

# LOGI - seconde partie

Sébastien Hemon

LRDE – EPITA

8 janvier 2016

## Seconde Partie

On aborde ici des problématiques liées aux notions logiques définies.

### Problème SAT

On appelle problème SAT (pour satisfiability) la donnée d'une formule  $\varphi$  de la logique propositionnelle pour laquelle il est demandé de savoir s'il existe une valuation  $\nu$  telle que  $\|\varphi\|_\nu = \text{vrai}$ .

**Exemple 0.0.1.** Donnons  $\varphi = A \Leftrightarrow (B \vee \neg C)$ . Cette formule est bien de  $\mathcal{F}_0$ . Ce n'est ni une tautologie, ni une antilogie. On peut en donner  $\mu$  vérifiant  $\mu(A) = \text{vrai}$ ,  $\mu(B) = \text{vrai}$  et  $\mu(C) = \text{faux}$  comme solution.

Par convention, on choisira d'écrire les formules d'une instance de SAT sous une forme précise afin de faciliter les méthodes systématiques de recherche de solutions (et aussi pour mieux coller avec les opérations structurelles des processeurs).

**Definition 0.0.2 (CNF).** On appelle *Forme Normale Conjonctive (CNF)* d'une formule  $\varphi$  de  $\mathcal{F}_0$  toute formule  $\psi$  sémantiquement équivalente à  $\varphi$  et s'écrivant comme conjonction de disjonctions de littéraux ou négations de littéraux (i.e. symboles de variables propositionnelles).

Ainsi, toute CNF d'une formule s'écrit sous la forme  $\bigvee \bigwedge (\alpha) v_{ij}$  avec  $(\alpha)$  désignant rien ou une négation  $\neg$  et les  $v_{ij}$  étant des littéraux. Implicitement, le nombre d'indices est fini du fait de l'emploi des notations.

**Exemple 0.0.3.** En réécrivant l'équivalence en implications, puis les implications par material implication, on obtient  $\varphi \equiv (\neg A \vee B \vee \neg C) \wedge (A \vee \neg(B \vee \neg C))$ . On utilise une loi de De Morgan pour obtenir alors :

$$\varphi \equiv (\neg A \vee B \vee \neg C) \wedge (A \vee (\neg B \wedge C)) \equiv (\neg A \vee B \vee \neg C) \wedge (A \vee \neg B) \wedge (A \vee C)$$

La dernière formule est une CNF de  $\varphi$ .

On définit la notion de DNF par analogie, en intervertissant les rôles de conjonction et disjonction. On peut énoncer :

**Théorème 0.0.4.** Toute formule de  $\mathcal{F}_0$  est sémantiquement équivalente à une DNF et à une CNF

On remarque que ce théorème ne formule pas l'unicité car elle n'est pas vérifiée. La démonstration permettra de le remarquer.

*Démonstration :* On démontre l'existence de la DNF. Le lecteur sera invité à procéder de même pour la CNF.

On le prouve par induction sur la construction des formules de  $\mathcal{F}_0$ .

- *Atomes :* On peut écrire que  $\top \equiv (A \wedge A) \vee (\neg A \wedge \neg A)$ . On a aussi  $\perp \equiv (A \wedge \neg A) \vee (A \wedge \neg A)$ . Quel que soit le littéral  $v$  (une lettre majuscule) on a  $v \equiv (v \wedge v) \vee (v \wedge v)$ .
- *Constructeurs :* Soient  $\varphi$  et  $\psi$  deux formules de  $\mathcal{F}_0$ . On suppose qu'il est possible d'écrire une DNF pour ces deux formules, notées respectivement  $\eta$  et  $\iota$ .

1. La formule  $\varphi \wedge \psi \equiv \eta \wedge \iota$  qui est en DNF.

2. La formule  $\neg\varphi \equiv \neg\eta$  va devoir subir un traitement par loi de De Morgan. Écrivons  $\eta \equiv \bigvee_{i \leq n} \bigwedge_{k_i \in I_i} (\alpha) v_{k_i}$  où  $\bigcup I_i = I$  contient tous les indices de littéraux employés. Il vient donc :

$$\neg\eta \equiv \bigwedge_{i=1}^n \bigvee_{k_i \in I_i} \overline{(\alpha)} v_{k_i}$$

Avec  $\overline{(\alpha)}$  une négation si  $(\alpha)$  désignait rien et rien si  $(\alpha)$  désignait une négation.

On se place alors dans  $\mathbb{A}$  où l'on pourra calculer en remplaçant  $\neg, \vee, \text{wedge}$  par  $\bar{\cdot}, +, \cdot$ .

On notera  $x_*$  la classe d'équivalence de la lettre  $v_*$  dans  $\mathbb{A}$ , ou de sa négation le cas échéant.

Ainsi :

$$\begin{aligned} \text{classe de } (\neg\eta) &= \prod_{i=1}^n \sum_{k_i \in I_i} x_{k_i} \\ &= \prod_{i=1}^n (x_{I_i(1)} + \cdots + x_{I_i(p_i)}) \\ &= (x_{I_1(1)} + \cdots + x_{I_1(p_1)}) (x_{I_2(1)} + \cdots + x_{I_2(p_2)}) \cdots (x_{I_n(1)} + \cdots + x_{I_n(p_n)}) \\ &= x_{I_1(1)} x_{I_2(1)} \cdots x_{I_n(1)} + \cdots + x_{I_1(p_1)} x_{I_2(p_2)} \cdots x_{I_n(p_n)} \\ &= \sum \prod x_{I_j(k_i)} \end{aligned}$$

où  $p_i$  désigne le cardinal de  $I_i$  et où les indices finaux restent à écrire, mais permettant néanmoins de conclure sans les connaître.

Ainsi,  $\neg\eta$  admet un représentant sous une forme de somme de produits dans  $\mathbb{A}$ , c'est-à-dire une DNF dans  $\mathcal{F}_0$ .

3. La formule  $\varphi \wedge \psi \neg(\neg\varphi \vee \neg\psi)$  se ramène aux cas précédents.
4. La formule  $\varphi \Rightarrow \psi \equiv \neg\varphi \vee \psi$  se ramène aux cas précédents.
5. La formule  $\varphi \Leftrightarrow \psi \equiv (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$  se ramène aux cas précédents.

L'algèbre a permis d'établir ce théorème de logique.

On retiendra que, par convention, les instances de SAT sont toujours formulées sous forme de DNF ou de CNF. Il appartient donc à chacun d'effectuer la transformation sur une formule. Les résultats suivants sont donnés sans démonstration :

**Théorème 0.0.5 (Cook).** *Le problème SAT est un problème NP-complet*

On rappellera que NP signifie qu'une solution proposée par un oracle (processus assimilable à une machine non-déterministe) est vérifiable en temps  $\mathcal{O}(p(n))$  avec  $p$  un polynôme et  $n$  la taille de l'entrée (ici, le nombre de clauses, c'est-à-dire de conjonctions pour une CNF ou de disjonctions pour une DNF).

La locution *complet* indique que tout problème NP est polynomialement réductible à SAT. L'enjeu pour la résolution de SAT est donc, en partie, le fait de résoudre par transformation tout problème NP.

## Formes prenexes

vues en cours, non exigibles à l'examen. Il s'agit de l'analogues aux CNF et DNF avec des formules de logiques du premier ordre. On a remplacé les  $\bigwedge$  par  $\forall$  et les  $\bigvee$  par  $\exists$ .

## Graphes de relations binaires

Nous avons vu que toute relation binaire admettait un paradigme graphe. Nous allons introduire des conventions afin d'en alléger les représentations et identifier les propriétés importantes mais lourdes à visualiser.

**Definition 0.0.6.** *Pour un ensemble  $E$ , on définit  $\Delta_E$  sa diagonale comme :*

$$\Delta_E = \{(x; x) ; x \in E\}$$

La diagonale est l'ensemble associé à la relation  $=$  dans  $E$ . Pour un graphe, il s'agit de toutes les boucles d'un sommet  $x$  vers lui-même.

**Proposition 0.0.7.** *Une relation  $R$  sur  $E$  est réflexive si, et seulement si, le sous-ensemble de  $E^2$  associé contient  $\Delta_E$*

(fait évident)

Ainsi, un graphe désigne une relation réflexive dès que l'on indique qu'il contient les boucles  $\Delta_E$ , sans pour autant les représenter. Cela allège considérablement l'affichage (que l'on troque contre une simple flag ou tag). Mais il y a plus lourd.

**Definition 0.0.8.** *Soit  $R$  une relation binaire de  $E$ , un ensemble. On définit la clôture transitive de  $R$  sur  $E$  comme une relation binaire  $Q$  par induction :*

- Atomes : Si  $xRy$  alors  $xQy$ .
- constructeurs : Chaque fois que  $xQy$  et  $yQz$  alors  $xQz$ .
- arrêt : Première étape où aucune nouvelle arrête n'est construite.

Ce qui suit est une évidence :

**Proposition 0.0.9.** *La clôture transitive de toute relation binaire est une relation binaire transitive.*

En pratique, ceci permet d'éviter d'avoir à représenter toutes les arêtes d'un graphe, par exemple d'une relation d'ordre. On construit un graphe d'une certaine relation mais l'on travaille avec sa clôture transitive. Ainsi, tout chemin induit une arête, même si elle n'est pas représentée. On troque alors un nombre quadratique d'arêtes contre un simple flag ou tag.

**Proposition 0.0.10.** *Un arbre orienté est la représentation d'un ordre en présence des tag clôture transitive et  $\Delta_E$ .*

La réciproque est fausse cependant. Voir les treillis.

**Theorem 0.0.11.** *Un graphe représente une relation d'équivalence si, et seulement si, chacune de ses composantes connexes est complète.*

Démonstration : Un tel graphe induit une partition de ses sommets.

On obtient alors, avec les conventions :

**Proposition 0.0.12.** *En présence des tags  $\Delta_E$  et clôture transitive, tout graphe non orienté est celui d'une relation d'équivalence.*

Il conviendra donc de bien comprendre que ces tags permettent de limiter la classe d'étude des graphes à une situation donnée. Il est donc également très important de signaler la présence ou non de ces tags.