Transformation de Tseintin

Julie Parreaux

2018-2019

Référence du développement : Aucune référence connue

Leçons où on présente le développement : 916 (logique propositionnelle).

1 Introduction

Les problèmes SAT et CNF-SAT sont des problèmes NP-complet. La mise sous forme normale d'une formule conjonctive (CNF) est donc une action non coûteuse (polynomiale via la réduction entre ces deux problèmes) et donne accès à des méthodes de résolution pour SAT. En effet, les SAT-solver qui tentent de résoudre le problème SAT avec une complexité temporelle en temps polynomiale (dans le cas moyen car dans le pire cas on sera en exponentielle) utilisent des algorithmes (comme DPLL) qui prennent en entrée des formules sous forme CNF voir même sous forme 3CNF. Avoir un algorithme efficace pour effectuer cette transformation est donc essentiel en pratique.

Il existe un algorithme qui effectue une transformation linéaire pour mettre une formule sous forme normale CNF. La transformation de Tseintin que nous étudions ici met sous forme normale une formule de taille linéaire avec la contrainte de l'équisatisfiabilité (et non l'équivalence). Mais comme la principale application est la résolution du problème SAT cela ne pose pas de problème.

Remarques sur le développement

1. Description de la transformation de Tseintin.
2. Preuve de la linéarité de la transformation.
3. Preuve de l'équisatisfiabilité de la transformation.

2 Mise sous forme normale conjonctive via Tseintin

Théorème. *Pour toute formule φ, il existe une formule sous forme CNF $T(\varphi)$ de taille linéaire en la taille de la formule φ telle que φ et $T(\varphi)$ sont équisatisfiables (**Attention** : pas d'équivalence entre les formules).*

Remarque : On fait même mieux car on est capable de construire une valuation pour φ.

Démonstration. Soit φ une formule.

— On note $|\varphi|$ la taille de la formule φ correspondant aux nombres de connecteurs logique de φ.

— On note $SF(\varphi)$ l'ensemble des sous-formule de φ (φ comprise) et $V(\varphi)$ l'ensemble des variables propositionnelles (proposition atomique) de φ.

Pour tout sous-formule $\psi \in SF(\varphi)$, on définie une nouvelle variable propositionnelle a_ψ (à qui on donne le sens a_ψ est vrai, lorsqu'on définit une valuation pour la formule contenant cette nouvelle variable).

— Comme $\{\neg, \vee\}$ est un système de connecteur complet, on peut supposer que φ ne contient que ces connecteurs. Si ce n'est pas le cas, on utilise les règles de transformation (ce sont des règles de réécriture) $a \wedge b \equiv \neg(\neg a \vee \neg b)$, $a \Rightarrow b \equiv \neg a \vee b$ et $a \Leftrightarrow b \equiv \neg(\neg(\neg a \vee b) \vee \neg(\neg b \vee a))$ pour remplacer les autres connecteurs. La formule ainsi obtenue est linéaire en la taille de φ la formule de départ (par exemple, on ajoute trois au nombre de $\vee$).

— On pose maintenant, $T(\varphi) = a_\varphi \wedge \bigwedge_{\psi \in SF(\varphi)\setminus V} \underbrace{t(\psi)}_{\text{ne contient que des } \neg \text{ et des } \vee}$ où

$$t(\neg\psi) = a_{\neg\psi} \Leftrightarrow \neg a_\psi = (a_\psi \vee a_{\neg\psi}) \wedge (\neg a_{\neg\psi} \vee \neg a_\psi)$$
$$t(\psi_1 \vee \psi_2) = a_{\psi_1 \vee \psi_2} \Leftrightarrow a_{\psi_1} \vee a_{\psi_2} = (a_{\psi_1 \vee \psi_2} \vee a_{\neg\psi_1}) \wedge (a_{\psi_1 \vee \psi_2} \vee \neg a_{\psi_2}) \wedge (\neg a_{\psi_1 \vee \psi_2} \vee \neg a_{\psi_1} \vee \neg a_{\psi_1})$$

On remarque que $|t(\neg\psi)| = 5$ et $|t(\psi_1 \vee \psi_2)| = 10$ (elles sont donc bornées).

Lemme. *$|T(\varphi)|$ est linéaire en $|\varphi|$.*

Démonstration. Soit φ une formule ne contenant que des connecteurs de $\{\vee, \neg\}$.

Montrons que $\#SP(\varphi) \leq 2|\varphi| + 1$ Montrons cette propriété par induction sur φ construit avec le système de connecteurs complet $\{\vee, \neg\}$. On pose $\mathcal{H}_\varphi$ l'hypothèse d'induction définie par $\mathcal{H}_\varphi = "\#SP(\varphi) \leq 2|\varphi| + 1"$.

Cas de base : $\varphi \in V$ Comme $\varphi \in V$, le nombre de connecteurs logique de φ est 0. Donc $|\varphi| = 0$. De plus, $\#SF(\varphi) = \#\{a_\varphi\} = 1$. On en déduit $\mathcal{H}_\varphi$.

Cas inductif : $\varphi = \neg\varphi_1$ Supposons que $\mathcal{H}_{\varphi_1}$ est vraie et montrons $\mathcal{H}_\varphi$. On a

$$\begin{array}{rcll} \#SF(\varphi) & = & 1 + \#SF(\varphi_1) & (SF(\varphi) = \varphi \cup SF(\varphi_1)) \\ & \leq & 1 + 2|\varphi_1| + 1 & \text{(hypothèse d'induction)} \\ & = & 2(|\varphi_1| + 1) = 2|\varphi| & (|\varphi| = |\varphi_1| + 1) \\ & \leq & 2|\varphi| + 1 & \end{array}$$

Cas inductif : $\varphi = \varphi_1 \vee \varphi_2$ Supposons que $\mathcal{H}_{\varphi_1}$ et $\mathcal{H}_{\varphi_2}$ sont vraies et montrons $\mathcal{H}_\varphi$. On a

$$\begin{array}{rcll} \#SF(\varphi) & = & 1 + \#SF(\varphi_1) + \#SF(\varphi_2) & (SF(\varphi) = \varphi \cup SF(\varphi_1) \cup SF(\varphi_2)) \\ & \leq & 1 + 2|\varphi_1| + 1 + 2|\varphi_2| + 1 & \text{(hypothèses d'inductions)} \\ & = & 2(|\varphi_1| + |\varphi_2| + 1) + 1 = 2|\varphi| + 1 & (|\varphi| = |\varphi_1| + |\varphi_2| + 1) \end{array}$$

D'ou $\mathcal{H}_\varphi$ pour toute formule φ du calcul propositionnel.

L'idée est que pour tout connecteur, on sépare le sous-formule en au plus deux sous-formules (on peut les voir comme des arbres).

Montrons que $|T(\varphi)|$ est linéaire en $|\varphi|$ On a

$$\begin{array}{rcll} |T(\varphi)| & = & 1 + \left|\bigwedge_{\psi \in SF(\varphi)\setminus V} t(\psi)\right| & \text{(définition de } T(\varphi)) \\ & \leq & 1 + k\#SF(\varphi) & \text{(majoration de la conjonction)} \\ & \leq & 1 + k(2|\varphi| + 1) & \text{(propriété précédente)} \\ & = & 1 + 2k|\varphi| + 1 + k & \end{array}$$

Comme k est bornée ($k \leq 10$), on est bien linéaire en la taille de φ. □

Lemme. *$T(\varphi)$ et φ sont équisatisfiable.*

Démonstration. $\Rightarrow$ Supposons que φ est satisfiable, alors il existe une valuation ν telle que $\nu(\varphi) = \text{vrai}$. On pose ν' telle que pour tout $\psi \in SF(\varphi)$, $\nu'(a_\varphi)\nu(\varphi)$. Montrons par induction sur φ que $\nu'(T(\varphi)) = \text{vrai}$. On pose l'hypothèse d'induction $\mathcal{H}_\varphi$: $"\nu(\varphi) = \text{vrai} \Rightarrow \nu'(\varphi) = \text{vrai}"$.

Cas de base : $\varphi \in V$ Supposons que $\nu(\varphi) = \text{vrai}$ et montrons que $\nu'(\varphi) = \text{vrai}$. Par définition de $T(\varphi)$, $T(\varphi) = a_\varphi \wedge \bigwedge_{\psi \in SF(\varphi)\setminus V} t(\psi) = a_\varphi \wedge \bigwedge_{\emptyset} t(\psi) = a_\varphi$ (car $SF(\varphi) = a_\varphi$ et $SF(\varphi) \setminus V = \emptyset$). On en déduit que $\nu'(T(\varphi)) = \nu'(a_\varphi) = \text{vrai}$

Cas inductif : $\varphi = \neg\varphi_1$ Supposons que $\mathcal{H}_{\varphi_1}$ est vraie et montrons $\mathcal{H}_\varphi$. Par définition de $T(\varphi)$, $T(\varphi) = a_\varphi \wedge \bigwedge_{\psi \in SF(\varphi_1)\setminus V} t(\psi) = a_\varphi \wedge T(\varphi_1)$ (car $\bigwedge_{\psi \in SF(\varphi_1)\setminus V} t(\psi) = T(\varphi_1)$). En l'évaluant avec ν', on en déduit que $\nu'(T(\varphi)) = \nu'(a_\varphi) \wedge \nu'(T(\varphi_1))$. Par hypothèse sur a_φ et en appliquant les hypothèses d'induction, $\nu'(a_\varphi) = \nu'(T(\varphi_1)) = \text{vrai}$. Donc $\nu'(T(\varphi)) = \text{vrai}$.

Cas inductif : $\varphi = \varphi_1 \vee \varphi_2$ Supposons que $\mathcal{H}_{\varphi_1}$ et $\mathcal{H}_{\varphi_2}$ sont vraies et montrons $\mathcal{H}_\varphi$. Par définition de $T(\varphi)$, $T(\varphi) = a_\varphi \wedge \bigwedge_{\psi \in SF(\varphi_1)\setminus V} t(\psi) \wedge \bigwedge_{\psi \in SF(\varphi_2)\setminus V} t(\psi) = a_\varphi \wedge T(\varphi_1) \wedge T(\varphi_2)$ (car $\bigwedge_{\psi \in SF(\varphi_i)\setminus V} t(\psi) = T(\varphi_i)$). En l'évaluant avec ν', on en déduit que $\nu'(T(\varphi)) = \nu'(a_\varphi) \wedge \nu'(T(\varphi_1)) \wedge \nu'(T(\varphi_2))$. Par hypothèse sur a_φ et en appliquant les hypothèses d'induction, $\nu'(a_\varphi) = \nu'(T(\varphi_1)) = \nu'(T(\varphi_2)) =$ vrai. Donc $\nu'(T(\varphi)) =$ vrai.

D'ou $\mathcal{H}_\varphi$ pour toute formule φ du calcul propositionnel.

$\Leftarrow$ Supposons que $T(\varphi)$ soit satisfiable : il existe une valuation ν tel que $\nu(T(\varphi)) =$ vrai. On pose ν' telle que pour tout $p \in V$, $\nu'(p) = \nu(a_\varphi)$. Alors, $\nu'(\varphi) =$ vrai car la variable associée à une sous-formule est équivalente à cette sous formule.

□

□

Exemple. On applique cette transformation à la formule $\varphi = (p \vee \neg q) \vee (\neg p)$. On obtient $T(\varphi) = a_\varphi \wedge t(\neg q) \wedge t(p \vee \neg q) \wedge t(\neg p) \wedge t((p \vee \neg q) \vee (\neg p))$. On conclut avec

$$\begin{aligned}\varphi = \;& a_\varphi \\ & \wedge(a_p \vee a_{\neg q}) \wedge (\neg a_p \vee \neg a_{\neg q}) \\ & \wedge(a_q \vee a_{\neg p}) \wedge (\neg a_q \vee \neg a_{\neg p}) \\ & \wedge(a_{p\vee\neg q} \vee \neg a_p) \wedge (a_{p\vee\neg q} \vee \neg a_{\neg q}) \wedge (\neg a_{p\vee\neg q} \vee a_p \vee a_{\neg p}) \\ & \wedge(a_{((p\vee\neg q)\vee(\neg p))} \vee \neg a_{(p\vee\neg q)}) \wedge (a_{((p\vee\neg q)\vee(\neg p))} \vee \neg a_{\neg q}) \wedge (\neg a_{((p\vee\neg q)\vee(\neg p))} \vee a_{(p\vee\neg q)} \vee a_{\neg p})\end{aligned}$$

3 Équivalence sémantique et équisatisfiabilité

Comparer deux formules c'est connaître les modèles qui les discrimine (satisfait une des formule mais pas l'autre). Lorsque nous ne pouvons pas les différentier, nous parlons d'équivalence sémantique (c'est l'équivalence que l'on souhaite). Cependant être sémantiquement équivalent est une contrainte forte pour notre but de résoudre la satisfiabilité. On va alors définir une deuxième notion ; l'équisatisfiablité.

Formules sémantiquement équivalentes [1, p.106] On rappelle qu'une formule est satisfiable s'il existe un modèle (une valuation dans notre cas) qui rend la formule vrai. On dit que la formule est satisfaite dans ce modèle. Une formule est dite valide si pour tout modèle la formule est rendue vrai.

Définition. Deux formules sont équivalentes sémantiquement si et seulement si elles sont satisfaites dans les même modèles, on note $\varphi \equiv \psi$.

Proposition. *$\varphi \equiv \psi$ si et seulement si $\varphi \Leftrightarrow \psi$ est valide.*

Idée de la démonstration. $\Rightarrow$ Soit un modèle, par leur équivalence, elles prennent les mêmes valeurs sur ce modèle d'où la validité.
$\Leftarrow$ Soit un modèle, comme $\varphi \Leftrightarrow \psi$ est valide, φ et ψ prennent les mêmes valeurs sur ce modèle.

□

Exemple. Les lois de De Morgan.

Remarque. La relation $\equiv$ est bien une relation d'équivalence (elle est réflexive, symétrique et transitive).

Formules équisatisfiables Souvent, on cherche une formule sémantiquement équivalente pour répondre au problème de la satisfiabilité. L'équivalence pour tous les modèles n'est donc pas nécessaire. Dans ce contexte, deux formules telles que si l'une est satisfiable alors l'autre l'est également suffit.

Définition. Deux formules φ et ψ sont dites équisatisfiable si et seulement si (φ est satisfiable si et seulement si ψ est satisfiable).

Autrement dit, deux formules sont dites équisatisfiable si l'une des deux admet un modèle qui la satisfait l'autre aussi. Cependant ce modèle peut être distinct (on ne demande pas que se soit le même modèle qui satisfont les deux formules).

Proposition. *Deux formules équivalentes sont équisatisfiables.*

Démonstration. Soit φ et ψ deux formules telles que $\varphi \equiv \psi$. Comme $\varphi \Leftrightarrow \psi$ est valide, φ est satisfiable si et seulement si ψ l'est aussi (elles le sont pour le même modèle). D'où l'équisatisfiabilité. □

4 Formes normales

Les formes normales sont une manière de représenté une formule par une autre formule dont les connecteurs logiques sont restreints et ordonnés selon un certain ordre [1, p.166]. Ces deux formules peuvent être équivalentes mais sont généralement équisatisfiable. Cette condition suffit puisque la mise sous forme normale est très souvent effectuée pour répondre au problème de la satisfiabilité d'une formule.

Forme normale négative La forme normale négative est à la base des deux formes suivante puisque les formes normales conjonctives et disjonctives sont négatives. Les négations ne peuvent porter que sur des variables propositionnelles et non sur des sous-formules.

Définition. Une forme normale négative est une forme normale engendrée par le grammaire

$$\varphi ::= \bot \mid \top \mid p \mid \neg p \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi)$$

où p est une variable propositionnelle.

Proposition. *Toute formule admet une forme normale équivalente.*

Idée de la démonstration. On effectue une traduction syntaxique inductive (sur la formule à traduire) où seul le cas de la négation est réécrit.

- $tr(\bot) = \bot$
- $tr(\top) = \top$
- $tr(p) = p$
- $tr(\neg p) = \neg p$
- $tr(\varphi \vee \psi) = tr(\varphi) \vee tr(\phi)$
- $tr(\varphi \wedge \psi) = tr(\varphi) \wedge tr(\phi)$
- $tr(\neg(\varphi \vee \psi)) = tr(\neg\varphi) \wedge tr(\neg\phi)$
- $tr(\neg(\varphi \wedge \psi)) = tr(\neg\varphi) \vee tr(\neg\phi)$

□

Forme normale conjonctive Les formes normales conjonctives sont les formes normales les plus utilisée : leur satisfiabilité restent NP-complet mais la traduction peut se faire en temps linéaire. Nous avons donc une représentation plus simple pour les SAT-solveur qui n'explose pas à la transformation. C'est pour cette raison que nous l'utilisons souvent.

Définition. Un littéral est une variable propositionnelle ou sa négation.

Définition. Une forme normale conjonctive est une formule de la forme $\bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m} l_{i,j} \right)$ où les $l_{i,j}$ sont des littéraux.

Remarque. Une forme normale conjonctive est une forme normale négative (la réciproque est fausse).

Définition. Une disjonction de littéraux est appelée une clause. Cela explique le fait que nous parlons de forme clausale : on a une autre représentation.

Proposition. *Pour toute formule ψ, il existe une formule φ sous forme normale conjonctive telle que ψ et φ soient équivalentes.*

Idées de la démonstration. **Preuve existentielle**
— ψ n'admet pas de modèle où elle est fausse (c'est une tautologie) : elle est équivalente à $\top$.
— ψ admet un modèle où elle est fausse : on prend alors la négation de la formule que l'on transforme en DNF (c'est possible car sur ces modèles la négation est vraie et ce sont les seuls modèles). On remet la négation à la formule obtenue.

Preuve constructive On met la formule sous forme normale négative puis application de ces règles de distributivité.

□

Remarque. Cette transformation peut exploser.

Théorème. *Le problème de validité* (CNF-SAT) *est dans NP.*

Idée de la démonstration. On fait une réduction à SAT à l'aide de la transformation de Tseintin qui est linéaire. □

Théorème. *Le problème de validité* (CNF-SAT) *est dans P.*

Idée de la démonstration. Une forme normale conjonctive est valide si et seulement si pour toute clause, il existe une variable telle que cette variable et sa négation soient dans la clause. Cette propriété implique un algorithme linéaire testant la validité.

$\Rightarrow$ Raisonnons par l'absurde et supposons qu'il existe une clause ne contenant pas la négation d'aucun de ces littéraux. Alors le modèle qui rend faux chacun de ces littéraux ne satisfait pas la formule (la clause n'est pas satisfaite car elle ne contient aucune négation des littéraux que nous avons mis à faux). Contradiction avec la validité de la formule

$\Leftarrow$ Soit une telle formule et soit un modèle pour φ. Comme dans chacune des clauses il existe une variable de cette clause telle que sa négation apparaissent alors la clause est vraie (car la négation ou la variable est vraie).

□

Forme normale disjonctive Les formules conjonctives (et uniquement conjonctives) permettent de représenter un modèle. En effet, elle est vrai si et seulement si les valeurs de vérités du modèles sont transcrite dans les littéraux de la formule. Un forme normale disjonctive est donc satisfiable si et seulement si une de ces clauses décrit un modèle.

Définition. Une forme normale disjonctive est une formule de la forme $\bigvee_{i=1}^{n}\left(\bigwedge_{j=1}^{m} l_{i,j}\right)$ où les $l_{i,j}$ sont des littéraux.

Remarque. Une forme normale disjonctive est une forme normale négative (la réciproque est fausse).

Proposition. *Pour toute formule ψ, il existe une formule φ sous forme normale disjonctive telle que ψ et φ soient équivalentes.*

Idées de la démonstration. **Preuve existentielle**

— ψ n'admet pas de modèle : elle est équivalente à n'importe quelle contradiction : $x \wedge \neg x$

— ψ admet un modèle : alors pour chacun de ces modèles on construit la formule conjonctive qui décrit sa valuation et on regroupe dans une disjonction la totalité de ces modèles.

Preuve constructive On procède de la même manière que pour les formes normales conjonctives en inversant les règles de distributivité (mise sous forme normale négative puis application de ces règles de distributivité). On obtient les même effet que pour la forme normale conjonctive : on a une explosion de la taille de la formule.

□

Théorème. *Le problème de satsifiabilité* (DNF-SAT) *est dans P.*

Idée de la preuve. Pour chacune des clause (tant qu'on n'a pas réussi à en rendre une vraie) : réinitialiser la valuation et affecter la valeur vrai à chacun des littéraux. □

Conséquence : Par le théorème de Cook et si $P \neq NP$, on sait qu'il n'existe pas de transformation polynomiale d'une formule en une formule sous forme normale disjonctive équisatisfiable (ou équivalente).

Contre-exemple. Soit $(\varphi_n)_{n\in\mathbb{N}}$ une famille de formule du calcul propositionnel telle que sa mise sous forme disjonctive est exponentielle (même pour l'équisatisfiabilité). On pose, pour tout $n \in \mathbb{N}$, $\varphi_n = (a_1 \vee b_1) \wedge \cdots \wedge (a_n \vee b_n)$ qui a $2n$ variables et $2n-1$ connecteurs. La forme normale disjonctive de φ_n s'écrit $\bigvee_{x_i\in\{a_i,b_i\}}(x_1 \wedge \cdots \wedge x_n)$ avec $2n$ disjonctions.

Théorème. *Le problème de validité* (DNF-SAT) *est NP-complet.*

Remarque. La dualité entre les eux formes normales pour les problèmes de validité et de satisfiabilité provient de la dualité entre ces deux problèmes et du fait que la négation d'une forme normale conjonctive donne une forme normale disjonctive (sous forme normale négative) et réciproquement.

5 Systèmes de connecteurs

L'ensemble des connecteurs logiques, appelé système de connecteurs, que nous choisissons pour écrire nos formules peuvent exprimer plus ou moins de choses [1, p.175]. Lorsqu'ils permettent d'exprimer toute la logique propositionnelle, on parle de système de connecteurs complets. La recherche de tels systèmes nous permet de nous assurer de la cohérence de la définition de notre logique (on exprime tous ce qu'on veut exprimer). On peut également chercher à exprimer la logique avec le moins de connecteurs possibles : on cherche donc des systèmes de connecteurs complets et minimaux.

Cette notion de système de connecteurs complets et minimaux existe dans toutes les logiques (modale, premier ordre, ...) mais ne sont pas présenté sous cette forme.

Définition. Un système de connecteur est dit complet si toute formule est équivalente à une formule qui ne contient que ces connecteurs là. Si, de plus, tous sous-ensembles de connecteurs issus de ce système ne forme un système de connecteur complet, on parlet de système complet minimal.

Proposition. *1. $\{\neg, \vee\}$ est un système complet minimal.*

2. $\{\neg, \wedge\}$ est un système complet minimal.

3. $\{\neg, \wedge, \vee\}$ est un système complet mais pas minimal.

4. $\{\mathsf{nand}\}$ est un système complet minimal où $P \ \mathsf{nand}\ Q \equiv \neg(P \wedge Q)$.

Idée de la démonstration. 1. Le système $\{\neg, \vee\}$ est un système complet par les règles de réécriture $a \wedge b \equiv \neg(\neg a \vee \neg b)$, $a \Rightarrow b \equiv \neg a \vee b$ et $a \Leftrightarrow b \equiv \neg(\neg(\neg a \vee b) \vee \neg(\neg b \vee a))$ remplaçant les autres connecteurs. Montrons qu'il est minimal.

$\emptyset$ Dans ce cas, on n'a que les variables propositionnelles. Alors, on ne peut trouver une formule qui ne soit qu'une variable équivalente à $p \vee q$.

$\neg$ On se ramène au cas précédent en remarquant que toutes les formules sont de cette forme : $r = \underbrace{\neg \dots \neg}_{n \text{ fois}} x$. Donc r est une variable positive si n est paire et sa négation sinon. On ne peut trouver une formule r équivalente à $p \vee q$.

$\vee$ On veut une infinité de formule non équivalente deux à deux. On commence par chercher une formule équivalente à $\neg P$ ce qui est impossible (raisonnement par l'absurde).

2. On ne fait que $\{\neg, \vee\}$, $\{\neg, \vee\}$ est analogue ($\vee$ est le dual de $\wedge$).

3. Le système $\{\neg, \vee, \wedge\}$ est un système complet par les lois de Morgan et non minimal car $\{\neg, \vee\} \subset \{\neg, \vee, \wedge\}$ est minimal.

4. Le système $\{\mathsf{nand}\}$ est minimal (pour la même raison que le cas vide précédent). Montrons qu'il est complet. Pour cela, on va montrer que toute formule exprimer dans le système $\{\neg, \vee\}$ est équivalente à une de notre système. On raisonne donc par induction sur la hauteur de notre formule.

$h = 0$ Dans ce cas $\varphi = p$ avec $p \in VAR$ et $\varphi \equiv (p\mathsf{nand})\mathsf{nand}(p\mathsf{nand})$.

$h > 0$ On distingue deux cas :

$\varphi = \neg\psi$ L'hypothèse d'induction nous donne ψ' et on pose $\varphi \equiv (\psi'\mathsf{nand}\psi')$.

$\varphi = \psi_1 \vee \psi_2$ L'hypothèse d'induction nous donne ψ_1' et ψ_2' et on pose $\varphi \equiv (\psi_1'\mathsf{nand}\psi_2')\ \mathsf{nand}\ (\psi_1'\mathsf{nand}\psi_2')$.

□

Remarque. Le système de connecteur que nous avons défini suffit pour décrire la logique propositionnelle car il est complet.

Références

[1] J. Duparc. *La logique pas à pas*. Presse polytechnicienne et université romandes, 2015.