v(f_\to([\![x]\!]_v, [\![y]\!]_v), f_\wedge(f_\neg([\![x]\!]_v), [\![y]\!]_v)), f_v([\![x]\!]_v, f_\neg([\![y]\!]_v))) = V.$

Si on souhaite trouver $[\![\varphi]\!]_v$ pour toute valuation v, on dresse la table de vérité (faire comme au chapitre 0 de maths, chaque ligne est une instruction et on les assemble pour obtenir l'expression voulue dans la table de vérité).

La table de vérité est une représentation de la fonction d'évaluation d'une formule.

Propriété :

Il existe 2^{2^n} tables de vérité distincts pour les formules propositionnelles à n variables propositionnelles.

Définition :

On appelle modèle de φ toute valuation v telle que $[\![\varphi]\!]_v = V$. L'ensemble des modèles de φ est noté $Mod(\varphi)$.

Définition :

Une formule propositionnelle est dite :

- satisfiable si elle admet un modèle.
- Une tautologie si toute valuation de la formule est un modèle.
- Une antilogie si elle admet aucun modèle.

Méthodologie pour déterminer si une formule φ est satisfiable, tautologique, antilogique :

Dresser la table de vérité de φ.

Notation :

Si φ est tautologique, on note $\vDash \varphi$.

Remarque sur les constantes logiques :

- $\bot$ est une antilogie
- $\top$ est une tautologie

2. Équivalence et conséquence sémantique

Définition :

On dit qu'une formule ψ est une conséquence sémantique d'une formule φ si pour toute valuation v telle que $[\![\varphi]\!]_v = V$, alors $[\![\psi]\!]_v = V$. Autrement dit, $Mod(\varphi) \subseteq Mod(\psi)$. On note alors $\varphi \vDash \psi$.

Exemple :

$x \wedge y \vDash x \rightarrow y$.

On le vérifie avec la table de vérité (toutes les valeurs vraies dans la colonne $x \wedge y$ doivent être vraie dans la colonne de $x \rightarrow y$, ici c'est le cas).

Propriété :

Si $\varphi_1 \vDash \varphi_2$ et $\varphi_2 \vDash \varphi_3$ alors $\varphi_1 \vDash \varphi_3$.

Généralisation :

Soit Γ un ensemble de formules. On note $\Gamma F \psi$ et on dit que ψ est une conséquence sémantique de l'ensemble Γ si pour toute valuation v telle que pour toute formule φ de Γ alors $[\![\varphi]\!]_v = V$, alors $[\![\psi]\!]_v = V$.

Exemple :

Montrons que $\{x \rightarrow y, y \rightarrow z\} \vDash \{x \rightarrow z\}$

$\Gamma = \{x \rightarrow y, y \rightarrow z\}$. Il faut que les où il y a V et V dans les lignes $x \rightarrow y$ et $y \rightarrow z$ soit également vrai dans la ligne $x \rightarrow z$. On utilise une table de vérité.

Définition :

Deux formules φ et ψ sont dites sémantiquement équivalentes si pour toute valuation v, $[\![\varphi]\!]_v = [\![\psi]\!]_v$. Autrement dit, $Mod(\varphi) = Mod(\psi)$. On note $\varphi \equiv \psi$.

Propriété :

$\varphi \equiv \psi$ si et seulement si $\varphi \vDash \psi$ et $\psi \vDash \varphi$.

Propriété :

$\equiv$ est une relation d'équivalence.

Propriété :

Si $\varphi_1 \equiv \varphi_2$ et $x \in \mathcal{V}$. alors $\varphi_1[\psi/x] \equiv \varphi_2[\psi/x]$ et $\psi[\varphi_1/x] \equiv \psi[\varphi_2/x]$.

Équivalences sémantiques fondamentales :

- Implication : $\varphi \to \psi \equiv \neg\varphi \vee \psi$.
- Lois de Morgan : $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$ et $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$.
- Contraposition : $\varphi \to \psi \equiv \neg\psi \to \neg\varphi$.
- Curryfication : $(\varphi \wedge \psi) \to \theta \equiv \varphi \to (\psi \to \theta)$.
- Double implication : $\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \wedge (\psi \to \varphi)$.
- Disjonction de cas : $(\varphi \to \psi) \wedge (\neg\varphi \to \psi) \equiv \psi$.
- Absurde : $(\neg\varphi \to \bot) \equiv \varphi$.
- Tiers exclus : $\varphi \wedge \neg\varphi \equiv \bot$ et $\varphi \vee \neg\varphi \equiv \top$.
- Double négation : $\neg\neg\varphi \equiv \varphi$.
- Elément neutre : $\varphi \wedge \top \equiv \varphi$ et $\varphi \vee \bot \equiv \varphi$.
- Elément absorbant : $\varphi \wedge \bot \equiv \bot$ et $\varphi \vee \top \equiv \top$.
- Commutativité : $\varphi \wedge \psi \equiv \psi \wedge \varphi$ et $\varphi \vee \psi \equiv \psi \vee \varphi$.
- Associativité : $(\varphi \wedge \psi) \wedge \theta \equiv \varphi \wedge (\psi \wedge \theta)$ et $(\varphi \vee \psi) \vee \theta \equiv \varphi \vee (\psi \vee \theta)$.
- Distributivité : $\varphi \wedge (\psi \vee \theta) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \theta)$ et $\varphi \vee (\psi \wedge \theta) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \theta)$.
- Idempotence : $\varphi \wedge \varphi \equiv \varphi$ et $\varphi \vee \varphi \equiv \varphi$.

Exemple :

Le problème est : trois personnes mangent ensemble : A, B, C. On sait que :

- (1) si A prend un dessert, B aussi.
- (2) soit B, soit C prennent un dessert, mais pas les deux.
- (3) A ou C prend : un dessert.
- (4) si C prend un dessert, A aussi.

La question est : qui a pris un dessert ?

Résolution :

On introduit les variables propositionnelles suivantes :

- a représentant “A prend un dessert”.
- b représentant “B prend un dessert”.
- c représentant “C prend un dessert”.

Les conditions du problème avec ces variables s’écrivent ainsi :

- $(1)\ a \to b$.
- $(2)\ (b \vee c) \wedge \neg(b \wedge c)$.
- $(3)\ a \vee c$.
- $(4)\ c \to a$.

La formule φ représentant le problème est :

$\varphi = (a \to b) \wedge ((b \vee c) \wedge \neg(b \wedge c)) \wedge (a \vee c) \wedge (c \to a)$.

Simplification :

$\varphi \equiv (\neg a \vee b) \wedge (b \vee c) \wedge (\neg b \vee \neg c) \wedge (a \vee c) \wedge (\neg c \vee a)$.

$\varphi \equiv (\neg a \vee b) \wedge (b \vee c) \wedge (\neg b \vee \neg c) \wedge ()a \vee (c \wedge \neg c))$.

$\varphi \equiv (\neg a \vee b) \wedge (b \vee c) \wedge (\neg b \vee \neg c) \wedge a$.

$\varphi \equiv a \wedge b \wedge (b \vee c) \wedge (\neg b \vee \neg c)$.

$\varphi \equiv a \wedge b \wedge \neg c$.

Donc “A prend un dessert” et “B prend un dessert” et “C ne prend pas de dessert”.

3. Systèmes complets de connecteurs

On a défini l’ensemble $\mathcal{P}$ avec les connecteurs $\{\neg, \wedge, \vee, \to, \leftrightarrow\}$. Mais les règles d’équivalences fondamentales nous permettent d’écrire des formules propositionnelles en n’utilisant que $\{\neg, \wedge, \vee\}$.

Définition :

Un ensemble de connecteurs forment un système complet si toute formule propositionnelle peut être écrite en utilisant uniquement ces connecteurs.

Théorème :

Le système $\{\neg, \wedge, \vee\}$ est complet.

Pour le montrer, on dresse les $2^{2^2} = 16$ tables de vérité existantes et pour chaque table, on trouve une formule utilisant uniquement ces trois connecteurs qui correspond.

x	y	φ	I
F	F	V	valuation $= \neg x \wedge \neg y$
F	V	V	valuation $= \neg x \wedge y$
V	F	F	valuation $= x \wedge \neg y$
V	V	F	valuation $= x \wedge y$

On écrit chaque ligne à l'aide des connecteurs $\wedge$ et $\neg$. On fait ensuite une disjonction (V) entre les lignes où la valeur de vérité est V. Ici, $\varphi = (\neg x \wedge \neg y) \vee (\neg x \wedge y)$.

III. Problème SAT

1. Formes normales

Objectif : normaliser les formules propositionnelles.

a. Formes normales simples

Définition :

Un littéral est soit une variable propositionnelle soit la négation d'une variable propositionnelle.

Exemple : $(x \wedge (\neg y \vee z)) \rightarrow \neg(\neg x \wedge \neg z)$. Les littéraux sont les variables munis des symboles adjacents ($\neg$ inclus).

Définition :

Une formule propositionnelle mise sous forme normale négative ne contient que des conjonctions, disjonctions, et littéraux.

Exemple : La formule précédente n'est pas une forme normale négative.

$(x \wedge (\neg y \vee z)) \vee y$ en est une tandis que $x \wedge \neg(y \vee z)$ n'en est pas une.

Méthode de mise sous forme normale négative :

- On trouve une formule équivalente sémantiquement en retirant les $\rightarrow$ et les $\leftrightarrow$ pour ne garder que $\wedge, \vee, \neg$.
- Retirer les $\neg$ qui ne sont pas sur des variables en utilisant les lois de Morgan.
- Retirer les doubles négations.

Définition :

- Une clause conjonctive est une conjonction de littéraux.
- Une clause disjonctive est une disjonction de littéraux.

Remarque :

$\bot$ est le neutre pour $\vee$, donc $\bot$ est considéré comme une clause disjonctive. De même $\top$ est le neutre de $\wedge$ donc $\top$ est considéré comme une clause conjonctive.

Définition :

Une formule en forme normale conjonctive est une conjonction de clauses disjonctives.

Exemple :

- $x \wedge (y \vee z) \wedge (\neg x \vee \neg z)$ est en forme normale conjonctive.
- $x \wedge y$ est en forme normale conjonctive (somme de deux clauses).
- $x \vee y$ est en forme normale conjonctive (une seule clause).

Une forme normale conjonctive est aussi en forme normale négative.

Méthode pour mettre une formule propositionnelle en forme normale conjonctive :

- Mettre la formule en forme normale conjonctive.
- Utiliser les équivalences fondamentales (distributivité notamment).

Exemple : $x \vee (y \wedge \neg z) \equiv (x \vee y) \wedge (x \vee \neg z)$

Définition :

Une formule en forme normale disjonctive est une disjonction de clauses conjonctives.

Exemple :

- $(x \wedge y) \vee (\neg x \wedge z) \vee y$ est en forme normale disjonctive (trois clauses conjonctives séparées par des $\vee$).
- $x \vee y$ est en forme normale disjonctive.
- $x \wedge y$ est en forme normale disjonctive (à 1 clause).

Une forme normale disjonctive est aussi en forme normale négative.

Méthode pour mettre une formule propositionnelle en forme normale disjonctive :

- Mettre la formule en forme normale négative.
- Utiliser les équivalences fondamentales (distributivité notamment).

Exemple : $x \wedge (y \vee \neg z) \equiv (x \wedge y) \vee (x \wedge \neg z)$.

Souvent, la mise en forme normale (conjonctive ou disjonctive) augmente la taille de la formule.

Solution alternative pour implémenter une formule :

On utilise une liste (la formule) contenant des listes (les clauses) contenant des entiers (littéraux) positifs (variable) et négatifs (négation des variables).

Exemple : $(x \wedge y \wedge z) \vee (x \wedge \neg y) \vee \neg z$ est en forme normale disjonctive.

On pose par exemple : $x \leftrightarrow 1, y \leftrightarrow 2, z \leftrightarrow 3$. L'implémentation de cette formule donne :

$[[1; 2; 3]; [1; -2]; [-3]]$

b. Formes normales canoniques

Méthode pour trouver une forme normae disjonctive et forme normale conjonctive canoniques :

On dresse la table de vérité.

Exemple : $\varphi = (x \wedge \neg z) \rightarrow (\neg x \wedge \neg(y \wedge \neg z))$.

x	y	z	φ	/
F	F	F	V	$\neg x \wedge \neg y \wedge \neg z$
F	F	V	V	$\neg x \wedge \neg y \wedge z$
F	V	F	V	
F	V	V	V	
V	F	F	F	
V	F	V	V	
V	V	F	F	
V	V	V	V	

Forme normale disjonctive (=V) :
$\varphi \equiv (\neg x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge \neg z) \vee (\neg x \wedge y \wedge z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge z)$
Forme normale conjonctive (=F) :

$\varphi \equiv \neg(x \wedge \neg y \wedge \neg z) \wedge \neg(x \wedge y \wedge \neg z) \equiv (\neg x \vee y \vee z) \wedge (\neg x \vee \neg y \vee z)$

Forme normale disjonctive canonique :

- Chaque valuation donne une clause conjonctive.
- La forme normale disjonctive canonique est une disjonction entre tous les modèles (lignes V) de la formule.

Forme normale conjonctive canonique :

- On fait une conjonction des négations des valuations donnant F
- Appliquer les lois de Morgan et la double négation sur chaque valuation.

Définition :

Un min-terme sur l'ensemble $\mathcal{V}$ est une conjonction de $|\mathcal{V}|$ dans laquelle chaque variable apparait une fois.

Définition :

La forme normale disjonctive canonique est une disjonction de min-termes différents.

Définition :

Un max-terme sur l'ensemble $\mathcal{V}$ est une disjonction de $|\mathcal{V}|$ littéraux dans laquelle chaque variable apparait une fois.

Définition :

La forme normale conjonctive est une conjonction de max-termes différents.

Théorème :

Toute formule propositionnelle est équivalente à une unique forme normale disjonctive canonique et une unique forme normale conjonctive canonique (à ordre près des min-termes/max-termes).

Intérêt :

Soit φ une formule propositionnelle.

φ tautologie $\Leftrightarrow$ sa forme normale disjonctive canonique contient tous les min-termes $\Leftrightarrow$ sa forme normale conjonctive contient aucun max-termes

φ antilogie $\Leftrightarrow$ sa forme normale disjonctive canonique contient rien $\equiv \bot \Leftrightarrow$ sa forme normale conjonctive contient tous les max-termes

φ satisfiable $\Leftrightarrow$ sa forme normale disjonctive canonique contient au moins un min-termes $\Leftrightarrow$ sa forme normale conjonctive contient au moins un max-termes

Donc si on possède les formes normales canoniques d'une formule, on détermine si elle sont satisfiable en $\mathcal{O}(1)$.

Inconvénient :

Taille des formes normales : Il y a 2^n (avec $n = |\mathcal{V}|$) lignes dans la table, donc 2^n termes répartis dans les formes normales, donc au moins 1 forme normale canonique a une taille exponentielle.

2. Problème K-SAT

Définition :

Un problème SAT est un problème de décision qui détermine si une formule propositionnelle est satisfiable.

SAT(φ) avec φ une formule propositionnelle : Renvoie Vrai si φ est satisfiable, Faux sinon.

Complexité :

Le problème SAT est NP-complet (on suppose qu'il n'existe aucun algorithme avec une complexité pire des cas non exponentielle).

Définition :

Un problème K-SAT est la restriction du problème SAT aux forme normale conjonctive dont chaque clause comporte au plus k littéraux.

Exemple : $\varphi = (\neg a \vee b) \wedge (b \vee c) \wedge (\neg b \vee \neg c) \wedge (a \vee c) \wedge (\neg c \vee a)$.

Le problème des desserts est une instance de 2-SAT.

a. Cas particuliers de 1-SAT et 2-SAT

- Problème 1-SAT :

Si on a $\varphi = x \wedge \ldots \wedge \neg x \wedge \ldots$ avec $x \in \mathcal{V}$ alors φ est insatisfiable.

φ est satisfiable si et seulement si pour chaque variable propositionnelle, φ ne contient jamais cette variable et sa négation.

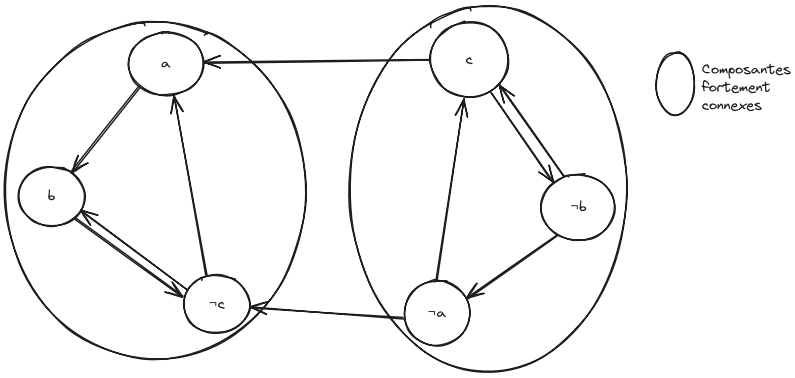
La complexité est linéaire en la taille de la formule.

- Problème 2-SAT :

On représente le problème par un graphe. Chaque clause est de la forme $(l_1 \vee l_2)$
Or $l_1 \vee l_2 \equiv \neg l1 \rightarrow l_2 \equiv \neg l_2 \rightarrow l_1$.

Dans le graphe orienté, on aura donc deux sommets $\neg l_1$ et l_2 avec un arc $(\neg l_1, l_2)$ et 2 sommets $\neg l_2$ et l_1 avec un arc $(\neg l_2, l_1)$ et ce pour chaque clause.

Exemple : Problème des desserts



Théorème :

Une instance de 2-SAT est satisfiable si et seulement si dans le graphe associé, aucune composante fortement connexe ne contient à la fois une variable propositionnelle et sa négation.

Preuve :

Supposons qu'il existe une composante fortement connexe qui contient $x \in \mathcal{V}$ et aussi $\neg x$.

Il y a donc un chemin allant de x à $\neg x$ et un allant de $\neg x$ à x donc $\varphi \vDash \neg x \leftrightarrow x$ (car $x \to \neg x$ et $\neg x \to x$ par transitivité de $\to$).

Or $\neg x \leftrightarrow x \equiv \bot$ donc $\varphi \vDash \bot$ donc φ admet aucun modèle donc φ insatisfiable.

(Sens $\Leftarrow$). On note C_l la composante fortement connexe de l. On s'intéresse au graphe des composantes fortement connexes, orienté et acyclique. (preuve par l'absurde). Ce graphe admet un ordre topologique $\prec$. Montrons que la valuation v suivante est un modèle de φ.

$$\begin{cases} v(X) = V \text{ si } C_{\neg x} \prec C_x \\ v(X) = F \text{ si } C_x \prec C_{\neg x} \end{cases}$$

(On est bien toujours dans 1 des 2 cas car $C_x \neq C_{\neg x}$).

Soit $(l \vee l')$ une clause de la forme normale conjonctive de φ. Il faut donc montrer que v satisfait $l \vee l'$, donc que v satisfait l oy v satisfait l'. Supposons par l'absurde que $v(l) = v(l') = F$.

$v(l) = F$ donc $C_l \prec C_{\neg l}$
$v(l') = F$ donc $C_{l'} \prec C_{\neg l'}$

Or si $l \vee l'$ est une clause, alors $(\neg l, l')$ est un arc $C_{\neg l} \preceq C_{l'}$, $(\neg l', l)$ est un arc donc $C_{\neg l'} \preceq C_l$. Par transitivité de $\prec$ on trouve $C_l \prec C_l \preceq C_{l'} \prec C_{\neg l'} \preceq C_l$, impossible donc v est un modèle de φ.

Complexité :

- Construction du graphe : $|S| = 2 \times |\mathcal{V}|$, $|A| = 2 \times \text{nombre de clauses}$, $\mathcal{O}(|S| + |A|)$.
- recherche des composantes fortement connexes : $\mathcal{O}(|S| + |A|)$.
- vérification des composantes : $\mathcal{O}(|S|)$

Total : $\mathcal{O}(|S| + |A|)$, même ordre de grandeur que la taille de formule, donc linéaire en taille de la formule.

b. Algorithme de Quine

La résolution de K-SAT pour $k > 2$ a un complexité exponentielle : on utilise un algorithme de recherche exhaustive

- soit une recherche par force brute : teste pour chaque valuation existante, s'il s'agit d'un modèle donc cela revient à construire la table de vérité.

  $2^{|V|}$ valuations à tester

  Evaluer la formule φ pour une valuation se fait en $\mathcal{O}(|\varphi|)$
- soit une recherche par retour sur trace (backtracking), c'est l'algorithme de Quine.

Principe :

On part d'une formule φ (à la racine de l'arbre). On choisit une variable propositionnelle x de φ.

- 1ère branche de l'arbre :

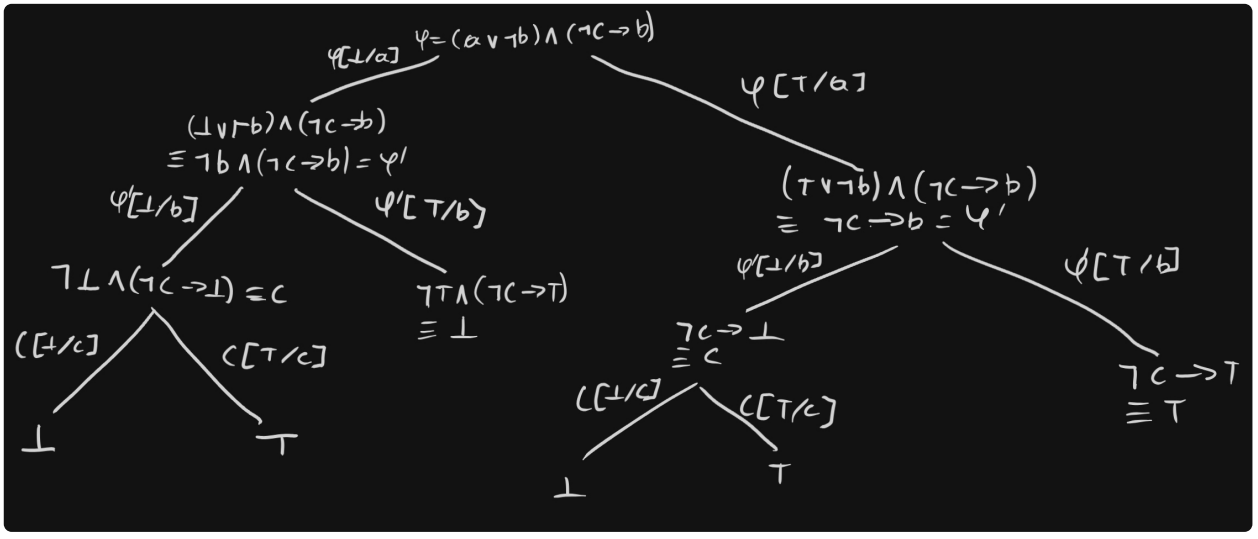
  On détermine $\varphi[\bot/x]$. On élimine les constantes logiques. On recommence l'algorithme avec la formule ainsi obtenue.
  - 2ème branche :

    On détermine $\varphi[\top/x]$. On élimine les constantes. On recommence.

On s'arrête quand la formule est $\top$ ou $\bot$ (aucune variable, feuilles).

Règles d'élimination des constantes logiques de Quine :

$\neg\bot \equiv \top$
$\neg\top \equiv \bot$
$\top \to \varphi \equiv \varphi$
$\varphi \to \top \equiv \top$
$\bot \to \varphi \equiv \bot$
$\varphi \to \bot \equiv \varphi$
$\top \vee \varphi \equiv \top$
$\varphi \vee \top \equiv \top$
$\bot \vee \varphi \equiv \varphi$
$\varphi \vee \bot \equiv \varphi$
$\top \wedge \varphi \equiv \varphi$
$\varphi \wedge \top \equiv \varphi$
$\bot \wedge \varphi \equiv \bot$
$\varphi \wedge \bot \equiv \bot$
$\top \leftrightarrow \varphi \equiv \varphi$
$\varphi \leftrightarrow \top \equiv \varphi$
$\bot \leftrightarrow \varphi \equiv \neg\varphi$
$\varphi \leftrightarrow \bot \equiv \neg\varphi$



Proposition :

- Toutes les feuilles sont à $\bot \Leftrightarrow \varphi$ antilogie.
- Au moins une feuille est $\top \Leftrightarrow \varphi$ satisfiable.
- Toutes les feuilles sont $\top \Leftrightarrow \varphi$ tautologie.

Se montre par induction structurelle.

Terminaison :

- à chaque substitution, le nombre de variables propositionnelles diminue strictement (de 1 variable).
- à chaque élimination, la taille de la formule diminue strictement (on vérifie les règles une par une).
- On prend l'ordre lexicographique (nombre de variables, taille de la formule)
  - On vérifie aisément la stricte décroissance.
  - ensemble bien fondé

donc termine.

c. Réduction d'un problème à SAT

Méthodologie :

Exemple : Est-il possible de colorer un graphe G en utilisant au plus K colonnes ?

Etape 1 :

On introduit toutes les variables propositionnelles.

Exemple : On a besoin de $|S| \times k$ variables. $x_{i,c}$ signifiant le sommet i est colorié avec la couleur c.

Etape 2 :

On établit la liste des conditions à respecter pour résoudre le problème.

Exemple :

- Chaque sommet est coloré avec 1 couleur.
- 2 sommets adjacents ne doivent pas avoir la même couleur.

Etape 3 :

Transformer chaque condition en une formule propositionnelle.

Etape 4 :

On résout SAT(φ) avec φ la conjonction de toutes les formules obtenues pour les conditions.