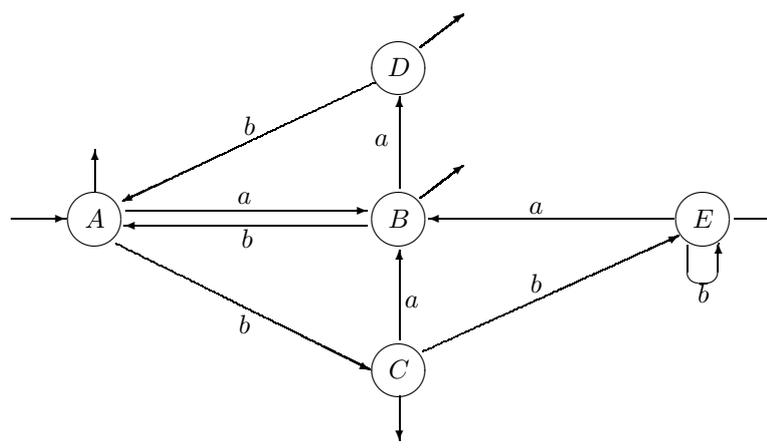


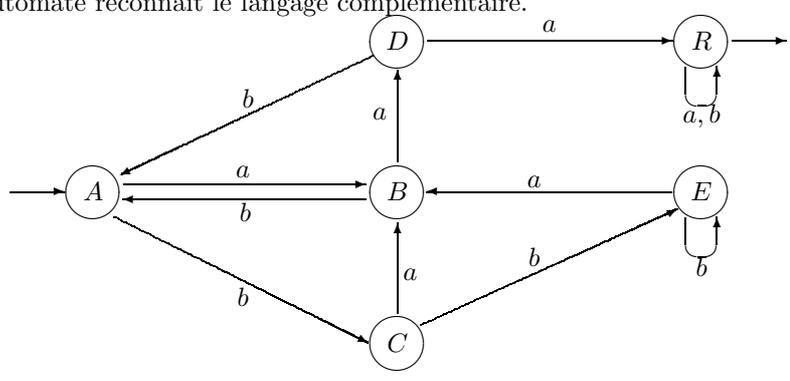
1 Exercice sur les automates

1. Détermination

	a	b
$A = \{p, r\}$	$\{q, r\}$	$\{p, q\}$
$B = \{q, r\}$	$\{q\}$	$\{p, r\}$
$C = \{p, q\}$	$\{q, r\}$	$\{p, q, r\}$
$D = \{q\}$		$\{p, r\}$
$E = \{p, q, r\}$	$\{q, r\}$	$\{p, q, r\}$



2. Complétons l'automate déterministe précédent en ajoutant un état rebut R et considérons l'automate "complémentaire" qui s'en déduit en rendant terminal l'état R et non terminal les autres états. On sait que ce nouvel automate reconnaît le langage complémentaire.



On constate que lorsqu'on arrive en l'état final R pour la première fois si et seulement si on vient de lire un mot se terminant par aaa . Ainsi le langage complémentaire de $L(\mathcal{A})$ est égal à l'ensemble des mots contenant aaa , d'où la conclusion pour $L(\mathcal{A})$.

3. Posons $E = b^*(aa^*bb^*)^*a^*$ et montrons que $A^* = L(E)$, c'est à dire en fait $A^* \subset L(E)$.
 Montrons par récurrence sur la longueur n d'un mot $m \in A^*$ qu'il est dans $L(E)$.
 . C'est immédiat si $n = 0$ et $n = 1$.
 . Supposons que tous les mots de longueur $n - 1$ sont dans $L(E)$.
 Soit u un mot de longueur n .
 * S'il se termine par a , alors il s'écrit $u = v.a$ avec v de longueur $n - 1$.
 D'après l'hypothèse de récurrence, $v \in L(b^*(aa^*bb^*)^*a^*)$, donc $u = v.c$ aussi.
 * S'il se termine par b , alors il s'écrit $u = v.b$ avec v de longueur $n - 1$.
 Si v se termine par a , alors $v \in L(b^*(aa^*bb^*)^*aa^*)$, donc $u \in L(b^*(aa^*bb^*)^*aa^*b) \subset L(b^*(aa^*bb^*)^*aa^*bb^*) \subset L(E)$.
 Si v se termine par b , alors $v \in L(b^*(aa^*bb^*)^*aa^*bb^*)$, donc $u \in L(b^*(aa^*bb^*)^*aa^*bb^*) \subset L(E)$.
4. Les mots ne contenant pas aaa sont tels que aa^* ne peut prendre que les valeurs ε , a et aa .
 Une expression rationnelle du langage $L(\mathcal{A})$ est donc : $b^*((a + aa)bb^*)^*(\varepsilon + a + aa)$.

2 Problème de logique

On notera \mathcal{V} l'ensemble des valuations (applications de \mathcal{A} dans \mathcal{B}).

1. $\underline{\forall i \in \mathcal{V}, I_i(F \Rightarrow G) = 1} \iff \forall i \in \mathcal{V}, \overline{I_i(F)} + I_i(G) = 1 \iff \forall i \in \mathcal{V}, (\overline{I_i(F)} = 1 \vee I_i(G) = 1)$
 $\iff \forall i \in \mathcal{V}, (I_i(F) = 0 \vee I_i(G) = 1)$
 $\iff \forall i \in \mathcal{V}, (I_i(F) = 1 \Rightarrow I_i(G) = 1)$ car $A \Rightarrow B \equiv (\neg A \vee B)$.

Ainsi $F \Rightarrow G$ est une tautologie si et seulement si chaque fois que $I_1(F) = 1$, alors $I_i(G) = 1$.

- 2 Supposons que $F \Rightarrow G$ soit une implication élémentaire.

- . Si \mathbf{f} figure dans F , alors $\forall i \in \mathcal{V}, I_i(F) = 0$, donc $I_i(F \Rightarrow G) = 1$.
- . Si \mathbf{v} figure dans G , alors $\forall i \in \mathcal{V}, I_i(G) = 1$, donc $I_i(F \Rightarrow G) = 1$.
- . Sinon, si F et G n'ont pas de variables en commun, alors pour toute valuation i mettant à 1 les variables de F et à 0 celles de G , alors $I_i(F) = 1$ et $I_i(G) = 0$, donc $I_i(F \Rightarrow G) = 0$.

Par contre, si F et G ont au moins une variable x en commun, pour toute valuation i :

- si $i(x) = 0$, alors $I_i(F) = 0$, donc $I_i(F \Rightarrow G) = 1$.
- si $i(x) = 1$, alors $I_i(G) = 1$, donc $I_i(F \Rightarrow G) = 1$.

Bilan : $F \Rightarrow G$ est une tautologie dans trois cas :

- ou bien \mathbf{f} figure dans F
- ou bien \mathbf{v} figure dans G
- ou bien \mathbf{f} ne figure pas dans F et \mathbf{v} ne figure pas dans G et F et G ont au moins une variable commune.

Ainsi, dans les 4 implications proposées par l'énoncé, seule la (b) n'est pas une tautologie.

3. Supposons $A \equiv B$. Alors $\forall i \in \mathcal{V}, I_i(A) = I_i(B)$, d'où $I_i(A \Rightarrow G) = \overline{I_i(A)} + I_i(G) = \overline{I_i(B)} + I_i(G) = I_i(B \Rightarrow G)$, donc $(A \Rightarrow G) \equiv (B \Rightarrow G)$.

4. Analogie car $I_i(F \Rightarrow A) = \overline{I_i(F)} + I_i(A) = \overline{I_i(F)} + I_i(B) = I_i(F \Rightarrow B)$.

Conséquence : si $A \equiv B$ et $A' \equiv B'$, alors $A \Rightarrow B \equiv A' \Rightarrow B'$, ce qui exprime que la relation d'équivalence \equiv est compatible avec le connecteur \Rightarrow .

5. $\forall i \in \mathcal{V}, I_i((A \wedge \mathbf{v}) \Rightarrow B) = \overline{I_i(A \wedge \mathbf{v})} + I_i(B) = \overline{I_i(A).I_i(\mathbf{v})} + I_i(B) = \overline{I_i(A).1} + I_i(B) = \overline{I_i(A)} + I_i(B) = I_i(A \Rightarrow B)$.

Ainsi $(A \wedge \mathbf{v}) \Rightarrow B \equiv A \Rightarrow B$.

6. $\forall i \in \mathcal{V}, I_i(A \Rightarrow (B \vee \mathbf{f})) = \overline{I_i(A)} + I_i(B \vee \mathbf{f}) = \overline{I_i(A)} + I_i(B) + 0 = \overline{I_i(A)} + I_i(B) = I_i(A \Rightarrow B)$.

Ainsi $(A \wedge \mathbf{f}) \Rightarrow B \equiv A \Rightarrow B$.

7. $\forall i \in \mathcal{V}, I_i(F \Rightarrow (G \vee \neg A)) = \overline{I_i(F)} + I_i(G \vee \neg A) = \overline{I_i(F)} + I_i(G) + I_i(\neg A) = \overline{I_i(F)} + I_i(G) + \overline{I_i(A)}$
 $= \overline{I_i(F)}.I_i(A) + I_i(G) = \overline{I_i(F \wedge A)} + I_i(G) = I_i((F \wedge A) \Rightarrow G)$,

donc $F \Rightarrow (G \vee \neg A) \equiv (F \wedge A) \Rightarrow G$.

Pour $G = \mathbf{f}$, on obtient : $F \Rightarrow \neg A \equiv F \Rightarrow (\mathbf{f} \vee \neg A) \equiv (F \wedge A) \Rightarrow \mathbf{f}$.

8. En remplaçant A par $\neg A$ dans 7, on obtient : $(F \wedge \neg A) \Rightarrow G \equiv F \Rightarrow (G \vee \neg(\neg A)) \equiv F \Rightarrow (G \vee A)$ puisque la relation \equiv est compatible avec les connecteurs.

Pour $F = \mathbf{v}$ et avec le résultat de 5, on obtient : $\neg A \Rightarrow G \equiv (\mathbf{v} \wedge \neg A) \Rightarrow G \equiv \mathbf{v} \Rightarrow (G \vee A)$.

9. $\forall i \in \mathcal{V}, I_i((F \wedge (A \cup B)) \Rightarrow G) = I_i(G) + \overline{I_i(F).(I_i(A) + I_i(B))} = I_i(G) + \overline{I_i(F \wedge A) + I_i(F \wedge B)}$
 $= I_i(G) + \overline{I_i(F \wedge A).I_i(F \wedge B)}$.

$I_i((F \wedge (A \cup B)) \Rightarrow G) = 1 \iff I_i(G) = 1$ ou $(\overline{I_i(F \wedge A)} = 1$ et $\overline{I_i(F \wedge B)} = 1)$

$\iff (I_i(G) = 1$ ou $\overline{I_i(F \wedge A)} = 1)$ et $(I_i(G) = 1$ ou $\overline{I_i(F \wedge B)} = 1)$

$\iff I_i((F \wedge A) \Rightarrow G) = 1$ et $I_i((F \wedge B) \Rightarrow G) = 1$.

Sachant qu'on peut "distribuer" \forall sur un "et", on en déduit que :

$F \wedge (A \cup B) \Rightarrow G$ est une tautologie si et seulement si $(F \wedge A) \Rightarrow G$ et $(F \wedge B) \Rightarrow G$ sont des tautologies.

En prenant $F = \mathbf{v}$ et en utilisant le résultat du **5**, on obtient immédiatement que :
 $(A \cup B) \Rightarrow G$ est une tautologie si et seulement si $A \Rightarrow G$ et $B \Rightarrow G$ sont des tautologies.

10. $F \Rightarrow (G \vee (A \wedge B)) \equiv F \Rightarrow (G \vee \neg(\neg A \vee \neg B)) \equiv (F \wedge (\neg A \vee \neg B)) \Rightarrow G$ (d'après **7**).

Ainsi $F \Rightarrow (G \vee (A \wedge B))$ est une tautologie si et seulement si $(F \wedge (\neg A \vee \neg B)) \Rightarrow G$ en est une, ce qui équivaut, d'après **9**, au fait que $(F \wedge \neg A) \Rightarrow G$ et $(F \wedge \neg B) \Rightarrow G$ sont des tautologies, c'est à dire, d'après **8**, que $(F \Rightarrow (G \vee A))$ et $(F \Rightarrow (G \vee B))$ sont des tautologies.

En particulier, pour $G = \mathbf{f}$ et en utilisant **6**, on obtient immédiatement que :

$F \Rightarrow (A \wedge B)$ est une tautologie si et seulement si $F \Rightarrow A$ et $F \Rightarrow B$ sont des tautologies.

11. $F \Rightarrow (G \vee (A \Rightarrow B)) \equiv F \Rightarrow (G \vee (\neg A \vee B)) \equiv F \Rightarrow ((G \vee B) \vee \neg A) \equiv (F \wedge A) \Rightarrow (G \vee B)$ d'après **7**.

En particulier, pour $G = \mathbf{f}$ et en utilisant **6**, on obtient immédiatement que :

$F \Rightarrow (A \Rightarrow B) \equiv (F \wedge A) \Rightarrow B$.

12. $F \wedge (A \Rightarrow B) \Rightarrow G \equiv (F \wedge (\neg A \vee B)) \Rightarrow G$ est une tautologie d'après **9**. si et seulement si $(F \wedge \neg A) \Rightarrow G$ et $(F \wedge B) \Rightarrow G$ sont des tautologies, c'est à dire d'après **8**. $F \Rightarrow (G \vee A)$ et $(F \wedge B) \Rightarrow G$ sont des tautologies.

En particulier, pour $F = \mathbf{v}$ et en utilisant **5**, on obtient immédiatement que :

$(A \Rightarrow B) \Rightarrow G$ est une tautologie si et seulement si $B \Rightarrow G$ et $\mathbf{v} \Rightarrow (G \vee A)$ sont des tautologies.

13. Récapitulons les règles obtenues dans les questions précédentes.

R_1	$(\neg A) \Rightarrow G$	$\mathbf{v} \Rightarrow (G \vee A)$	R_7	$F \Rightarrow \neg A$	$(F \wedge A) \Rightarrow \mathbf{f}$
R_2	$(F \wedge \neg A) \Rightarrow G$	$F \Rightarrow (G \vee A)$	R_8	$F \Rightarrow (G \vee \neg A)$	$(F \wedge A) \Rightarrow G$
R_3	$(A \vee B) \Rightarrow G$	$\begin{cases} A \Rightarrow G \\ B \Rightarrow G \end{cases}$	R_9	$F \Rightarrow (A \wedge B)$	$\begin{cases} F \Rightarrow A \\ F \Rightarrow B \end{cases}$
R_4	$(F \wedge (A \vee B)) \Rightarrow G$	$\begin{cases} (F \wedge A) \Rightarrow G \\ (F \wedge B) \Rightarrow G \end{cases}$	R_{10}	$F \Rightarrow (G \vee (A \wedge B))$	$\begin{cases} F \Rightarrow (G \vee A) \\ F \Rightarrow (G \vee B) \end{cases}$
R_5	$(A \Rightarrow B) \Rightarrow G$	$\begin{cases} B \Rightarrow G \\ \mathbf{v} \Rightarrow (G \vee A) \end{cases}$	R_{11}	$F \Rightarrow (A \Rightarrow B)$	$(F \wedge A) \Rightarrow B$
R_6	$(F \wedge (A \Rightarrow B)) \Rightarrow G$	$\begin{cases} (F \wedge B) \Rightarrow G \\ F \Rightarrow (G \vee A) \end{cases}$	R_{12}	$F \Rightarrow (G \vee (A \Rightarrow B))$	$(F \wedge A) \Rightarrow (G \vee B)$

On veut déterminer si $(F \Rightarrow G)$ est une tautologie.

Pour cela on considère la liste $L_1 = (F_1, \dots, F_m)$ des formules atomiques ou non de sorte que $F = F_1 \wedge \dots \wedge F_m$ et la liste $L_2 = (G_1, \dots, G_n)$ des formules atomiques ou non de sorte que $G = G_1 \vee \dots \vee G_n$.

- Si L_1 contient au moins une formule non atomique F_i
 - . si F_i est de la forme $A \wedge B$, on enlève F_i et on rajoute les formules A et B à L_1 .
 - . si F_i est de la forme $\neg A$, on applique la règle R_1 si F_i est la seule formule non atomique de L_1 ou la règle R_2 sinon.
 - . si F_i est de la forme $A \vee B$, on applique la règle R_4 , ce qui ramène à l'étude de deux tautologies sous forme d'une implication.
 - . si F_i est de la forme $A \Rightarrow B$, on applique la règle R_5 si F_i est la seule formule non atomique de L_1 ou la règle R_6 sinon, ce qui ramène à l'étude de deux tautologies sous forme d'une implication.
- Si L_2 contient au moins une formule non atomique G_j
 - . si G_j est de la forme $A \vee B$, on enlève G_j et on rajoute les formules A et B à L_2 .
 - . si G_j est de la forme $\neg A$, on applique la règle R_7 si G_j est la seule formule non atomique de L_2 ou la règle R_8 sinon.
 - . si G_j est de la forme $A \wedge B$, on applique la règle R_{10} , ce qui ramène à l'étude de deux tautologies sous forme d'une implication.
 - . si G_j est de la forme $A \Rightarrow B$, on applique la règle R_{11} si G_j est la seule formule non atomique de L_2 ou la règle R_{12} sinon, ce qui ramène à l'étude de deux tautologies sous forme d'une implication.

Chaque application d'une des 12 règles supprime un des trois connecteurs "indésirables" de l'une des listes L_1 et L_2 , et cela, sans en faire réapparaître dans l'autre liste

On recommence tant que l'une des deux listes L_1 et L_2 contiennent au moins une formule non atomique.

On verra plus loin que l'algorithme se termine.

On est alors ramené à une ou plusieurs implications élémentaires dont on sait vérifier si ce sont des tautologies (cf. **2.**)

14. Comme $F \equiv \mathbf{f} \vee F \equiv \mathbf{v} \Rightarrow F$, il suffit donc d'appliquer l'algorithme de la question **13** à $\mathbf{v} \Rightarrow F$.

15. Comme $\neg F \equiv \neg F \vee \mathbf{f} \equiv F \Rightarrow \mathbf{f}$, il suffit donc d'appliquer l'algorithme de la question **13** à $F \Rightarrow \mathbf{f}$.

16. $\mu_i(F \Rightarrow G) = \mu_g(F) + \mu_d(G)$.

(a) si les y_i et z_j sont des formules atomiques, alors :

$$\mu_i((y_1 \wedge \dots \wedge y_m) \Rightarrow (z_1 \vee \dots \vee z_n)) = \mu_g(y_1 \wedge \dots \wedge y_m) + \mu_d(z_1 \vee \dots \vee z_n) = \sum_{i=1}^m \mu_c(y_i) + \sum_{j=1}^n \mu_c(z_j) = 0 + 0 = 0$$

(valable même si $m = 1$ ou $n = 1$).

(b) Si $F = F_1 \wedge \dots \wedge F_m$ et $G = G_1 \wedge \dots \wedge G_n$ avec éventuellement $m = 1$ ou $n = 1$, alors :

$$\mu_g(F \wedge G) = \sum_{i=1}^m \mu_c(F_i) + \sum_{j=1}^n \mu_c(G_j) = \mu_g(F) + \mu_g(G).$$

$$\text{De plus } \mu_c(F) = (m - 1) + \sum_{i=1}^m \mu_c(F_i) = m - 1 + \mu_g(F) \geq \mu_g(F).$$

(c) Si $F = F_1 \vee \dots \vee F_m$ et $G = G_1 \vee \dots \vee G_n$ avec éventuellement $m = 1$ ou $n = 1$, alors :

$$\mu_d(F \vee G) = \sum_{i=1}^m \mu_c(F_i) + \sum_{j=1}^n \mu_c(G_j) = \mu_d(F) + \mu_d(G).$$

$$\text{De plus } \mu_c(F) = (m - 1) + \sum_{i=1}^m \mu_c(F_i) = m - 1 + \mu_d(F) \leq \mu_d(F).$$

(d) L'application à l'implication $F \Rightarrow G$ de chacune des 12 règles ramène à une ou deux implications dont la i -mesure est strictement inférieure. Vérifions-le pour quelques unes d'entre elles.

$$\begin{aligned} \cdot \mu_i((F \wedge \neg A) \Rightarrow G) &= \mu_g(F) + \mu_g(\neg A) + \mu_d(G) = \mu_g(F) + \mu_d(G) + \mu_c(A) + 1 \\ &> \mu_g(F) + \mu_d(G) + \mu_d(A) = \mu_i((F \Rightarrow (G \vee A))). \end{aligned}$$

$$\begin{aligned} \cdot \mu_i((F \wedge (A \vee B)) \Rightarrow G) &= \mu_g(F) + \mu_c(A \vee B) + \mu_d(G) \\ &> \mu_g(F) + \mu_c(A) + \mu_d(G) \geq \mu_g(F) + \mu_g(A) + \mu_d(G) = \mu_i((F \wedge A) \Rightarrow G). \end{aligned}$$

$$\begin{aligned} \cdot \mu_i(F \Rightarrow (G \vee (A \Rightarrow B))) &= \mu_g(F) + \mu_d(G) + \mu_c(A \Rightarrow B) \\ &> \mu_g(F) + \mu_d(G) + \mu_c(A) + \mu_c(B) \geq \mu_g(F) + \mu_g(A) + \mu_d(G) + \mu_d(B) = \mu_i((F \wedge A) \Rightarrow \\ &(G \vee B)). \end{aligned}$$

La suite des i -mesures étant à valeurs entières positives strictement décroissante est nécessairement finie. Donc l'algorithme de la question **13.** s'arrête, ce qui se produit si et seulement si l'implication obtenue est atomique.

17 $T1 = (x_1 \Rightarrow (x_2 \Rightarrow x_3)) \Rightarrow ((x_1 \Rightarrow x_2) \Rightarrow (x_1 \Rightarrow x_3))$.

L'application de la règle 5 donne $T_1 \equiv (T_2 \wedge T_3)$ avec :

$$T_2 = (x_2 \Rightarrow x_3) \Rightarrow ((x_1 \Rightarrow x_2) \Rightarrow (x_1 \Rightarrow x_3)) \quad \text{et} \quad T_3 = \mathbf{v} \Rightarrow x_1 \vee ((x_1 \Rightarrow x_2) \Rightarrow (x_1 \Rightarrow x_3)),$$

A son tour T_2 se décompose en $T_2 \equiv (T_4 \wedge T_5)$ avec :

$$T_4 = x_3 \Rightarrow ((x_1 \Rightarrow x_2) \Rightarrow (x_1 \Rightarrow x_3)) \quad \text{et} \quad T_5 = \mathbf{v} \Rightarrow x_2 \vee ((x_1 \Rightarrow x_2) \Rightarrow (x_1 \Rightarrow x_3)).$$

En utilisant R_{12} , $T_3 \equiv (\mathbf{v} \wedge (x_1 \Rightarrow x_2)) \Rightarrow (x_1 \vee (x_1 \Rightarrow x_3))$ qui se décompose d'après R_6 en $T_2 \equiv (T_6 \wedge T_7)$ avec :

$$T_6 = (\mathbf{v} \wedge x_2) \Rightarrow (x_1 \vee (x_1 \Rightarrow x_3)) \quad \text{et} \quad T_7 = \mathbf{v} \Rightarrow (x_1 \vee (x_1 \Rightarrow x_3)).$$

On a immédiatement, en utilisant R_{12} : $T_6 \equiv (\mathbf{v} \wedge x_2 \wedge x_1) \Rightarrow (x_1 \vee x_3)$ qui est une implication élémentaire car la variable x_1 figure à la fois dans les deux membres, donc c'est une tautologie.

$T_7 \equiv (\mathbf{v} \wedge x_1) \Rightarrow (x_1 \vee x_3)$ qui est une tautologie.

D'après R_{11} , $T_4 \equiv (x_3 \wedge (x_1 \Rightarrow x_2)) \Rightarrow (x_1 \Rightarrow x_3)$ qui se décompose d'après R_6 en $T_2 \equiv (T_8 \wedge T_9)$ avec :

$T_8 = (x_3 \wedge x_2) \Rightarrow (x_1 \Rightarrow x_3)$ et $T_9 = x_3 \Rightarrow x_1 \vee (x_1 \Rightarrow x_3)$.

D'après R_{11} , $T_8 = (x_3 \wedge x_2 \wedge x_1) \Rightarrow x_3$ qui est une tautologie.

D'après R_{12} , $T_9 = (x_3 \wedge x_1) \Rightarrow (x_1 \vee x_3)$ qui est une tautologie.

D'après R_{12} , $T_5 \equiv (\mathbf{v} \wedge (x_1 \Rightarrow x_2)) \Rightarrow (x_2 \vee (x_1 \Rightarrow x_3))$ qui se décompose d'après R_6 en en $T_{10} \wedge T_{11}$ avec :

$T_{10} = (\mathbf{v} \wedge x_2) \Rightarrow (x_2 \vee (x_1 \Rightarrow x_3)) \equiv (\mathbf{v} \wedge x_2 \wedge x_1) \Rightarrow (x_2 \vee x_3)$ (d'après R_{12}) : c'est une tautologie.

$T_{11} = \mathbf{v} \Rightarrow (x_1 \vee x_2 \vee (x_1 \Rightarrow x_3)) \equiv (\mathbf{v} \wedge x_1) \Rightarrow (x_1 \vee x_2 \vee x_3)$ (d'après R_{12}) : c'est une tautologie.

Au total, T_1 est bien une tautologie.

3 Problème de programmation en Caml

```

type formule = | vrai
               | faux
               | variable of int
               | et       of formule * formule
               | ou       of formule * formule
               | implique of formule * formule
               | non      of formule ;;

type liste_de_formules == formule list ;;

let liste_vide () = let l = [vrai] in tl(l) ;;

let ajoute (f:formule) l = f::l ;;

let est_vide l = l = tl[vrai] ;;

let un_element (l:formule list) = match l with
  | [] -> failwith "Liste vide1"
  | t::q -> t ;;

let reste (l:formule list) = match l with
  | [] -> failwith "Liste vide2"
  | t::q -> q ;;

type ensemble_de_formules = { V : bool;
                              F : bool;
                              tbl : bool vect;
                              lf : formule list } ;;

let ensemble_vide () = { V = false;
                        F = false;
                        tbl = make_vect VMAX false;
                        lf = [] } ;;

let que_des_variables e = e.lf=[] ;;

let intersection e1 e2 =
  let rec aux i =

```

```

    if i = VMAX then false
    else (e1.tbl.(i) && e2.tbl.(i)) || aux (i+1)
in aux 0 ;;

let ajouter_formule e f = match f with
|      vrai  -> {V = true; F = e.F;  tbl = e.tbl; lf = e.lf}
|      faux  -> {V = e.V ; F = true; tbl = e.tbl; lf = e.lf}
| variable(i) -> let t = e.tbl in t.(i) <- true;
                  {V = e.V; F = e.F; tbl = t; lf = e.lf}
|      _     -> {V = e.V; F = e.F; tbl = e.tbl; lf = ajoute f e.lf}
;;

let formule_suivante e =
  ({V = e.V; F = e.F; tbl = e.tbl; lf = reste e.lf}, un_element e.lf) ;;

let rec verifier u v =
  if que_des_variables u && que_des_variables v
  then u.F || v.V || intersection u v
  else if not (que_des_variables u)
    then let (uu,f) = formule_suivante u in match f with

| et(a,b) -> verifier (ajouter_formule (ajouter_formule uu a) b) v

| non a    -> let w = if que_des_variables uu
                    then (* R1 *) ajouter_formule uu vrai
                    else (* R2 *) uu
                in verifier w (ajouter_formule v a)

| ou(a,b) -> (* R3 et R4 *)
              verifier (ajouter_formule uu a) v
              && verifier (ajouter_formule uu b) v

| implique(a,b) -> let w = if que_des_variables uu
                        then (* R5 *) ajouter_formule uu vrai
                        else (* R6 *) uu
                    in  verifier (ajouter_formule uu b) v
                        && verifier w (ajouter_formule v a)

| _ -> failwith "Erreur 1"

    else if not (que_des_variables v)
      then let (vv,g) = formule_suivante v in match g with

| ou(a,b) -> verifier u (ajouter_formule (ajouter_formule vv a) b)

| non a    -> let w = if que_des_variables vv
                    then (* R7 *) ajouter_formule vv faux
                    else (* R8 *) vv
                in verifier (ajouter_formule u a) w

| et(a,b) -> (* R9 et R10 *)
              verifier u (ajouter_formule vv a)
              && verifier u (ajouter_formule vv a)

| implique(a,b) -> (* R11 et R12 *)
                  verifier(ajouter_formule u a)(ajouter_formule vv b)

| _ -> failwith "Erreur 2"

```

```
else failwith "Erreur 3" ;;

let tautologie f =
  verifier (ajouter_formule (ensemble_vide()) vrai)
    (ajouter_formule (ensemble_vide()) f) ;;
```

Fin du corrigé