Logique

Arnaud Durand et Pierre Gervais

November 25, 2016

Contents

Part I
Calcul propositionnel

1 Syntaxe

Le *calcul propositionnel* est un langage *inductivement* et *librement engendré* par un ensemble de règles.
C'est à dire qu'une formule ne peut pas être obtenu de deux façons différentes.

Définition 1. Soit $\mathcal{P}$ un ensemble de constantes propositionnelles, on définit $\mathcal{F}_\mathcal{P}$ le calcul propositionnel sur $\mathcal{P}$ obtenu par les règles suivantes :

- si $p \in \mathcal{P}$, alors $p \in \mathcal{F}_\mathcal{P}$
- $\perp \in \mathcal{F}_\mathcal{P}$
- si $F \in \mathcal{F}_\mathcal{P}$, alors $(\neg F) \in \mathcal{F}_\mathcal{P}$
- si $F, G \in \mathcal{F}_\mathcal{P}$ alors $(F \vee G), (F \wedge G), (F \to G) \in \mathcal{F}_\mathcal{P}$

Notation 1. S'il n'y a pas d'ambiguïté, on notera $\mathcal{F}_\mathcal{P} = \mathcal{F}$

Définition 2. Une définition alternative de $\mathcal{F}$ est $\mathcal{F} = \bigcup_{n \geqslant 0} \mathcal{F}_n$ où

- $\mathcal{F}_0 = \mathcal{P}$
- $\mathcal{F}_{n+1} = \mathcal{F}_n \cup \{(\neg F) \mid F \in \mathcal{F}_n)\} \cup \{(F \star G) \mid F, G \in \mathcal{F}_n,\ \star \in \{\wedge, \vee, \to\}\}$, avec $n \geqslant 0$

On définit la *hauteur* d'une formule F par le plus petit n tel que $F \in \mathcal{F}_n$.

Remarque 1. Ce langage est fortement parenthésé et toute formule peut être représentée par un arbre de décomposition.

Propriété 1. *Propriété de lecture unique*
Pour tout $F \in \mathcal{F}$, un seul de ces cas est vrai :

1. $F \in \mathcal{P}$
2. *Il existe un unique $G \in \mathcal{F}$ tel que $F = (\neg G)$*
3. *Il existe d'uniques $G, H \in \mathcal{F}$ et $\star \in \{\vee, \wedge, \to\}$ tels que $F = G \star H$*

C'est-à-dire que toute formule ne peut se décomposer que d'une seule façon.

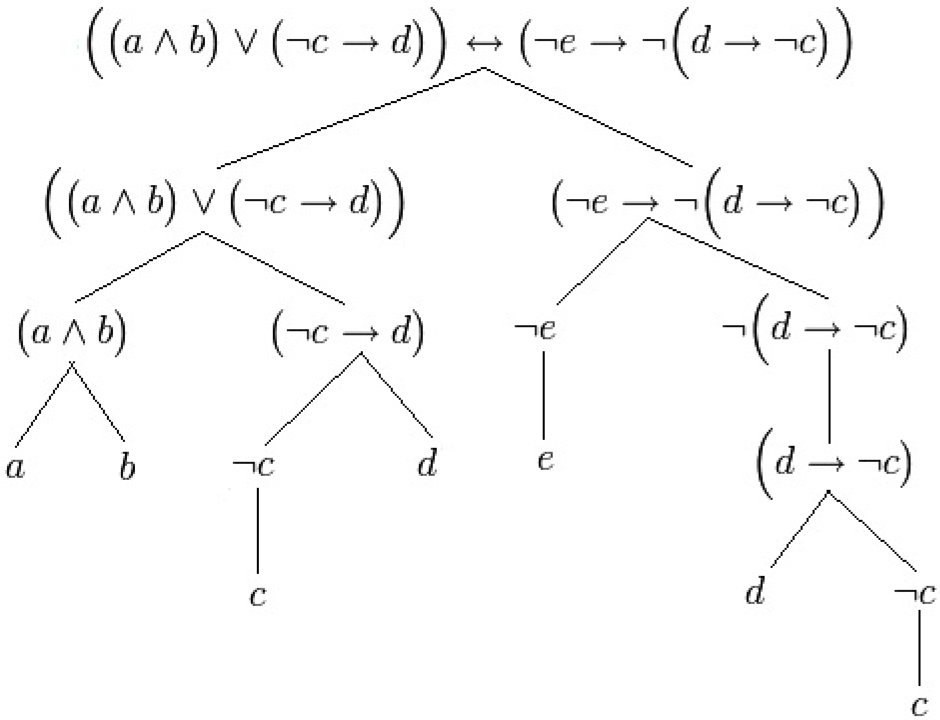


Figure 1: Arbre de décomposition

1.1 Raisonnements

On démontrera généralement les propriétés s'appliquant à $\mathcal{F}$ par induction : pour démontrer une proposition A s'appliquant à $\mathcal{F}$, on la démontre sur $\mathcal{P}$ et pour tout $(F \star G)$ et $(\neg F)$ où on suppose que $F, G \in \mathcal{F}$ vérifient A et $\star \in \{\wedge, \vee, \rightarrow\}$.

1.2 Définition alternative de $\mathcal{F}_{\mathcal{P}}$

Soit $\Sigma = \mathcal{P} \cup \{(,), \neg, \wedge, \vee, \rightarrow, \perp\}$, Σ^* est l'ensemble des mots sur Σ.

Exemple 1.

- $F = (\wedge\neg x_1)((\in \Sigma^*$
- $F = (\neg x_1) \in \Sigma^*$

Définition 3. $\mathcal{F}$ est le plus petit sous-ensemble de Σ^* contenant $\mathcal{P} \cup \{\perp\}$ et **clos** par les opérations

1. $(F, G) \longmapsto (F \vee G)$
2. $(F, G) \longmapsto (F \wedge G)$

3. $(F, G) \longmapsto (F \rightarrow G)$

Remarque 2. On peut montrer que les deux définitions correspondent. $\mathcal{F}$ satisfait la propriété de lecture unique (voir TD).

1.2.1 Sous-formule, hauteur, arbre de décomposition

Définition 4. Soit $F \in \mathcal{F}$, on définit $\mathcal{S}(F)$ l'ensemble des *sous-formules de* F telles que

- si $F \in \mathcal{P}$, $\mathcal{S}(F) = \{F\}$
- si $F = (\neg G)$ alors $\mathcal{S}(F) = \{F\} \cup \mathcal{S}(G)$
- si $F = (G \star H)$ où $\star \in \{\wedge, \vee, \rightarrow\}$, alors $\mathcal{S}(F) = \{F\} \cup \mathcal{S}(G) \cup \mathcal{S}(H)$

TODO : vérifier dernier point

Définition 5. Soit $F \in \mathcal{F}$ on définit la *hauteur* $h(F)$ de F par

- $h(F) = 0$, si $F \in \mathcal{P}$
- si $= (\neg G)$, alors $h(F) = 1 + h(G)$
- si $F = (G \star H)$, alors $h(F) = 1 + \max\{h(G), h(H)\}$

Définition 6. Soit $F \in \mathcal{F}$, l'*arbre de décomposition de* F $arb(F)$ est un graphe étiqueté défini par

1. si $F \in \mathcal{P}$, $arb(F)$ est réduit à un sommet étiqueté par F.
2. si $F = (\neg G)$, alors $arb(F) = \neg - arb(G)$
3. si $F = (G \star H)$, alors $arb(F) = G - \star - H$

Notation 2. Soit F une formule, $var(F)$ est l'ensemble des variables de F, $occ(F)$ est le multi-ensemble des variables de F et $arb(F)$ est le graphe

- dont les sommets sont V
- et muni d'une fonction d'étiquetage $\lambda : V \longrightarrow \{\neg, \bot, \vee, \wedge, \rightarrow\} \cup var(F)$.

Remarque 3. Toutes les définitions sont univoques par la propriété de lecture unique.

Remarque 4. On définit la hauteur d'une formule par la hauteur de son arbre de décomposition, c'est-à-dire la distance maximum entre les feuilles et la racine.

Notation 3.

- $\top$ comme abréviation pour $(\bot \rightarrow \bot)$
- $(p \longleftrightarrow q)$ pour $(p \leftarrow q) \wedge (p \rightarrow q)$
- $\bigwedge_{i=1}^{n} A_i = (((A_1 \wedge A_2) \wedge A_3) ... \wedge A_n)$

2 Sémantique

On s'intéresse à des propositions dont la valeur de vérité est soit vrai soit faux. On a besoin d'une **interprétation** (en terme de vrai ou faux) de ces constantes propositionnelles.

Définition 7. Une *valuation* est une fonction $v \ : \ \mathcal{P} \longrightarrow \{0,1\}$. Étant donné une valuation v, on définit l'*interprétation* $\overline{v} \ : \ \mathcal{F} \longrightarrow \{0,1\}$ comme ceci

- si $F = p \in \mathcal{P}$ alors $\overline{v} = v(p)$
- si $F = (\neg G) \in \mathcal{P}$ alors $\overline{v}(F) = 1$ si et seulement si $\overline{v}(G) = 0$
- $\overline{v}(\bot) = 0$
- $\overline{v}(F \wedge G) = 1$ si et seulement si $\overline{v}(F) = \overline{v}(G) = 1$

On peut décrire l'interprétation d'une formule par sa *table de vérité* :

F	G	$\neg G$	$F \wedge G$	$F \rightarrow G$
0	0	1	0	1
0	1	1	0	1

On définit formellement la *table de vérité* par une fonction $v \ : \ \{0,1\}^{\mathcal{P}} \longrightarrow \{0,1\}$

Définition 8.

- $F \in \mathcal{F}$ est dit *satisfaisable* s'il existe une valuation v de $\mathcal{P}$ tel que $\overline{v}(F) = 1$
- F est dit *valide* si pour toute valuation v de $\mathcal{P}$, $\overline{v}(F) = 1$, on dit aussi que F est une *tautologie*.
- F et G sont dites *équivalentes*, notées $F \equiv G$, si pour toute valuation v, $\overline{v}(F) = \overline{v}(G)$
- On note $v \models F \iff \overline{v}(F) = 1$, c'est à dire si et seulement si v satisfait F.

Exercice 1. Vérifier que $F \equiv G$ si et seulement si $F \leftrightarrow G$ est valide.

Proposition 1. *Pour tout $F \in \mathcal{F}$, F est satisfaisable si et seulement si $(\neg F)$ n'est pas valide.*

3 Exemples de formalisation

3.1 Contraites de compatibilité/exclusion

Problème : On possède n produits chimiques à ranger dans $k \leqslant n$ conteneurs. Certains produits ne peuvent pas être stockés ensemble dans un conteneur.

La contrainte est donnée sous la forme d'un ensemble $\mathcal{L} \subseteq [n]$ tel que $I = \{i_1, ..., i_k\} \subseteq \mathcal{L}$ si et seulement si les produits $i_1, .., i_k$ ne peuvent pas être stockés ensemble.

Enjeu : Écrire une formule propositionnelle F telle que F est satisfaisable si le problème a une solution.

Les variables propositionnelles $\mathcal{P} = p(i,j),\ i \leqslant n,\ j \leqslant k$ sont interprétées par "le produit chimique i est dans le camion j".

On exprime deux propositions :

- Chaque produit se trouve dans un unique conteneur : $F = \underbrace{\left(\bigwedge_{i\leqslant n}\left(\bigvee_{j\leqslant k} p(i,j)\right)\right)}_{\substack{\text{Chaque produit } i \text{ est stocké} \\ \text{dans au moins un camion } j}} \wedge \underbrace{\left(\bigwedge_{i\leqslant n}\left(\bigwedge_{\substack{j,j'\leqslant k\\ j\neq j'}}(\neg\,(p(i,j)\wedge p(i,j')))\right)\right)}_{\substack{\text{Pour chaque produit } i \\ \text{et chaque paire de camions } j\neq j' \\ \text{il est faux que } i \text{ est à la fois dans } j \text{ et } j'}}$

- On respecte les incompatibilités : $G = \underbrace{\bigwedge_{I\subseteq\mathcal{L}}\left(\bigwedge_{j\leqslant k}\neg\left(\bigwedge_{i\in I} p(i,j)\right)\right)}_{\substack{\text{Pour chaque ensemble } I \text{ de produits} \\ \text{ne pouvant pas être stockés ensemble} \\ \text{et pour chaque camion } j, \\ \text{aucun produit de } I \text{ n'est présent dans le camion}}}$

4 Équivalence logique usuelles

Proposition 2. *Substitution*

Soient $H_1, ..., H_n \in \mathcal{F}$.

- *Si F est une tautologie, la formule $F' = F[H_1/p_1, ..., H_n/p_n]$ est une tautologie, où F' est la formule dans laquelle on remplace chaque p_i par H_i.*

- *Si $F \equiv G$ alors $F[H_1/p_1, ..., H_n/p_n] \equiv G[H_1/p_1, ..., H_n/p_n]$*

Exemple 2. Soient $F = (p_1 \longrightarrow p_1)$ et $H = ((q_1 \wedge \neg q_3) \vee q_3)$

Si F est une tautologie, alors $(((q_1 \wedge \neg q_3) \vee q_3) \longrightarrow ((q_1 \wedge \neg q_3) \vee q_3))$ est une tautologie.

Remarque 5. La réciproque est fausse.

Lemme 1. *Soit $\mathcal{P} = \{p_1, ..., p_n\}$, $F \in \mathcal{F}$ et $H_1, ..., H_n \in \mathcal{F}$.*

Soit v une valuation de $\mathcal{P}$ avec $\forall i,\ v(H_i) = \delta_i$.

Alors la valuation v' définie par $v'(p_i) = \delta_i$ vérifie $\overline{v}(F[H_1/p_1, ..., H_n/p_n]) = \overline{v'}(F)$

Preuve 1. Démontrons le lemme par induction structurelle sur F.

On notera $F[H_1/p_1, ..., H_n/p_n] := F[\overline{H}/\overline{p}]$

- Si $F = \perp$, alors $v(\perp) = v'(\perp) = 0$.

- Si $F = p_1 \in \mathcal{P}$, alors $F' = F[H_1/p_1] = H_1$ <u>et</u> $v'(p_1) = \delta_1 = v(H_1)$.

- Si $F = \neg G$, par hypothèse d'induction $v(G[\overline{H}/\overline{p}]) = v'(G)$.

  $v(F[\overline{H}/\overline{p}]) = 1$

  $v(G[\overline{H}/\overline{p}]) = 1 - v'(G) = v'(F)$

- Si $F = G_1 \wedge G_2$, par hypothèse d'induction

$$\begin{cases} v(G_1[\overline{H}/\overline{p}]) = v'(G_1) \\ v(G_2[\overline{H}/\overline{p}]) = v'(G_2) \end{cases}$$

or $v(F[\overline{H}/\overline{p}]) = v(G_1[\overline{H}/\overline{p}]) \cdot v(G_2[\overline{H}/\overline{p}]) = v'(G_1) \cdot v'(G_2) = v'(F)$

- etc.

$\square$

Preuve 2. Preuve de la proposition Comme $F \equiv G$ si et seulement si $(F \longleftrightarrow G)$ est une tautologie.

On prouve seulement seulement la première partie de la proposition.

Supposons F ue tautologie, soit v une valuation de $\mathcal{P}$, par le lemme précédent il existe v' telle que $v(F[\overline{H}/\overline{p}]) = v'(F)$ comme F est une tautologie, $v'(F) = 1$ et $v(F[\overline{H}/\overline{p}])$ est une tautologie.

$\square$

Par la suite, pour toutes formules propositionnelles A, B et C, les équivalences suivantes se montreront en substituant des variables aux formules.

Propriété 2.

- *Négation :* $\neg\neg A \equiv A$
- *Lois de Morgan :* $\neg(A \wedge B) \equiv \neg A \vee \neg B$, $\neg(A \vee B) \equiv \neg A \wedge \neg B$, $\neg(A \longrightarrow B) \equiv A \wedge \neg B$
- *Associativité :* $(A \wedge (B \wedge C)) \equiv ((A \wedge B) \wedge C)$ *et* $(A \vee (B \vee C)) \equiv ((A \vee B) \vee C)$
- *Expression des connecteurs :* $\bot \equiv (A \wedge \neg A)$ *et* $\top \equiv (A \vee \neg A)$
- *Distributivité :* $(A \vee (B \wedge C)) \equiv ((A \vee B) \wedge (A \vee C)$
- *Idempotence :* $A \vee A \equiv A \wedge A \equiv A$
- *Absorption :* $(A \wedge \bot)\bot$ *et* $(A \vee \top) \equiv \top$
- *Neutre* $(A \wedge \top) \equiv A$ *et* $(A \vee \bot) \equiv A$

5 Formules normales

Ici $\mathcal{P} = \{x_1, ..., x_n\}$

Définition 9.

- Un *littéral* est une variable ou une négation de variable.
- Une *clause* (ou *disjonction élémentaire*)est une formule C de la forme $C = \bigvee_{i \in A} x_i \vee \bigvee_{i \in B} \neg x_i$ où $A, B \subseteq \{1, ..., n\}$.
- La longueur d'une clause C notée $|C|$ est son nombre de variables : $|C| = |A| + |B|$.
- Si $|C| = 1$, C est une *clause unitaire*.

Définition 10.

- F est sous *forme normale disjonctive* (*FND* ou *DNF*) si F est une disjonction de conjonctions élémentaires, c'est à dire si F est de la forme

$$F = \bigvee_{i=1}^{m} \left(\bigwedge_{j \in A_i} x_j \wedge \bigwedge_{j \in B_i} \neg x_j \right)$$

- F est sous *forme normale conjonctive* (*FNC* ou *CNF*) si F est une conjonction de disjonctions élémentaires, c'est à dire si F est de la forme

$$F = \bigwedge_{i=1}^{m} \left(\bigvee_{j \in A_i} x_j \wedge \bigvee_{j \in B_i} \neg x_j \right)$$

- Si la longueur de chaque clause est bornée par $l > 0$, on dit que F est l-FNC (de même pour l-FND).

Définition 11. Soit F sous forme FNC, c'est à dire si $F = \bigwedge\limits_{i \in I} C_i$ où $(C_i)_{i \in I}$ sont des clauses, la longueur $|F|$ de F est :

$$|F| = \sum_{i \in I} |C_i|$$

Exemple 3.

- $F = (p \wedge q \neg r) \vee (\neg p \wedge r) \vee \neg r$ est une 3-FND.
- $F = p \vee \neg q$ est une 2-FNC à une clause et une 1-FND à deux clauses unitaires.

Remarque 6. Le nombre de clauses différentes possibles est borné : $|I| \leqslant 2^{|\mathcal{P}|}$

5.1 Tables de vérité et formes normales

Soit v une valuation de $\mathcal{P}$, on note :

$$C_v^1 = \left(\bigwedge_{i \in A_1} x_i \right) \wedge \left(\bigwedge_{i \in B_1} \neg x_i \right)$$

où $\forall x_i \in \mathcal{P}, \begin{cases} v(x_i) = 1 \Longleftrightarrow i \in A \\ v(x_i) = 0 \Longleftrightarrow i \in B \end{cases}$.
On a $A_1 \sqcup B_1 = \mathcal{P}$.

Lemme 2. *Soit F une formule, F est équivalente à la FND suivante :*

$$\bigvee_{\substack{v \,:\, \mathcal{P} \longrightarrow \{0,1\} \\ \overline{v}(F) = 1}} C_v^1$$

On note à présent :

$$C_v^0 = \left(\bigvee_{i \in B} x_i \right) \vee \left(\bigvee_{i \in A} \neg x_i \right)$$

Lemme 3. *Soit F une formule, F est équivalente à la FNC suivante :*

$$\bigwedge_{\substack{v\ :\ \mathcal{P}\longrightarrow\{0,1\}\\ \overline{v}(F)=0}} C_v^0$$

5.2 Systèmes complets de connecteurs

Définition 12. Un système de connecteurs est fonctionnellement complet si toute table de vérité est représentable ar une formule utilisant ces seuls connecteurs.

Corollaire 1. $\{\wedge,\vee,\neg\}$ *est fonctionnellement complet, de même pour* $\{\wedge,\neg\}$*,* $\{\vee,\neg\}$*,* $\{\rightarrow,\neg\}$ *et* $\{\perp,\rightarrow\}$*.*

5.3 Mise sous forme normale

On cherche comment construire une forme normale équivalente à une formule F donnée, il faut :

- construire la table de vérité de F
- en déduire une FND ou une FNC

Cette méthode est très coûteuse car elle oblige à en passer par l'examen de toutes les valuations.
Une autre méthode consiste à jouer avec les équivalences pour trouver une FND ou FNC.

6 Formes normales complètes et bornes inférieures de FND

Définition 13. Soient F une formule sur $\mathcal{P}$ et C une conjonction élémentaire.
On dit que C est *impliquant* de F (ou que F absorbe C) si $(C \rightarrow F)$ est valide.

Remarque 7. $(C \rightarrow F)$ est valide si et seulement si :

$$\forall v\ :\ \mathcal{P} \longrightarrow \{0,1\},\ (\overline{v}(C) = 1 \Longrightarrow \overline{v}(F) = 1)$$

Le terme d'absorption est justifié par le fait que $(C \rightarrow F)$ soit valide si et seulement si $C \vee F \equiv F$.

Définition 14. Soit C_1 une conjonction élémentaire, C_1 est un impliquant premier de F si :

- C_1 est un impliquant de F
- aucun impliquant C_2 de F n'absorbe C_1, c'est-à-dire : pour toute conjonction élémentaire C_2, si $(C_2 \rightarrow F)$ est valide alors $(C_1 \rightarrow C_2)$ n'est pas valide (c'est à dire $C_1 \vee C_2 \not\equiv C_2$).

En résumé, C_1 est un impliquant de C_2 si et seulement si $(C_1 \rightarrow C_2)$ est valide, si et seulement si $(C_1 \vee C_2 \equiv C_2)$, et si et seulement si C_2 absorbe C_1.

Exemple 4. $C_2 = (x_1 \wedge \neg x_2 \wedge \neg x_3)$ est un impliquant de $C_1 = (x_1 \wedge \neg x_2)$ car pour toute valuation v, si $\overline{v}(C_2) = 1$, alors $\begin{cases} v(x_1) = 1 \\ v(x_2) = v(x_2) = 0 \end{cases}$ et donc $\overline{v}(C_2) = 1$.
C_2 est "plus général" que C_1.

Lemme 4. *Soient* $C_1 = \bigwedge_{i \in A_1} x_i \wedge \bigwedge_{i \in B_1} \neg x_i$ *et* $C_2 = \bigwedge_{i \in A_2} x_i \wedge \bigwedge_{i \in B_2} \neg x_i$

C_1 est un impliquant de C_2 (C_2 absorbe C_1) si et seulement si $A_2 \subseteq A_1$ et $B_2 \subseteq B_1$.

Exercice 2. Le démontrer.

Exemple 5. $C_1 = x_1 \wedge x_2 \wedge \neg x_2 \wedge \neg x_4$ et $C_2 = x_1 \wedge \neg x_3$

C_2 absorbe C_1.

Lemme 5. *Soient* $C = \bigwedge_{i \in A} l_i$ *où* $(l_i)_i$ *sont des littéraux et* F *est une formule.*

C est un impliquant premier de F si et seulement si :

- *C est un impliquant de F*
- *Pour tout $j \in A$, on a que $C_j = \bigwedge_{\substack{i \in A \\ i \neq j}} l_i$ n'est pas un impliquant de F.*

Exercice 3. Le démontrer.

Remarque 8. Soit $F = \bigvee_{i \in I} D_i$, chaque D_i est un impliquant de F.

Proposition 3. *Toute formule propositionnelle peut être mise sous forme normale disjonctive dont toutes les clauses élémentaires sont les impliquants premiers de F. Une telle FND est dite complète.*

Preuve 3. Soit n le nombre de variables, à équivalence logique près il n'existe qu'un nombre fini de conjonctions élémentaire (au plus 3^n) donc d'impliquants premiers de F.

Par exemple : $\mathcal{P} = \{x_1, x_2, x_3\}$,chaque variable peut apparaître ou pas dans la clause.
Si elle apparaît, cela peut être positivement ou négativement.
Soient $(D_i)_{i \leqslant m}$ des impliquants premiers de F.

- F absorbe chacun de ses impliquants
- F est équivalente à une FND : $\bigvee_{i \in A} C_i$

donc par associativité de la disjonction :

$$F \equiv \bigvee_{i \in A} C_i \vee \bigvee_{i=1}^{m} D_i$$

Pour tout j, par absorption, $\bigvee_{i \in A} C_i \vee D_j \equiv F$.

Chaque C_i est un impliquant :

- Soit C_i est premier
- Soit C_i est absorbé par un impliquant premier D_j (c'est à dire $C_i \vee D_j \equiv D_j$).

Par associativité et absorptions successives, on a :

$$F = \bigvee_{i=1}^{m} D_i$$

Notation 4. Par abus de langage, on appelle les modèles de F l'ensemble des valuations S_F satisfaisant F.

Si C est un impliquant de F, $S_C \subseteq S_F$ et si C est un impliquant premier de F, S_C est maximal au sens de l'inclusion.

Définition 15. Soient $F \in \mathcal{F}_{\mathcal{P}}$, $F = \bigvee\limits_{i \in I} C_i$ et C_i des conjonctions élémentaires.

- F est *première* si chaque C_i est un impliquant premier.
- F est *redondante* s'il existe $j \in I$ tel que $F \equiv \bigvee\limits_{\substack{i \in I \\ i \neq j}} C_i$
- F est *non-redondante* si pour tout $j \in I$, $F \not\equiv \bigvee\limits_{\substack{i \in I \\ i \neq j}} C_i$.

La redondance est une généralisation de la notion d'absorption : si C_j vérifie $F \equiv \bigvee\limits_{\substack{i \in I \\ i \neq j}} C_i$, alors C_j est absorbée par $\bigvee\limits_{\substack{i \in I \\ i \neq j}} C_i$

Exemple 6. $C_1 = x \wedge y$, $C_2 = x \wedge \neg y$, $C_3 = x$ et $F = \bigvee\limits_{i=1}^{3} C_i$, on a $F \equiv \bigvee\limits_{i=1}^{2} C_i \equiv C_3$ car C_1 et C_2 sont des impliquants de C_3.

Lemme 6. *Si F est complète, alors F est première, mais l'inverse n'est pas vrai si la forme complète est redondante.*

Ces notion sont utiles pour caractériser des FND (de longueurs) minimales de certaines formules.

6.1 Formules propositionnelles positives

Définition 16. Soit $\mathcal{P}$ un ensemble de variables propositionnelles, on définit l'ordre $\leqslant$ sur les valuations par

$$\forall v_1, v_2 \ : \ \mathcal{P} \longrightarrow \{0,1\}, \ v_1 \leqslant v_2 \Longleftrightarrow \forall x \in \mathcal{P}, v_1(x) \leqslant v_2(x)$$

Définition 17. Une formule F est *positive* si pour tout valuation v_1, v_2 telles que $v_1 \leqslant v_2$ on a $\overline{v_1}(F) \leqslant \overline{v_2}(F)$.

Proposition 4. *Soit F une formule, elle est positive si et seulement si tous les impliquants premiers de F sont positifs, c'est-à-dire si elle ne contient aucun littéral négatif.*

Remarque 9. Si une formule est positive alors elle peut être écrite sans utiliser de négation.

Preuve 4.

F positive $\Longrightarrow$ F tous les littéraux de F sont positifs

Soit F une formule positive, et C un impliquant premier de F contenant un littéral négatif.

$$C = \bigwedge_{i \in A} x_i \wedge \bigwedge_{i \in B} \neg x_i, \ B \neq \emptyset$$

$C' = \bigwedge\limits_{i \in A} x_i$ absorbe C, or C est un impliquant premier, donc C ne peut pas être absorbé par un autre impliquant de F.

Alors C' n'est pas un impliquant de F, d'où l'existence d'une valuation v satisfaisant C' mais pas F.

Si $\overline{v}(C') = 1$ alors $\forall i \in 1,\ v(x_i) = 1$ et $\forall i \in B,\ v(x_i) \in \{0, 1\}$.
C est un impliquant de F donc pour toute valuation v', si $\overline{v'}(C) = 1$ alors $\overline{v'}(F) = 1$, par exemple :

$$\begin{cases} v'(x_i) = 1,\ \forall i \in A \\ v'(x_i) = 0,\ \forall i \in B \end{cases}$$

On a donc :

$$\begin{cases} \overline{v}(F) = 0,\ \forall i \in A \\ \overline{v'}(F) = 1,\ \forall i \in B \\ v' \leqslant v \end{cases}$$

On sait que F est positive alors $1 = v'(F) \leqslant v(F) = 0$, contradiction. ✓

F tous les littéraux de F sont positifs $\Longrightarrow$ F positive

On considère une forme (par exemple complète) de F :

$$F \equiv \bigvee_{ii \in I} C_i$$

où $(c_i)_i$ sont des impliquants premiers de F.
Par hypothèse chaque C_i est positif, on doit montrer que F est positive, c'est-à-dire que

$$\forall v, v'\ :\ (v \leqslant v' \Longrightarrow \overline{v}(F) \leqslant \overline{v'}(F))$$

Soit v une valuation, on pose $A_v = \{i | v(x_i) = 0\}$.
Par définition, pour tout j on a

$$\overline{v}(C_j) = 0 \Longleftrightarrow C_j \text{ ne contient quue des variables de } A_v$$

Soit v' une autre valuation telle que $v \leqslant v'$, on a $A_{v'} \subseteq A_v$, donc $\{C_j \mid \overline{v}(C_j) = 1\} \subseteq \{C_j \mid \overline{v'}(C_j) = 1\}$, d'où :

$$\overline{v}(F) \leqslant \overline{v'}(F)$$

✓

Proposition 5. *Soit F une formule positive, alors la FND complète de F est positive et non-redondante. Elle est l'unique FND première de F.*

Remarque 10. Il n'existe pas de forme équivalente plus courte : aucun impliquant premier n'est redondant.

Preuve 5. Soit F une formule positive de variables $x_1, ..., x_n$.
Par les propositions précédentes, la FND complète $\bigvee_{i \in I} C_i$ de F n'a que des impliquants premiers positifs.

Supposons parl'absurde qu'il existe $j \in I$ telle que C_j soit superflue (que F soit redondante), c'est à dire que $F \equiv \bigvee_{\substack{i \in I \\ i \neq j}} C_i$.

On pose $C_j = \bigwedge_{a \in A} x_a$, considérons la valuation v telle que $v(x_a) = 1 \Longleftrightarrow a \in A$. On a $\overline{v}(C_j) = 1$ donc $\overline{v}(F) = 1$.

Or $F \equiv \bigvee_{\substack{i \in I \\ i \neq j}} C_i$, donc il existe $i \neq j$ tel que $\overline{v}(C_i) = 1$.

Posons à présent $C_i = \bigwedge_{b \in B} x_b$, on sait que C_i et C_j sont des impliquants premiers de F

Donc C_i (resp. C_j) n'absorbe pas C_j (resp. C_i), d'après une proposition précédente $A \not\subseteq B$ et $B \not\subseteq A$, il existe alors $b \in B$ tel que $b \notin A$, donc $v(b) = 0$ et $\overline{v}(C_i) = 0$, ce qui est contradictoire.

Part II
La méthode de résolution

7 Notion d'équisatisfaisabilité

Problème 1 : Déterminer si une formule en FNC est satisfaisable

Problème 2 : Mettre une formule sous forme FNC en préservant l'équivalence.

Première stratégie pour le premier problème : Construire une table de vérité : 2^n lignes où n est le nombre de variables.

Deuxième stratégie : Construire la FND de la formule : $F = \bigvee_{i \in I} D_i$ avec $D_i = p_i^1 \wedge p_i^2 ... \wedge p_i^k$.

F est satisfaisable si et seulement s'il existe un $i \in I$ tel que D_i soit satisfaisable, ce qui est facile à tester : D_i est satisfaisable si on n'a pour aucune variable $p \wedge ... \neg p$.

On cherche seulement à savoir si une formule G est satisfaisable, on n'a pas besoin de construire une formule équivalente, on peut se contenter de construire une formule *équisatisfaisable*.

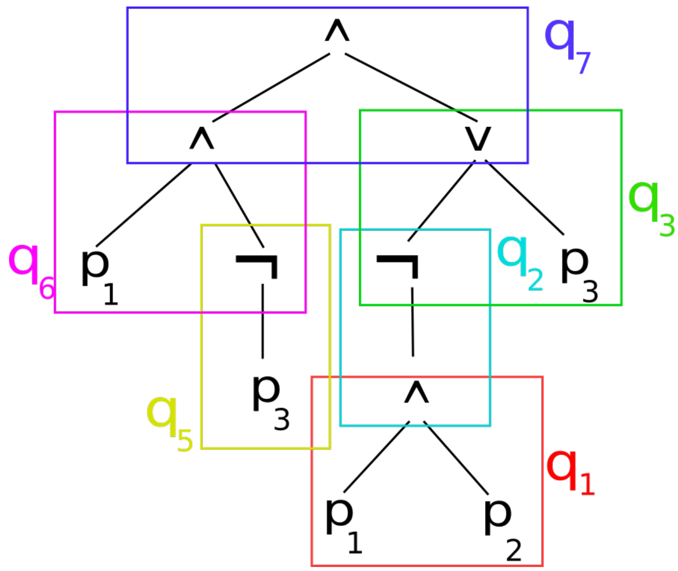
Définition 18. $F \in \mathcal{F}_\mathcal{P}$ et $G \in \mathcal{F}_\mathcal{Q}$ sont équisatisfaisables si elles sont toutes les deux satisfaisables ou non-satisfaisables.

Exemple 7. $F = x \vee y$, avec $\mathcal{P} = \{x, y\}$ et $F = ((x \vee y) \leftrightarrow p) \wedge p$, avec $\mathcal{Q} = \{x, y, p\}$

Proposition 6. *Soit $F \in \mathcal{F}_\mathcal{P}$, il existe $\mathcal{Q}$, $G \in \mathcal{F}_\mathcal{Q}$ avec $G = \bigwedge_{i \in I} C_i$ en FNC tels que F et G soient équisatisfaisables et*

- $|\mathcal{Q}| \leqslant |SF(F)|$
- $|I| \leqslant |SF(F)| + 1$
- $\forall i \in I,\ |C_i| \leqslant 3$

Preuve 6. On le montre sur un exemple.



$\mathcal{P} = \{p_1, p_2, p_3\}, \mathcal{Q} = \mathcal{P} \cup \{q_1, ...q_7\}$
On pose $G \in \mathcal{F}_\mathcal{Q}$ avec : $G =$

$$
\begin{array}{l}
(q_1 \leftrightarrow (p_1 \wedge p_2)) \\
\wedge(q_2 \leftrightarrow q_1) \\
\wedge(q_3 \leftrightarrow (q_2 \wedge p_3)) \\
\wedge(q_5 \leftrightarrow \neg p_3) \\
\wedge(q_6 \leftrightarrow (p_1 \wedge q_5)) \\
\wedge(q_7 \leftrightarrow (q_6 \wedge q_3)) \\
\wedge q_7
\end{array}
$$

On transforme chaque pseudo-clause $(q_i \leftrightarrow (q_j \vee q_k))$ en une conjonction de 3-clauses équivalentes :

$$(q_i \leftrightarrow (q_j \vee q_k))$$

$$(q_i \rightarrow (q_j \vee q_k)) \wedge ((q_j \vee q_k) \rightarrow q_i)$$

$$(\neg q_j \vee q_i) \wedge (\neg q_k \vee q_i) \wedge (\neg q_i \vee q_j \vee q_k)$$

8 Méthode de résolution

On va montrer qu'une formule n'est pas satisfaisable, on va utiliser une méthode de réfutation.
en considérant $p, \neg p \notin C_1 \vee C_2$, on n'utilisera qu'une seule règle de déduction :
On notera la règle de déduction "de $(C_1 \vee p)$ et $(C_2 \vee \neg p)$ on déduit $(C_1 \vee C_2)$" par

$$\frac{(C_1 \vee p) \quad (C_2 \vee \neg p)}{C_1 \vee C_2}$$

Cette règle préserve la satisfaisabilité :
Soit v satisfaisant $(C_1 \vee p)$ et $(C_2 \vee \neg p)$, alors $\overline{v}(C_1) = 1$ ou $\overline{v}(p) = 1$

- Si v satisfait C_1 alors v satisfait $C_1 \vee C_2$

- Si v satisfait p alors $\overline{v}(\neg p) = 0$ et nécessairement $\overline{C_2} = 1$, d'où $\overline{v}(C_1 \vee C_2) = 1$.

Remarque 11. $(C_1 \vee p) \wedge (C_2 \vee \neg p)$ est satisfaisable si et seulement si $(C_1 \vee p) \wedge (C_2 \vee \neg p) \wedge (C_1 \vee C_2)$

Définition 19. $C_1 \vee C_2$ est appelé le *résolvant* de $C_1 \vee p$ et $C_2 \vee \neg p$.

Idée : D'une formule $\varphi = \bigwedge_{i=1}^{n} C_i$ et une variable $p \in \mathcal{P}$, on construit $\varphi' = \varphi \wedge (C'_i \vee C'_j)$ avec $C'_i = C_i \vee p$ et $C'_j = C_j \vee \neg p$, jusqu'à ce qu'on ne puisse plus appliquer la méthode ou jusqu'à ce qu'on ait produit la clause vide $\Box$ sachant que pour toute variable p, on a :

$$\frac{p \quad \neg p}{\Box}$$

NB : pour toute valuation v, $v(\Box) = 0$.

Définition 20. Une preuve de réfutation par résolution de $\varphi = \bigwedge_{i=1}^{n} C_i$ (avec φ en FNC) est une suite $r_1, ...r_m$ telle que pour tout $i \leqslant m$

- $r_i \in \{C_1, ...C_n\}$

- r_i se déduit par résolution de r_j et r_k avec $j, k < i$

et $r_m = \Box$

Exemple 8. $\varphi = (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee q) \wedge (\neg p \vee \neg q)$

1. r_1 : $p \vee q$
2. r_2 : $p \vee \neg q$
3. r_3 : $\neg p \vee q$
4. r_4 : $\neg p \vee \neg q$
5. r_5 : q
6. r_6 : $\neg q$ résolution de r_2 et r_4
7. r_7 : $\Box$ résolution de r_5 et r_6

Définition 21. Un système de réfutation est

- *cohérent* si toute formule réfutable n'est pas satisfaisable.

- *complet* si toute formule non satisfaisable est réfutable

Théorème 1. *La méthode de résolution est cohérente et complète, c'est à dire pour tout formule φ :*
φ est réfutable par résolution si et seulement si elle n'est pas satisfaisable

Preuve 7.
réfutable $\Longrightarrow$ non satisfaisable

Il existe une preuve par réfutation $r_1, ...r_m = \square$.

On suppose par l'absurde que φ est satisfaisable, alors il existe une valuation $v : var(\varphi) \longrightarrow \{0, 1\}$ satisfaisant φ

Pour tout $i \leqslant m$:

- soit $r_i \in \varphi$, dans ce cas$\overline{v}(r_i) = 1$
- soit r_i est obtenu par résolution depuis r_j et r_k aveec $j, k \leqslant i$, dans ce cas $\overline{v}(r_j) = \overline{v}(r_k) = 1$ alors $\overline{v}(r_i) = 1$, or $\overline{v}(r_m) = 0$, contradiction

✓

non satisfaisable $\Longrightarrow$ réfutable

On le montre par induction sur le nombre k de variables de φ

$\varphi = \bigwedge_{i=1}^{m} C_i$

$\boldsymbol{k = 0}$

$\varphi = \phi$ ou $\varphi = \square$ ✓

$\boldsymbol{k \geqslant 0}$

On suppose l'équivalence vraie jusqu'à un certain $k \geqslant 0$

Soit $p \in var(\varphi)$, on regroupe les clauses contenant p, celles contenant $\neg p$ et les autres.

$$S = \bigwedge_{i \,:\, (p \vee \neg p) \notin C_i} C_i$$

$$Res_p = \bigwedge_{\substack{C \vee p \in S \\ D \vee \neg p \in S}} (C \vee D)$$

$$\varphi_p = Res_p \wedge \overbrace{\bigwedge_{\substack{C \in S \\ p \notin C \\ \neg p \notin C}} C}^{\psi_p}$$

On utilisera le lemme suivant : si φ n'est pas satisfaisable alors φ_p n'est pas satisfaisable.

$|var(\varphi_p)| \leqslant |var(\varphi)| - 1 = k$ on applique l'hypothèse d'induction : φ_p est réfutable, or les clauses φ_p sont :

- soit des clauses de ψ_p
- soit des clasues obtenues par résolution à partir de clauses de φ

  Donc aucune φ_p n'est réfutable, on peut obtenir une preuve de réfutation de φ à partir de celles de φ

  autrement dit : (φ_p est réfutable $\Longrightarrow$ un sous-ensemble φ' de φ est réfutable donc φ est réfutable).

✓

✓

Montrons le lemme :

Preuve 8. On suppose φ non-satisfaisable et φ_p satisfaisable.
Soit v satisfaisant φ_p, par construction de φ_p $p \notin var(\varphi_p)$.
On peut étendre v en v_0, v_1 telles que :

- $\forall p \neq q,\ v(q) = v_0(p) = v_1(q)$
- $v_0(p) = 0$, $v_1(p) = 1$

φ n'est pas satisfaisable : $\overline{v_0}(\varphi) = 0$ et $\overline{v_1}(\varphi) = 0$
Il existe $C \in \varphi$ telle que $\overline{v_0}(C) = 0$ et $D \in \varphi$ telle que $\overline{v_1}(D) = 0$
Si $C \in \psi_p$ alors $C \in \varphi_p$ et donc $\overline{v_0}(C) = 1 = \overline{v}(C)$ donc $C \notin \psi_p$, $C \in \varphi_p$
De même $D \notin \psi_p$ $(D \in \varphi)$, donc $C, D \in \{E \mid E \in \varphi, p \in E \text{ ou } \neg p \in E\}$

- on suppose $\neg p \in C$, c'est à dire $C = C' \vee \neg p$, dans ce cas $\overline{v_0}(C) = \overline{v_0}(C')$ ou $\overline{v_0}(C) = \overline{v_0}(\neg p) = 1$, contradiction, donc $p \in C$
- De même $\neg p \in D$

Donc $C = C' \vee p$, $p, \neg p \notin C'$
$D = D' \vee \neg p$, $p, \neg p \notin D'$
Donc $C' \vee D' \in Res_p$ et $C' \vee D' \in \varphi_p$
$\overline{v}(C' \vee D') = 1$
or $\overline{v_0}(C) = 0$

- $C = C' \vee p$, $p, \neg p \notin C'$
- $\overline{v_0}(C) = \overline{v}(C') = 0$

de même $D = D' \vee \neg p$ et $p, \neg p \in D'$ donc $\overline{v_1}(D) = \overline{v}(D') = 0$, contradiction.

La preuve donne un algorithme

$S \leftarrow \varphi$
while $var(S) \neq \emptyset$ *et* $\Box \notin S$ **do**
  choisir $p \in var(S)$
  $S \leftarrow S \backslash \{C \in S \mid p \in C \text{ et } \neg p \in C\}$
  $Res \leftarrow \{C \cup P \mid C \cup \{p\} \in S \text{ et } D \cup \{\neg p\} \in S\}$
  $S \leftarrow Res \cup \{C \in S \mid p \notin C \text{ et } \neg p \in C\}$
end
if $\Box \in S$ **then**
  return 0
else
  return 1
end

Algorithm 1: DP-Résolution (Davis-Putnam, années 60)

On en déduit un tet de validité deφ en regardant si $\neg\varphi$ est non-satisfaisable.

Définition 22. La *longueur* d'une preuve par résolution est le nombre d'étapes de résolution utilisées dans la preuve

L'algorithme DP peut être très inefficace sur des formules "très simple", par exemple le principe des tiroirs.

On note PHP_n (*pigeon hole principle*) le fait qu'il n'existe pas de fonction injective de $[n+1]$ à $[n]$)

On note $\neg PHP_n = \varphi \wedge \psi$ avec $\varphi = \bigwedge_{i=1}^{n+1} p_{i,1} \vee ... \vee p_{i,n}$ et $\psi = \bigwedge_{j=1}^{n} \bigwedge_{1 \leqslant i < i' < n+1} \neg(p_{i,j} \wedge p_{i',j})$

Théorème 2. *(Haken) Il existe un entier n_0 tel que toute preuve par résolution de $\neg PHP_n$ est de longueur au moins $2^{n/32}$ avec $n \geqslant n_0$.*

9 Application à certains types de formules

Définition 23. Soit C une clause telle que $var(C) \subseteq \mathcal{P}$, C est une *clause de Horn* si C contient au plus un littéral positif.

C'est à dire qu'une clause de Horn est de la forme $\neg p_1 \vee \neg p_2 ... \neg p_k \vee q$, ou encore sous la forme $(p_1 \wedge p_2 ... \wedge p_k) \rightarrow q$

Définition 24. φ en FNC est de Horn si toute ses clauses sont de Horn

Règle de résolution unitaire :

$$\frac{\neg p_1 \vee ... \vee \neg p_k \vee q \quad q}{\neg p_1 \vee ... \vee \neg p_k}$$

$$\frac{\neg p_1 \vee ... \vee \neg p_k \vee q \quad p_i}{\neg p_1 \vee ... \vee \neg p_{i-1} \vee \neg p_{i+1} ... \vee p_k \vee q}$$

$$\frac{C \vee q \quad \neg q}{C}$$

Sur l'exemple du début on ne peut pas appliquer la règle de résolution unitaire ($p \vee q$, $\neg p \vee q$, ...)

Remarque 12. La méthode de réolution unitaire n'est pas complète pour le calcul propositionnel, c'est-à-dire qu'il existe des enoncés non-satisfaisables qui ne sont pas réfutables

Théorème 3. *La méhode de DP-résolution unitaire est cohérente et complète pour les formules de Horn*

On peut décider de la satisfaisabilité de φ en un nombre d'étapes linéaire en la taille de φ.

Notation 5. Pour tout formule φ on note

$$unit(\varphi) = \{C \in \varphi \mid |C| = 1\}$$

Remarque 13. Si φ est de Horn et $unit(\varphi) = \emptyset$ alors φ est satisfaisable.

La valuation $\forall q \in var(\varphi),\ v(q) = 0$ est telle que $\overline{v}(\varphi) = 1$

Part III

Procédure de décision pour la satisfaisabilité : DPLL

DPLL = Davis, Putnam, Logeman et Loveland, 1962

1. On choisit $p \in var(\varphi)$
2. φ est satisfaisable si et seulement si $\varphi[p/1]$ ou $\varphi[p/0]$ est satisfaisante

$S \leftarrow \varphi$
for $q \in var(\varphi)$ **do**
  $v(q) \leftarrow 0$
end
while $unit(S) \neq \emptyset$ *et* $\Box \notin S$ **do**
  choisir $\{l\} \in unit(\varphi)$
  $v(l) \leftarrow 1$
  $S \leftarrow S \backslash \{C \in S \mid l \in C \wedge \neg l \in C\}$
  $Res_p \leftarrow \{C \mid C \cup \{\neg p\} \in S\}$
  $S \leftarrow \{C \mid C \cup \{\neg p\} \in S\}$
end
if $\Box \in S$ **then**
  return 0
else
  return v
end

Algorithm 2: DP-Resolution-Horn(φ)

Propagation unitaire :

Proposition 7. *Soit $\varphi \in \mathcal{F}_\mathcal{P}$ en FNC, $C \in unit(\varphi)$, c'est-à-dire $C = l$ où $l = x_i$ ou $l = \neg x_i$. Alors φ est satisfaisable si et seulement si $\varphi[x/\top]$ est satisfaisable.*

Les clauses unitaires forcent l'assignation des variables concernées.

Littéraux purs :

Définition 25. $x \in \mathcal{P}$ est de *polarité positive* (respectivement *négative*) si $\forall C \in \varphi,\ x \in var(C) \implies x \in C$ (respectivement $x \in var(C) \implies \neg x \in C$)

En notant pour toute formule F et toute variable y : "y apparaît dans F" par $y \in F$.
On note $Pol^+(\varphi)$ l'ensemble des variables de polarité positive de φ et $Pol^-(\varphi)$ celles de polarité négative.

Proposition 8. *Si $x \in Pol^+(\varphi)$ alors φ est satisfaisable si et seulement si $\varphi[x/\top]$ est satisfaisable.*
Si $x \in Pol^-(\varphi)$ alors φ est satisfaisable si et seulement si $\varphi[x/\bot]$ est satisfaisable.

Preuve 9.
On suppose φ satisfaisable
Soit $\varphi = \bigwedge_{i \in I} C_i \wedge \bigwedge_{j \in J}$ et $x \in Pol^+(\varphi)$ où $x \in C_i,\ \forall i \in I$ et $x, \neg x \notin C_j,\ \forall j \in J$.
Il existe une valuation v satisfaisant φ, soit $v' = v_{|\mathcal{P} \backslash \{x\}}$.
$\overline{v}(\varphi) = 1$ alors $\overline{v}\left(\bigwedge_{j \in J} C_j\right) = 1$, or $x, \neg x \notin X_j,\ \forall j \in J$ donc $\overline{v'}\left(\bigwedge_{j \in J} C_j\right) = 1$
or

$$\varphi[x/\top] = \underbrace{\bigwedge_{i \in I} C_i[x/\top]}_{\top} \wedge \underbrace{\bigwedge_{j \in J} C_j[x/\top]}_{\text{satisfait par } v'}$$

✓

Si φ est satisfaisable

Soit $v \ : \ \mathcal{P}\backslash\{x\} \longrightarrow \{0,1\}$ satisfaisant $\varphi[x/\top]$, dans ce cas $v' \ : \ \mathcal{P} \longrightarrow \{0,1\}$ définie par $v'(x) = 1$ et $v'(y) = v(y), \ \forall \neq x$

φ est satisfaisable. ✓

if $\varphi = \top$ **then**
  return 1
else if $\varphi = \bot$ **then**
  return 0
else if $unit(\varphi) \neq \emptyset$ **then**
  Choisir $l \in unit(\varphi)$
  return DPLL($\varphi[l/\top]$)
else if $Pol^+(\varphi) \cup Pol^-(\varphi) \neq \emptyset$ **then**
  Choisir $x \in Pol^+(\varphi)$
  Choisir $y \in Pol^-(\varphi)$
  return DPLL($\varphi[x/\top][y/\bot]$)
else
  Choisir $x \in var(\varphi)$
  return $DPLL(\varphi[x/\top])$ ou $DPLL(\varphi[x/\bot])$
end

Algorithm 3: DPLL(φ)

Remarque 14. DPLL est très performant en pratique.
Il dépend fortement de l'heuistique du choix de x dans la dernière étape.

Heuristiques pour le choix de x

Schoning/Toran : The satisfaisability theorem
l un littéral, $k \in \mathbb{N}$ et φ une formule.

$$f_k(l) = \text{Card}\{C \ : \ C \in \varphi, |C| = k, l \in C\}$$

$$f(l) = \sum_{k \in \mathbb{N}} f_k(l)$$

$f(l)$ est le nombre d'occurrences de l dans φ.
DLIS : choisir $f(l)$ maximal et poser $l = \top$.
On choisit x tel que $f(x) + f(\neg x)$ maximal et $x = \top$ si $f(x) > f(\neg x)$, $x = \bot$ sinon.
Un SAT solver est dépendant du choix de x. Il dépend de la stratégie de backtrack.
CDCL : conflict drivers clause legacy :

- construire un graphe de conflits entre choix d'assignations de variables
- apprendre les raisons qui ont amené à une contradiction (identifier un ensemble de choix contradictoires, par exemple $x_1 = 0, y_1 = 1, z_2 = 1$)
- ajouter une clause qui empêche cette situation de se reproduire (par exemple $\neg x_1 \vee y_1 \vee \neg z_2$)

Part IV
Compléments

10 Calcul propositionnel

10.1 Théorème de lecture unique

Définition 26. Soient $w_0, w_1 = a_1...a_n \in \mathcal{M}$, on dit que w_0 est un segment initial de w_1, noté $w_0 \subseteq w_1$ si $w_0 = a_1...a_i$ avec $1 \leqslant i \leqslant n$, et w_0 est un segment propre, noté $w_0 \subsetneq w_1$ si $i < n$.

Lemme 7. *Soit $F \in \mathcal{F}$ et $G \subsetneq F$, alors $M \notin \mathcal{F}$*
Autrement dit, aucune formule n'est le préfixe d'une autre.

Proposition 9. *On note $o[F]$ le nombre de parenthèses ouvrantes d'une formule F et $f[F]$ pour ses parenthèses fermées.*

1. $\forall F \in \mathcal{F},\ o[F] = f[F]$
2. $\forall F \in \mathcal{F}, \forall M \in \Sigma^*,\ M \subsetneq F \Longrightarrow \begin{cases} o[M] > f[M],\ \textit{et donc } M \notin \mathcal{F} & (a) \\ \textbf{\textit{x-ou}}\ M = \neg ... \neg \notin \mathcal{F} & (b) \\ \textbf{\textit{x-ou}}\ M = \varepsilon \notin \mathcal{F} & (c) \end{cases}$

Preuve 10. Soient $F \in \mathcal{F}$ et $M \subsetneq F$, montrons le second point.

- Si $F = \neg G = \neg g_1...g_n$
  - cas (c) : $M = \varepsilon$
  - cas (b) : $M = \neg$
  - $M = \neg g_1...g_i \subsetneq G,\ i < n$
    alors soit $o[M] = o[g_1...g_i] > f[g_1...g_i] = f[M]$, ce qui rentre dans le cas (a)
    soit $g_1...g_i = \underbrace{\neg...\neg}_{i \text{ fois}}$, alors $M = \underbrace{\neg...\neg}_{i+1 \text{ fois}}$: on est encore dans le cas (b).
- Si $F = (G \circ H) = (g_1...g_m \circ h_1...h_n)$ et $M \subsetneq F$, soit $M = \varepsilon$ (cas (c)), soit $M \neq \varepsilon$ avec
  - $M = ($ alors $o[M] = 1 > f[M] = 0$
  - $M = (g_1...g_i,\ 1 \leqslant i \leqslant m$, donc $o[M] = o[g_1...g_i] + 1 > f[M] = f[g_1...g_i]$
  - $M = (G\circ$ donc $o[M] = 1 + o[G] > f[M] = f[G]$
  - $M = (G \circ h_1...h_i,\ 1 \leqslant i \leqslant n$, alors $o[M] = 1 + o[G] + o[h_1...h_i]$
    $o[M] = 1 + f[G] + o[h_1...h_i] > f[G] + f[h_1...h_i] = f[(G \circ h_1...h_i] = f[M]$
- Si $F \in \mathcal{P},\ M = \varepsilon$, c'est le cas (c).

$\square$

Preuve 11. Soit $F \in \mathcal{F}$

- Si $F \in \mathcal{P}$ pour tout $q \in \mathcal{P} \backslash \{F\}$, $q \neq F$.
  $\forall G \in \mathcal{F},\ F \neq \neg G$ car $|\neg G| \geqslant 2 > 1 = |F|$
  $\forall G, H \in \mathcal{F},\ \forall \star \in \{\wedge, \vee, \longrightarrow\},\ (G \star H) \neq F$ car $|F| = 1 < 5 \leqslant |(G \star H)|$
- Si $F = \neg G$ avec $G \in \mathcal{F}$, pour tout $q \in \mathcal{F}$ on a $q \neq F$.
  $\forall H \neq G$ on a $\neg H \neq F$
  $\neg G \neq (H \star K)$ pour toute formules H et G et tout opérateur $\star$.
- Si $F = (G_1 \star G_2)$, supposons $F = (H_1 \circ H_2)$ que l'on réécrit

  $$a_1 ... a_k \star b_1 ... b_l = c_1 ... c_m \circ d_1 ... d_n$$

  Montrons $G_1 = H_1$, ce qui impliquera $\star = \circ$ et $G_2 = H_2$.

  On est face à l'un des deux cas :

  $$(*)\ \underbrace{a_1 a_2 a_3 ... a_k}_{G_1 \in \mathcal{F}} \subseteq \underbrace{c_1 c_2 c_3 ... c_m}_{H_1 \in \mathcal{F}}$$

  $$(**)\ c_1 c_2 c_3 ... c_m \subseteq a_1 a_2 a_3 ... a_n$$

  les deux cas sont symétriques, on suppose $(*)$ et par l'absurde que $G_1 \neq H_1$, c'est à dire $G_1 \subsetneq H_1$, ce qui implique d'après le lemme $G_1 \notin \mathcal{F}$.

  On a également $\forall F \in \mathcal{F},\ \neg G \neq (G_1 \star G_2)$ et $\forall p \in \mathcal{P},\ p \neq (G_1 \star G_2)$.

$\square$

Exercice 3 Soit $C = \bigwedge_{i \in I} l_i$ et F telle que C soit un impliquant de F.

S'il existe j tel que C_j implique F, alors C n'est pas premier

Supposons qu'il existe j tel que C_j soit un impliquant de F.
$C = C_j \wedge l_j$ alors si une valuation v est telle que $\overline{v}(C) = 1$, c'est que $\overline{v}(C_j) = \overline{v}(l_j) = 1$
Cela signifie que C est un impliquant de C_j, C n'est donc pas un impliquant premier de F. $\checkmark$

Si pour tout j C_j n'implique pas F, alors C est premier

$\checkmark$