(Oral Blanc).

NOM : **Prénom :** **Jury :**

Algèbre ← Entourez l'épreuve → **Analyse**

Sujet choisi : 916 : Formule du calcul propositionnel : représentation, formes normales, satisfiabilité. Applications.

Autre sujet : 924 : Théories et modèles en logique du 1^{er} ordre. Exemples.

①

I Définitions

<u>Def 1</u> : On note $\mathcal{V}$ l'ensemble des variables propositionnelles c'est à dire $\mathcal{V} = \{p, q, r, \dots\}$ est un ensemble dénombrable de lettres.

• On note $\mathcal{L}_p$ l'ensemble des formules du langage propositionnel sur $\mathcal{V}$. $\mathcal{L}_p$ est définit inductivement :

(i) tout $p \in \mathcal{V}$ est une formule

(ii) $\bot$ et $\top$ sont des formules (elles représentent le vrai et le faux)

(iii) Si F_1 et F_2 sont des formules et $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ alors $(\neg F_1)$ et $(F_1 * F_2)$ sont des formules.

<u>Ex 2</u> : • $(\neg((p \vee q) \rightarrow (p_1 \wedge r)))$ est une formule.

• pqv n'est pas une formule.

<u>Def 2</u> : • Les formules atomiques sont $\top, \bot$ et p pour $p \in \mathcal{V}$.

• On définit inductivement l'ensemble des sous-formules

- $sousf(A) = \{A\}$ si A est atomique
- $sousf(\neg A) = \{\neg A\} \cup sousf(A)$
- $sousf(A_1 * A_2) = \{A_1 * A_2\} \cup sousf(A_1) \cup sousf(A_2)$.

<u>Rq 3</u> : Selon la situation, on peut vouloir représenter les formules de différentes manières. En voici des exemples.

<u>Def 4</u> : Soit $A \in \mathcal{L}_p$. Arb_A est l'arbre syntaxique de A. C'est un arbre binaire (G, R, D) définit par induction :

- Si A est atomique alors $Arb_A = \langle \phi, A, \phi \rangle$
- Si $A = \neg A_1$ alors $Arb_A = \langle Arb_{A_1}, \neg, \phi \rangle$.
- Si $A = A_1 * A_2$ alors $Arb_A = \langle Arb_{A_1}, *, Arb_{A_2} \rangle$ où $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

<u>Prop 5</u> : On a le théorème de lecture unique :
Pour tout $F \in \mathcal{L}_p$, on est dans un seul de ces trois cas :

(i) $F \in \mathcal{V}$ ou $F = \bot$ ou $F = \top$ (les trois cas s'excluent).

(ii) il existe une unique formule G tq $F = \neg G$

(iii) il existe un unique couple (F_1, F_2) et un unique $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ tel que $F = F_1 * F_2$.

<u>Ex 6</u> : Si dans ~~Def~~ 1 on avait à la place de (iii) :

(iii') Si F_1 et F_2 sont des formules et $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ alors $\neg F_1$ et $F_1 * F_2$ sont des formules alors on aurait pas de théorème de lecture unique.

<u>Coro 7</u> : Arb_A est unique.

<u>Def 8</u> : La notation polonaise d'une formule est définie inductivement par :

$pol(\top) = \top$; $pol(\bot) = \bot$; $pol(p) = p$ pour $p \in \mathcal{V}$;
$pol(\neg A) = \neg\, pol(A)$ et $pol(F_1 * F_2) = *\, pol(F_1)\, pol(F_2)$.

<u>Rq 9</u> : La notation polonaise permet de se passer de parenthèse.

<u>Def 10</u> : Une formule sans $\rightarrow$ et $\leftrightarrow$ peut être représentée par un circuit logique avec les portes :
—o— : "non", =D— "et" ; ⊐D— "ou".

<u>Rq 11</u> : Voir annexe 1 pour l'équivalence de ces définitions.

①

II Sémantique : Donner du sens à une formule

Def 11 : On note $Bool = \{0,1\}$. Une fonction booléenne à n arguments est une fonction $f : Bool^n \to Bool$.
On définit la fonction unaire $f_\neg$ par $f_\neg(0) = 1$ et $f_\neg(1) = 0$
On définit les fonctions binaires $f_\wedge, f_\vee, f_\rightarrow, f_\leftrightarrow$ par la table suivante :

Argument 1	Argument 2	$f_\vee$	$f_\wedge$	$f_\rightarrow$	$f_\leftrightarrow$
1	1	1	1	1	1
1	0	1	0	0	0
0	1	1	0	1	0
0	0	0	0	1	1

Def 12 : • Une valuation est une fonction $\mathcal{I} : \mathcal{V} \to Bool$ (ou interprétation)
• Soit $\mathcal{I}$ une valuation on étend $\mathcal{I}$ en $\tilde{\mathcal{I}} : \mathcal{L}_p \to Bool$ par induction :
- Si $p \in \mathcal{V}$ alors $\tilde{\mathcal{I}}(p) = \mathcal{I}(p)$
- $\tilde{\mathcal{I}}(\top) = 1$ et $\tilde{\mathcal{I}}(\bot) = 0$
- $\tilde{\mathcal{I}}(\neg A) = f_\neg(\tilde{\mathcal{I}}(A))$
- Si $* \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$ alors $\tilde{\mathcal{I}}(A_1 * A_2) = f_*(\tilde{\mathcal{I}}(A_1), \tilde{\mathcal{I}}(A_2))$

Ex 13 : On représente ~~les~~ valuations sous forme de tableau
Voir annexe 2

Def 14 : • $\mathcal{I}$ est un modèle de A ssi $\tilde{\mathcal{I}}(A) = 1$. On note $\mathcal{I} \models A$.
• Une formule A (ou un ensemble de formules E) est satisfiable si il existe un modèle de A (ou de toute formule de G).
• Si A (ou E) n'a pas de modèle, il est dit insatisfiable.
• A est valide si $\mathcal{I} \models A$ pour tout $\mathcal{I}$. On note alors $\models A$.

Ex 15 : • $p \wedge (\neg p)$ est insatisfiable.
• $(p \vee \neg p)$ (tiers exclu) et $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$ (loi de De Morgan) sont valides.
• $(p \wedge q)$ est satisfiable mais n'est pas valide.
• $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$ est valide.

Théo 16 : Soient A et A' des formules et $p \in \mathcal{V}$.
Si A est valide alors $A[x := A']$ est valide.

Def 17 : Soient E et G des ensembles de formules. On dit que G est une conséquence logique de E (noté $E \models G$) si tout modèle de E est modèle de G.

Ex 18 : • $\emptyset \models E$ ssi $\models E$ (E est valide).
• $p, q \models p \wedge q$; $p \wedge q \models q$; $p \rightarrow q, q \rightarrow r \models p \rightarrow r$.

Def 19 : Soient A_1 et A_2 des formules. A_1 et A_2 sont logiquement équivalentes (noté $A_1 \equiv A_2$) si $A_1 \models A_2$ et $A_2 \models A_1$.

Ex 19 : • $A_1 \equiv A_2$ ssi $\models A_1 \leftrightarrow A_2$.
• $A \wedge B \equiv B \wedge A$; $A \vee \bot \equiv A$; $A \wedge \top \equiv A$;
• $A \vee (G \wedge H) \equiv (A \vee G) \wedge (A \vee H)$ et $A \wedge (G \vee H) \equiv (A \wedge G) \vee (A \wedge H)$.

Théo 20 : Soient A une formule, $B \in sousf(A)$ et C tq $B \equiv C$.
Si on note A' la formule où on a remplacé B par C dans A alors $A' \equiv A$.

Ex 21 : $(\neg(\neg p \vee q)) \wedge (s \vee \neg s) \equiv (p \wedge \neg q) \wedge (s \vee \neg s) \equiv (p \wedge \neg q) \wedge \top$
$\equiv p \wedge \neg q$.

III Mise sous formes logiquement équivalentes

① Modification du nombre de connecteurs

3

Def 23 : Soit A une formule dont les variables sont $p_1 \dots p_n$; $n \geq 0$.
Soit $f_A : Bool^n \to Bool$ définie par $f_A(v_1 \dots v_n) = \hat{I}(A)$ où $I(p_i) = v_i$ pour tout i.
On dit que f_A est réalisée par A.

Ex 24 : f_7 est réalisé par $\neg p$; $\neg\neg\neg p$; $\neg p (p \vee \neg p)$ ...

Def 25 : Un ensemble C de connecteurs muni de leurs interprétations fonctionnelles f_c pour $c \in C$ est dit fonctionnellement complet par rapport à E où E est un ensemble de fonctions booléennes si toute fonction de E est réalisée par une formule utilisant les connecteurs de C.

Théo 26 : • $\{\neg, \vee, \wedge\}$ est fonctionnellement complet par rapport à l'ensemble de toutes les fonctions booléennes.
• $\{\neg, \vee\}$, $\{\neg, \wedge\}$, $\{\neg, \to\}$ le sont aussi.

Ex 27 : •

p_1	p_2	$f(p_1,p_2)$
1	1	1
1	0	0
0	1	1
0	0	0

f est réalisée par : $(p_1 \wedge p_2) \vee (\neg p_1 \wedge p_2)$

• On note $\downarrow$ le non-ou. où f_1 est définie par :

p_1	p_2	$f_1(p_1,p_2)$
1	1	0
1	0	0
0	1	0
0	0	1

$\{\downarrow\}$ est fonctionnellement complet.

Rq 28 : Soient $x = \overline{x_1 \dots x_n}^2$ et $y = \overline{y_1 \dots y_n}^2$ des entiers écrits en base 2.
$f : Bool^{2n} \to Bool$ qui donne le $i^{ème}$ bit de la somme $(1 \leq i \leq 2n+2)$ peut être représentée avec une formule booléenne.

Dev 1

Prop 29 : À l'aide du paradigme diviser pour régner, on peut construire un additionneur n-bits (où $n = 2^q$) avec $O(n \log n)$ portes et qui fonctionne en temps $T(n) = 3 \cdot (1 + \log_2(n))$.

(b) les formes normales

Def 30 : Une formule est sous forme normale de négation (f.n.n) si toute occurence de $\neg$ s'applique à une sous-formule atomique.

Prop 31 : Toute formule est logiquement équivalente à une formule sous forme f.n.n.

Def 32 : • Un littéral est une formule atomique ou la négation d'une formule atomique.
• Une clause est une formule de la forme $(l_1 \vee \dots l_m)$ où les l_i sont des littéraux
• Une formule A est dite sous forme normale conjonctive (f.n.c) si elle est de la forme $D_1 \wedge \dots \wedge D_q$ où les D_i sont des clauses.

Ex 33 : • $(x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2 \vee \bar{x}_3)$ est une f.n.c.
• On peut définir les formes normales disjonctives.

Théo 34 : Toute formule est logiquement équivalente à une formule sous forme f.n.c.

Def 35 : k-SAT est le problème de décision suivant :
Entrée : F une formule f.n.c tq toute clause est de taille k.
Sortie : F est-elle satisfiable ?

Prop 36 : 2-SAT $\in$ P

Dev 2

Théo 37 (Cook) : 3-SAT $\in$ NP-C.

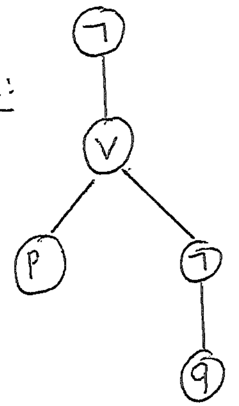
3

4

Annexe 1 :

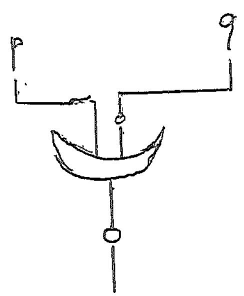
la formule $(\neg(p\vee(\neg q)))$ est représentée par :

Def 4 :



Def 8 : $\neg \vee p \neg q$

Def 10 :



Annexe 2 :

Pour I : $p \mapsto 1$
$q \mapsto 1$

p	q	$\neg q$	$p\vee(\neg q)$	$(\neg(p\vee(\neg q)))$
1	1	0	1	0

5