Logique propositionnelle

P. Skler

Mai 2024

Table des matières

I Eléments de base du calcul propositionnel

D'un point de vue formel, une logique est définie par une syntaxe, c'est à dire la donnée d'un ensemble de symboles et de règles. Il s'agit donc d'un langage (dont les mots sont appelés les formules logiques), à qui on associe une sémantique permettant d'attribuer une valeur (le vrai ou le faux) aux symboles et aux formules. On distingue deux types de logique : la logique des propositions, qui définit les lois formelles du raisonnement, et la logique des prédicats, qui formalise le langage des mathématiques en s'autorisant l'usage de quantificateurs (en général $\exists$ et $\forall$). Par exemple, « $(a$ ou $b) \Rightarrow b$ »est une formule de la logique propositionnelle, alors que « $\exists x, (x$ ou $b) \Rightarrow b$ »relève de la logique des prédicats.
La sémantique permet d'attribuer aux variables libres une valeur (vrai ou faux) et d'en déduire, à l'aide de règles de calcul, la valeur d'une formule. Par exemple, pour $a = vrai$ et $b = faux$ la formule « $(a$ ou $b) \Rightarrow b$ »est fausse. En revanche, pour $b = faux$ la formule « $\exists x, (x$ ou $b) \Rightarrow b$ »est vraie.
Dans la suite de ce cours, nous nous intéresserons exclusivement à la logique propositionnelle.

1) Définitions

Définition I - 1 : Proposition
En mathématique une proposition est une phrase non ambiguë à laquelle on peut attribuer une valeur de vérité : vrai ou faux. Cette valeur peut dépendre de paramètres contenus dans la proposition.

Définition I - 2 : Variable propositionnelle
On appelle variable propositionnelle une variable pouvant prendre deux valeurs : « vraie », « faux ».

Remarque : on note souvent les valeurs **V** et **F**, mais on rencontre aussi la notation $\top$ et $\bot$.

Définition I - 3 : Connecteurs logiques
les connecteurs logiques sont des constructeurs qui vont permettre de construire des formules logiques. On utilisera les suivants :
- la négation $\neg$, c'est un connecteur d'arité 1
- la conjonction $\wedge$, c'est un connecteur d'arité 2
- la disjonction $\vee$, c'est un connecteur d'arité 2
- l'implication $\rightarrow$, c'est un connecteur d'arité 2
- l'équivalence $\leftrightarrow$, c'est un connecteur d'arité 2

Définition I - 4 : Formule propositionnelle
On considère $\mathcal{P}$ un ensemble (non vide et au plus dénombrable) de variables propositionnelles, et les constantes **V** et **F**.
Une formule propositionnelle est alors
- **V** ou **F**
- p, avec $p \in \mathcal{P}$
- $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ ou $(A \leftrightarrow B)$, où A et B sont des formules propositionnelles.

Autre formulation : définition inductive de l'ensemble des formules propositionnelles.

Définition I - 5 : Formule propositionnelle
On considère $\mathcal{P}$ un ensemble (non vide et au plus dénombrable) de variables propositionnelles, et les constantes **V** et **F**.
L'ensemble des formules propositionnelles $\mathcal{F}$ sur $\mathcal{P}$ est alors défini par :
- **V**$\in \mathcal{F}$, **F**$\in \mathcal{F}$
- si $p \in \mathcal{P}$ alors $p \in \mathcal{F}$
- si $(A, B) \in \mathcal{F}^2$ alors $(\neg A) \in \mathcal{F}$, $(A \wedge B) \in \mathcal{F}$, $(A \vee B) \in \mathcal{F}$, $(A \rightarrow B) \in \mathcal{F}$ ou $(A \leftrightarrow B) \in \mathcal{F}$

Remarque : Il existe d'autres connecteurs logiques qui permettent de construire encore des formules proposi-

tionnelles.

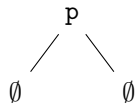
Propriété I - 1 : Unicité de l'écriture
Dans l'écriture syntaxique des formules : $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \to B)$ ou $(A \leftrightarrow B)$, les formules A et B sont uniques.

Remarque : Ecriture infixée totalement paranthésée :
Les parenthèses sont des éléments de la syntaxe des formules logiques, elles sont nécessaires pour qu'il n'y ait pas d'ambiguïté.
On décide souvent de ne pas écrire les parenthèses les plus externes.
On convient d'un ordre de priorité pour alléger l'écriture : la négation est prioritaire sur les autres connecteurs. Par exemple on écrira $\neg x \wedge y$ au lieu de $((\neg x) \wedge y)$.

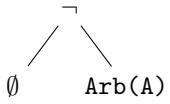
2) Représentation par un arbre

Toute formule logique de l'ensemble défini ci-dessus peut être représentée par un arbre binaire étiqueté unique, défini récursivement de la façon suivante :

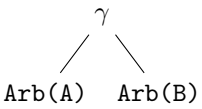
- une variable propositionnelle p est représentée par l'arbre dont l'étiquette contient p et dont les deux sous-arbres sont vides



- $(\neg A)$ est représentée par l'arbre dont l'étiquette contient $\neg$, dont le sous-arbre droit est l'arbre représentant la formule A et dont le sous-arbre gauche est vide



- si γ est l'un des symboles $\vee, \wedge, \to, \leftrightarrow$, alors $(A\ \gamma\ B)$ est représentée par l'arbre dont l'étiquette contient γ, dont le sous-arbre gauche (*resp.* droit) est l'arbre représentant la formule A (*resp.* B).

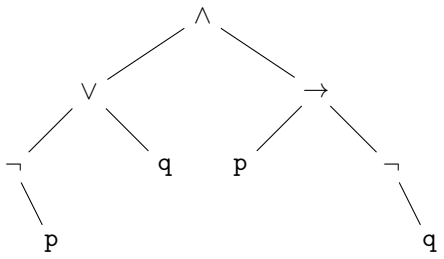


Inversement, à partir de tout arbre vérifiant les conditions suivantes :

- les feuilles (nœuds dont les deux sous-arbres sont vides) sont étiquetées par une variable propositionnelle ;
- les nœuds dont le sous-arbre gauche est vide mais pas le droit sont étiquetés par $\neg$;
- les nœuds dont les deux sous-arbres sont non vides sont étiquetés par l'un des symboles $\vee, \wedge, \to, \leftrightarrow$

Il est facile de reconstruire la formule logique correspondante (grâce à un parcours infixé de l'arbre avec ajout des parenthèses).

Exemple - 1 :
La formule $(\neg p \vee q) \wedge (p \to \neg q)$ se représente alors par :



Définition I - 6 : Taille et hauteur d'une formule

Soit $\mathcal{F}$ l'ensemble des formules propositionnelles sur $\mathcal{P}$.
On définit la taille d'une formule par :
- $\texttt{taille}(\mathbf{F})=\texttt{taille}(\mathbf{V})= 1$
- si $p \in \mathcal{P}$ alors $\texttt{taille}(p)= 1$
- pour $(A, B) \in \mathcal{F}$,
  - $\star$ $\texttt{taille}(\neg A)= 1+\texttt{taille}(A)$
  - $\star$ $\texttt{taille}(A \vee B)= 1+\texttt{taille}(A)+\texttt{taille}(B)$
  - $\star$ $\texttt{taille}(A \wedge B)= 1+\texttt{taille}(A)+\texttt{taille}(B)$
  - $\star$ $\texttt{taille}(A \to B)= 1+\texttt{taille}(A)+\texttt{taille}(B)$
  - $\star$ $\texttt{taille}(A \leftrightarrow B)= 1+\texttt{taille}(A)+\texttt{taille}(B)$

De même on définit la hauteur d'une formule par :
- $\texttt{hauteur}(\mathbf{F})=\texttt{hauteur}(\mathbf{V})= 0$
- si $p \in \mathcal{P}$ alors $\texttt{hauteur}(p)= 0$
- pour $(A, B) \in \mathcal{F}$,
  - $\star$ $\texttt{hauteur}(\neg A)= 1+\texttt{hauteur}(A)$
  - $\star$ $\texttt{hauteur}(A \vee B)= 1 + \max(\texttt{hauteur}(A),\texttt{hauteur}(B))$
  - $\star$ $\texttt{hauteur}(A \wedge B)= 1 + \max(\texttt{hauteur}(A),\texttt{hauteur}(B))$
  - $\star$ $\texttt{hauteur}(A \to B)= 1 + \max(\texttt{hauteur}(A),\texttt{hauteur}(B))$
  - $\star$ $\texttt{hauteur}(A \leftrightarrow B)= 1 + \max(\texttt{hauteur}(A),\texttt{hauteur}(B))$

Remarque : On remarque que la taille d'une formule correspond au nombre de noeuds de l'arbre associé, et la hauteur de la formule est la hauteur de ce même arbre.

Définition I - 7 : Sous formules

Soit A une formule propositionnelle. On définit l'ensemble des sous-formules de A par :
- si $A \in \mathcal{P}$, alors $sf(A) = \{A\}$
- si $A = \neg B$, alors $sf(A) = \{A\} \cup sf(B)$
- si $A = B\gamma C$, avec $\gamma \in \{\vee, \wedge, \to, \leftrightarrow\}$, alors $sf(A) = \{A\} \cup sf(B) \cup sf(C)$

Remarque : On retrouve donc l'ensemble des formules correspondant aux sous-arbres, au sens large, de l'arbre associé à la formule de départ.

II Sémantique

1) Evaluation des formules logiques

Définition II - 1 : Valuation

Soit $\mathcal{P}$ un ensemble de variables propositionnelles. On appelle **valuation** de $\mathcal{P}$ toute application de $\mathcal{P}$ dans $\{V, F\}$.

Remarque : En pratique on utilise souvent des valuations à valeurs dans $\{0, 1\}$, où 0 est pour « Faux »et 1 pour « Vrai ».

On parle aussi de distribution de vérité ou de contexte ou encore d'interprétation.

Si $\mathcal{P}$ est de cardinal n alors il existe 2^n distributions de vérités.

a) Interprétation des connecteurs

Soit σ une valuation sur $\mathcal{P}$, on définit alors la valeur de vérité v_σ des connecteurs de la façon suivante, pour $(p, q) \in \mathcal{P}^2$, :
- $v_\sigma(p) = \sigma(p)$
- $v_\sigma(\neg p) = 1$ si et seulement si $\sigma(p) = 0$
- $v_\sigma(p \wedge q) = 1$ si et seulement si $\sigma(p) = 1$ et $\sigma(q) = 1$
- $v_\sigma(p \vee q) = 1$ si et seulement si $\sigma(p) = 1$ ou $\sigma(q) = 1$
- $v_\sigma(p \to q) = 0$ si et seulement si $\sigma(p) = 1$ et $\sigma(q) = 0$
- $v_\sigma(p \leftrightarrow q) = 1$ si et seulement si $\sigma(p) = \sigma(q)$

b) Évaluation d'une formule

Une valuation σ étant fixée, on définit récursivement la **valeur** $v_\sigma(A)$ d'une formule logique A de la façon suivante :

- si $A = p$, variable propositionnelle, alors $v_\sigma(A) = \sigma(p)$
- si $A = \neg B$, alors $v_\sigma(A) = 1 - v_\sigma(B)$
- si $A = B \wedge C$, alors $v_\sigma(A) = v_\sigma(B) \times v_\sigma(C) = \min(v_\sigma(B), v_\sigma(C))$
- si $A = B \vee C$, alors $v_\sigma(A) = v_\sigma(B) + v_\sigma(C) - v_\sigma(B) \times v_\sigma(C) = \max(v_\sigma(B), v_\sigma(C))$
- si $A = B \to C$, alors $v_\sigma(A) = 1 - v_\sigma(B) + v_\sigma(B) \times v_\sigma(C)$
- si $A = B \leftrightarrow C$, alors $v_\sigma(A) = 1 - v_\sigma(B) - v_\sigma(C) + 2v_\sigma(B) \times v_\sigma(C)$

On définit de même l'algorithme d'évaluation d'une formule logique à partir de sa représentation par un arbre.

c) Tables de vérité

Soient A une formule logique et $\{p_1, \dots, p_n\}$ l'ensemble des variables propositionnelles intervenant dans A. Lorsque n n'est pas trop grand (en pratique $n \leqslant 3 \dots$), on regroupe parfois dans un tableau (la *table de vérité* de A) les 2^n distributions de vérité possibles et les valeurs correspondantes prises par A.

Exemple - 1 :

p	$\neg p$
1	0
0	1

p	q	$p \wedge q$
1	1	1
1	0	0
0	1	0
0	0	0

p	q	$p \vee q$
1	1	1
1	0	1
0	1	1
0	0	0

p	q	$p \to q$
1	1	1
1	0	0
0	1	1
0	0	1

p	q	$p \leftrightarrow q$
1	1	1
1	0	0
0	1	0
0	0	1

On peut regrouper dans un même tableau les tables de vérité de plusieurs formules faisant intervenir les mêmes variables propositionnelles. Surtout dans le cas où les différentes colonnes sont des sous formules de la dernière.

Exemple - 2 :

p	q	r	$p \wedge q$	$(p \wedge q) \vee r$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

2) Tautologies et satisfiabilité

Définition II - 2 : Satisfiabilité et modèle

Soit A une formule propositionnelle. On dit que cette formule est **satisfiable** (ou satisfaisable) s'il existe une valuation σ telle que $v_\sigma(A) = 1$.
Dans ce cas σ est appelée **modèle** de A.
L'ensemble des modèles de A est l'ensemble des valuations σ telle que $v_\sigma(A) = 1$.

Définition II - 3 : Tautologie, antilogie

On appelle **tautologie** toute formule propositionnelle A telle que toute distribution de vérité est un modèle pour A. On note parfois $A = \top$.
On appelle **antilogie** ou contradiction, toute formule propositionnelle A telle que pour toute valuation σ, $v_\sigma(A) = 0$. On note parfois $A = \bot$.

Exemple - 3 :
$(p \vee \neg p))$ est une tautologie, $(p \wedge \neg p))$ est une contradiction.

Définition II - 4 : Problème SAT
Soit A une formule propositionnelle. On appelle problème SAT de cette formule la recherche de l'existence d'un modèle pour A.

Remarque : Si A possède n variables propositionnelles, il existe 2^n distributions de vérités possibles. La recherche du problème SAT est donc de complexité exponentielle. On dit que c'est un problème NP-complet.

3) Equivalence sur les formules

Définition II - 5 :
Soient A et B deux formules propositionnelles, $(p_1, \dots, p_n)$ une famille de variables propositionnelles distinctes, contenant toutes celles intervenant dans A ou dans B, les assertions suivantes sont équivalentes :

(**i**) Pour toute distribution de vérité σ, sur $(p_1, \dots, p_n)$, $v_\sigma(A) = v_\sigma(B)$
(**ii**) $(A \leftrightarrow B)$ est une tautologie.

Lorsqu'elles sont vraies, les formules A et B sont dites **équivalentes**.
On écrit $A \equiv B$.
$\equiv$ est une relation d'équivalence sur l'ensemble des formules propositionnelles.

4) Équivalences fondamentales

Si l'on ne s'intéresse qu'à la sémantique, on peut remplacer toute formule logique par une formule équivalente. Ce paragraphe fournit un "catalogue" d'équivalences classiques utile dans de telles transformations.

Soient F_1, F_2, F_3 trois formules logiques. On a les équivalences suivantes (pour alléger, on utilise les symboles évoqués plus haut et l'on omet les parenthèses externes des formules situées de part et d'autre du symbole $\equiv$) :

- $F_1 \wedge F_1 \equiv F_1$; $F_1 \vee F_1 \equiv F_1$ (*idempotence*).
- $F_1 \wedge F_2 \equiv F_2 \wedge F_1$; $F_1 \vee F_2 \equiv F_2 \vee F_1$ (*commutativité*).
- $(F_1 \wedge F_2) \wedge F_3 \equiv F_2 \wedge (F_1 \wedge F_3)$; $(F_1 \vee F_2) \vee F_3 \equiv F_2 \vee (F_1 \vee F_3)$ (*associativité*).
- $(F_1 \wedge F_2) \vee F_3 \equiv (F_1 \vee F_3) \wedge (F_2 \vee F_3)$; $(F_1 \vee F_2) \wedge F_3 \equiv (F_1 \wedge F_3) \vee (F_2 \wedge F_3)$ (*distributivité*).
- $\neg(\neg F_1) \equiv F_1$ (**loi du tiers exclu**).
- $\neg(F_1 \wedge F_2) \equiv (\neg F_1) \vee (\neg F_2)$; $\neg(F_1 \vee F_2) \equiv (\neg F_1) \wedge (\neg F_2)$ (**lois de De Morgan**).
- $F_1 \Rightarrow F_2 \equiv (\neg F_2) \Rightarrow (\neg F_1)$ (*contraposition*). t
- $F_1 \Rightarrow F_2 \equiv (\neg F_1) \vee F_2$ (**expression de l'implication en fonction de la négation et de la disjonction**).

Chacune des équivalences ci-dessus se démontre par simple comparaison des tables de vérités des formules obtenues en remplaçant F_1, F_2, F_3 par des variables propositionnelles p_1, p_2, p_3, ceci en vertu du résultat suivant :

Théorème II - 1 : Invariance de l'équivalence par substitution

Soient G et H deux formules logiques équivalentes, dépendant des variables propositionnelles $p_1, \dots, p_n$ et soient $F_1, \dots, F_n$ des formules logiques. Alors les formules logiques G' et H' obtenues en substituant $F_1, \dots, F_n$ à $p_1, \dots, p_n$ dans G et H sont encore équivalentes.

5) Conséquence logique entre deux formules

Définition II - 6 : Conséquence logique

Soit A et B deux formules propositionnelles.
On dit que B est une conséquence logique de A si tout modèle σ de A est aussi un modèle pour B.
Autrement dit, si $v_\sigma(A) = 1$ alors $v_\sigma(B) = 1$.
On note alors $A \models B$.

Propriété II - 1 :

Deux formules propositionnelles A et B sont équivalentes si et seulement si A est une conséquence logique de B et B est une conséquence logique de A.

Définition II - 7 : Généralisation

Soit Γ un ensemble de formules propositionnelles et A une formule propositionnelle. On dit que A est une conséquence logique de Γ si tout σ qui est un modèle de toute formule de Γ (on parle aussi de modèle de Γ) est aussi un modèle A.
On note $\Gamma \models A$.

III Formes normales d'une formule logique

Compte tenu de l'associativité des opérateurs $\vee$ et $\wedge$, on convient d'écrire sans parenthèses les itérations de la forme $\bigvee_{k=1}^{n} F_k = F_1 \vee \cdots \vee F_n$ ou de la forme $\bigwedge_{k=1}^{n} F_k = F_1 \wedge \cdots \wedge F_n$, où $F_1, \dots, F_n$ sont des formules logiques.

On appellera **disjonction** toute formule logique s'écrivant $F_1 \vee \cdots \vee F_n$, où chaque F_k est un **littéral**, c'est-à-dire soit une variable propositionnelle p_k, soit la négation d'une variable propositionnelle $\neg p_k$.

De même, on appellera **conjonction** toute formule logique s'écrivant $F_1 \wedge \cdots \wedge F_n$, où chaque F_k est un *littéral*.

1) Forme normale disjonctive

Soient F une formule logique et $\{p_1, \dots, p_n\}$ l'ensemble des variables propositionnelles intervenant dans F.

Pour toute application $\sigma : \{p_1, \dots, p_n\} \to \{0,1\}$ et pour tout k de $\mathbb{N}_n$, on définit la conjonction C_σ en posant

$$C_\sigma = F_1 \wedge \cdots \wedge F_n \text{ , avec } \begin{cases} F_k = p_k \text{ si } \sigma(p_k) = 1 \\ F_k = \neg p_k \text{ si } \sigma(p_k) = 0 \end{cases} .$$

On vérifie alors que, pour toute application $\tau : \{p_1, \dots, p_n\} \to \{0,1\}$

$$v_\tau(C_\sigma) = \begin{cases} 1 \text{ si } \tau = \sigma \\ 0 \text{ si } \tau \neq \sigma \end{cases} .$$

Théorème III - 1 : Forme normale disjonctive

Soit A une formule propositionnelle dont les variables propositionnelles sont dans $\{p_1, \dots, p_n\}$. Soit Σ l'ensemble des applications $\sigma : \{p_1, \dots, p_n\} \to \{0,1\}$ telles que $v_\sigma(A) = 1$. (C'est l'ensemble des modèles de A)

A est équivalente à la disjonction $G = \bigvee_{\sigma \in \Sigma} C_\sigma$

(où les C_σ sont les conjonctions définies ci-dessus).

G est appelée **forme normale disjonctive** de A.

Remarque :

1) Toutes les conjonctions C_σ contiennent une fois et une seule chacune des variables propositionnelles de A, soit telle quelle, soit par sa négation.

2) On peut obtenir la forme normale disjonctive à partir de la table de vérité de A. On commence pour cela par supprimer toutes les lignes pour lesquelles l'évaluation de A donne 0 ; il subsiste les lignes correspondant aux éléments σ de l'ensemble Σ ci-dessus ; pour chacune de ces lignes, on construit la conjonction C_σ en prenant chacune des variables propositionnelles, telle quelle si elle a la valeur 1 dans cette ligne, niée sinon.

Exemple :

soit $A = \big(((\neg p) \vee q) \wedge r\big) \Rightarrow \big(p \wedge (\neg q)\big)$; construisons la table de vérité de A :

p	q	r	$(\neg p) \vee q$	$((\neg p) \vee q) \wedge r$	$p \wedge (\neg q)$	A
1	1	1	1	1	0	0
1	1	0	1	0	0	1
1	0	1	0	0	1	1
1	0	0	0	0	1	1
0	1	1	1	1	0	0
0	1	0	1	0	0	1
0	0	1	1	1	0	0
0	0	0	1	0	0	1

L'algorithme ci-dessus fournit :

$$G = \big(p \wedge q \wedge (\neg r)\big) \vee \big(p \wedge (\neg q) \wedge r\big) \vee \big(p \wedge (\neg q) \wedge (\neg r)\big) \vee \big((\neg p) \wedge q \wedge (\neg r)\big) \vee \big((\neg p) \wedge (\neg q) \wedge (\neg r)\big) .$$

2) Forme normale conjonctive

De même toute formule logique A admet une **forme normale conjonctive**, du type $H = \bigwedge_{\sigma \in \Sigma} D_\sigma$, où les D_σ sont des disjonctions, contenant une fois et une seule chacune des variables propositionnelles de A, soit telle quelle, soit par sa négation.

H s'obtient en appliquant les lois de de Morgan à $\neg G'$, où G' est la forme normale disjonctive de $\neg A$. La formule obtenue est équivalente à $\neg(\neg A)$, donc à A ; elle est bien de la forme ci-dessus, Σ étant l'ensemble des applications σ de $\{p_1, \dots, p_n\}$ dans $\{0, 1\}$ telles que $v_\sigma(A) = 0$.
Exemple : reprenons l'exemple précédent ; l'algorithme appliqué à $\neg A$ (c'est-à-dire aux lignes pour lesquelles l'évaluation de F donne 0) fournit la forme normale disjonctive de $\neg A$:

$$G' = \big(p \wedge q \wedge r\big) \vee \big((\neg p) \wedge q \wedge r\big) \vee \big((\neg p) \wedge (\neg q) \wedge r\big)\,.$$

On « morganise »pour en déduire la forme normale conjonctive de F :

$$H = \big((\neg p) \vee (\neg q) \vee (\neg r)\big) \wedge \big(p \vee (\neg q) \vee (\neg r)\big) \wedge \big(p \vee q \vee (\neg r)\big)\,.$$

3) Problème SAT

Définition III - 1 :
Soit V un ensemble fini de variables propositionnelles. On appelle clause une disjonction de littéraux sur V, chaque variable apparaissant au plus une fois.
On dit que c'est une p-clause si p variables apparaissent exactement.

Définition III - 2 :
Le problème p-SAT pour $p > 0$ est la restriction du problème SAT sur les conjonctions de p-clauses.

Proposition III - 1 :
Le nombre de p-clauses sur un ensemble de n variables est polynomial en n, à p fixé

On peut supposer que les p-clauses d'une instance du problème p-SAT toutes différentes (il est très facile de supprimer les doublons). On est donc ramené à la question : « Existe t-il un algorithme de complexité $O(n^k)$ pour le problème p-SAT, avec $k > 0$? »
Les réponses à cette question sont :
- si $p = 1$, oui : il suffit que la formule ne contienne pas à la fois p_i et $\neg p_i$;
- si $p = 2$, oui ;
- si $p \geqslant 3$: on ne sait pas. Celui qui résoudra la question recevra 1 million de dollars !