Alexandre DIKOVSKY
Université de Nantes

Logique Mathématique pour l'Informatique

Cours 3. Méthodes et systèmes formels de preuve.

3.1. Méthode de Quine. La méthode est basée sur les transformations équivalentes utilisant les équivalences de la liste 2 :

$$
\begin{array}{ll}
(e_{2,1})\ \neg 0 \equiv 1 & (e_{2,7})\ 1 \to \phi \equiv \phi \\
(e_{2,2})\ \neg 1 \equiv 0 & (e_{2,8})\ 0 \to \phi \equiv 1 \\
(e_{2,3})\ 1 \wedge \phi \equiv \phi & (e_{2,9})\ \phi \to 1 \equiv 1 \\
(e_{2,4})\ 0 \wedge \phi \equiv 0 & (e_{2,10})\ \phi \to 0 \equiv \neg\phi \\
(e_{2,5})\ 1 \vee \phi \equiv 1 & (e_{2,11})\ \phi \leftrightarrow 1 \equiv \phi \\
(e_{2,6})\ 0 \vee \phi \equiv \phi & (e_{2,12})\ \phi \leftrightarrow 0 \equiv \neg\phi.
\end{array}
$$

Remarque. Si en cours de transformation équivalente

$$\phi \equiv\!\!\Rightarrow_{liste_2} \psi$$

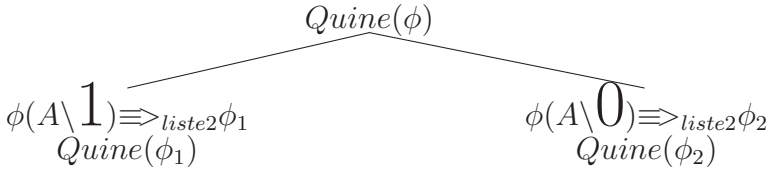
les équivalences sont appliquées uniquemment dans le sens de gauche à droite, alors $|\phi| > |\psi|$ (ce qui raccourcie les formules).

Soit un contexte I.
$I_{A=1}(X) = \begin{cases} I(X), & si\ X \neq A, \\ 1, & sinon \end{cases}$ et $I_{A=0}(X) = \begin{cases} I(X), & si\ X \neq A, \\ 0, & sinon \end{cases}$

Proposition 1. *(Sur les pseudoconstantes* $1, 0$ *et les valeurs booléennes)*
$\phi(A\backslash 1)^I = (\phi)^{(I_{A=1})}$ *et* $\phi(A\backslash 0)^I = (\phi)^{(I_{A=0})}$ *pour toute formule* ϕ *et tout contexte* I.

Ainsi on peut utiliser $0, 1$ comme les constantes booléennes dans les formules.

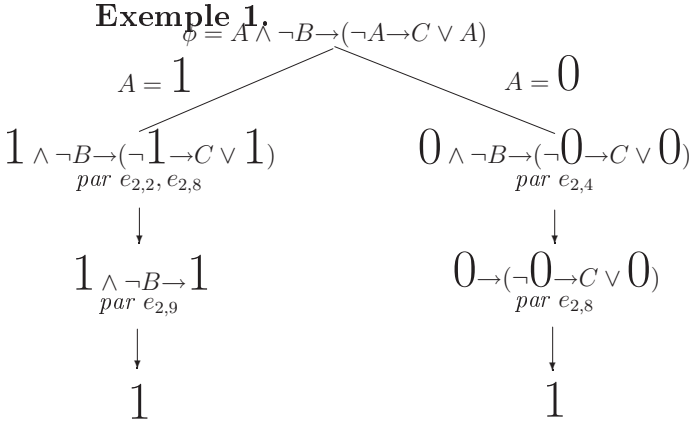
Méthode de Quine : $\phi \equiv\!\!\Rightarrow_{liste2} \psi$: ψ est le résultat des transformation droites de ϕ par équivalences de la liste 2 jusqu'à la saturation.



Proposition 2.
(1) La méthode se converge sur toute formule ϕ*, c'est-à-dire, dans toute branche chaque lettre est éliminée et* ϕ *est réduite à* 1 *ou à* 0*;*
(2) $\models \phi$ *(CONTR(*ϕ*)) ssi toute branche réduit* ϕ *à* 1 *(respectivement à* 0*);*
(3) SAT(ϕ*) ssi il existe une branche qui réduit* ϕ *à* 1*.*

Les diagrammes de Quine décrivent explicitement les modèles des formules et souvent ils sont plus compacts que les TV. Par exemple, $\{A \mapsto 1\} \models A \wedge \neg B \rightarrow (\neg A \rightarrow C \vee A)$, $\{A \mapsto 0\} \models A \wedge \neg B \rightarrow (\neg A \rightarrow C \vee A)$. Ainsi $\models A \wedge \neg B \rightarrow (\neg A \rightarrow C \vee A)$.

Exemple 1.

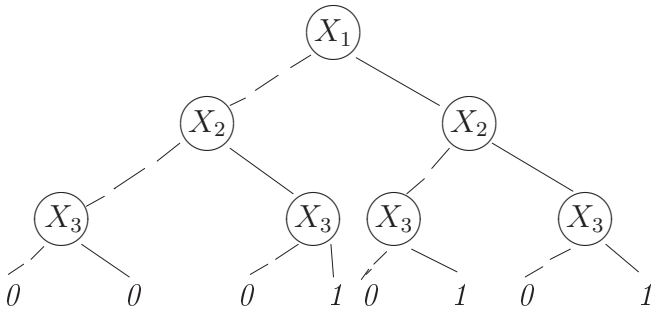


3.2. Définition des modèles par les diagrammes optimaux : OBDD.

On peut présenter les TV comme les arbres d'une taille exponentielle (par rapport au nombre des variables) :

Exemple 2.

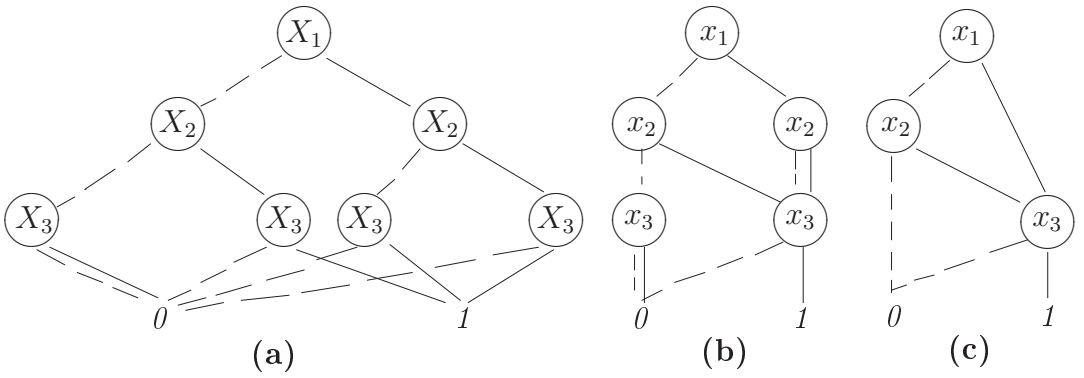
X_1	X_2	X_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



(Ligne continue correspond à $x_i = 1$, ligne pointillée à $x_i = 0$).

Évidemment, ce diagramme peut être optimisé en utilisant les règles suivantes.
Notation : $SD_0(u)$: sous diagramme qui correspond au lien $x = 0$ d'un nœud u étiquetté par la variable $var(u) = x$; $SD_1(u)$: le sous diagramme correspondant pour $x = 1$.
R_1. **Terminaux redondants :** fusionner tous les nœuds 1 et tous les nœuds 0 et rediriger respectivement les liens (voir ex. 3 (a)).
R_2. **Nonterminaux redondants :** si $var(u) = var(v)$, $SD_0(u) = SD_0(v)$ et $SD_1(u) = SD_1(v)$, alors on supprime un des nœuds u, v (voir ex. 3 (b)).
R_3. **Tests redondants :** si $SD_0(u) = SD_1(u)$ pour un nœud u, alors on supprime u (voir ex. 3 (c)).

Exemple 3.

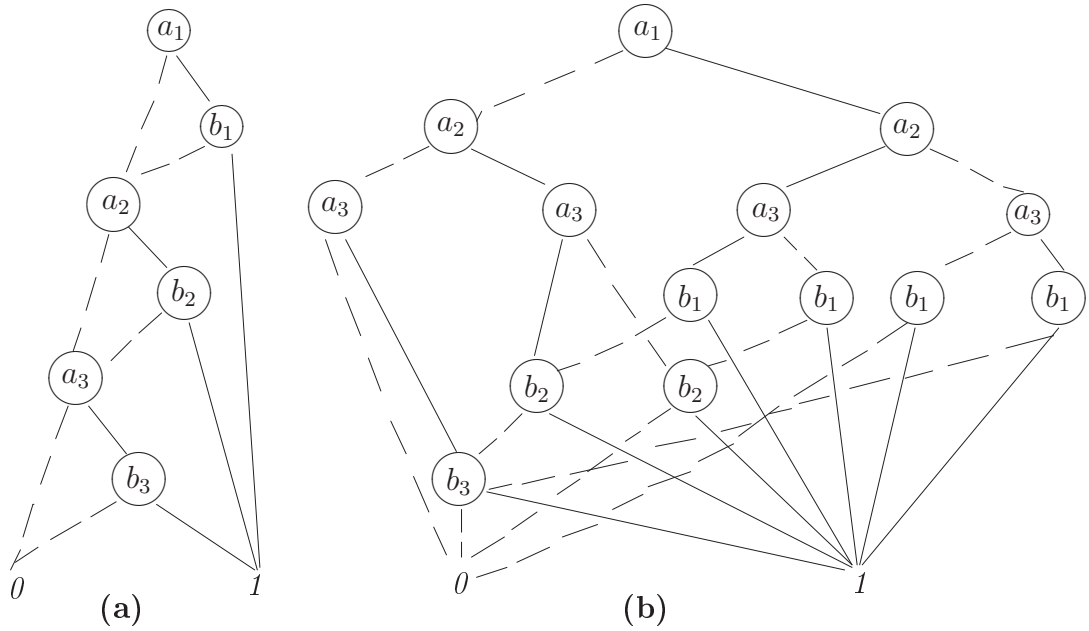


Définition 1. *Le diagramme optimal pour une FB f et* **pour un ordre fixe sur les variables de** *f s'appelle* **OBDD** *(*ordered binary decision diagram*).*

Proposition 3. *1. Pour un ordre fixe sur les variables, deux OBDD de la même FB f sont isomorphes.*
2. OBDD de f définit les mêmes modèles que la TV de f.

Attention : pour les ordres différents des variables, les OBDD peuvent être différents !

Exemple 4.



Les diagrammes de l'ex. 4 sont deux OBDD de la FB exprimée par la formule

$$a_1 \wedge b_1 \vee a_2 \wedge b_2 \vee a_3 \wedge b_3.$$

Le diagramme **(a)** correspond à l'ordre $a_1 < b_2 < a_2 < b_2 < a_3 < b_3$ et le diagramme **(b)** correspond à l'ordre $a_1 < a_2 < a_3 < b_1 < b_2 < b_3$. Le premier a une taille **linéaire** par rapport au nombre des variables, tandis que le deuxième a une taille **exponentielle.**
Ce n'est pas toujours le cas qu'il existe un ordre qui donne un OBDD d'une taille linéaire.

Par exemple, on peut prouver que pour le schéma de multiplication tous les ordres donnent un OBDD d'une taille exponentielle. Mais souvent les ordres optimisant existent.

Question : Comment construire un OBDD d'une formule propositionnelle ϕ **dans un ordre de croissance de variables** (et non à partir de sa TV) ?
Réponse : par récurrence sur les sous formules de ϕ dans le sens "premier dans la profondeur" selon l'ordre donné des variables, en utilisant le principe suivant de Boole-Shannon :

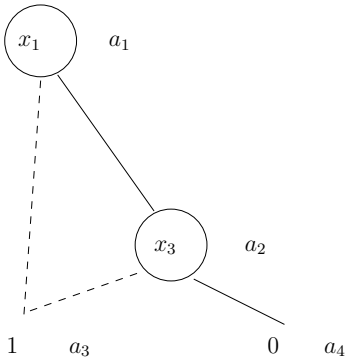
$$f < op > g \equiv \neg x \wedge (f|_{x=0} < op > g|_{x=0}) \vee x \wedge (f|_{x=1} < op > g|_{x=1}) \quad (1)$$

où $< op >$ et la conjonction ou la disjonction et :

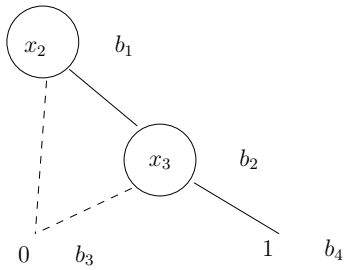
$$f|_{x=b} = \begin{cases} f, & si\ x < var(racine_f), \\ SD_0(racine_f), & x = var(racine_f),\ b = 0, \\ SD_1(racine_f), & x = var(racine_f),\ b = 1, \end{cases}$$

et en appliquant les règles $R_1 - R_3$ après chaque appel récursif. Voici un exemple :

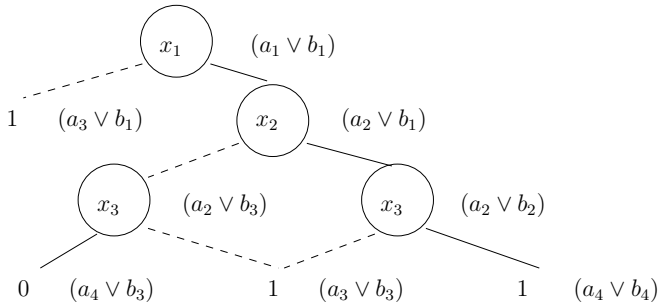
Exemple 5. *Soit la formule $\neg(x_1 \wedge x_3) \vee x_2 \wedge x_3$ et l'ordre $x_1 < x_2 < x_3$. Soit déjà calculés l'OBDD (a) pour la sous formule $\neg(x_1 \wedge x_3)$ et l'OBDD (b) pour la sous formule $x_2 \wedge x_3$. Alors voici un pas de calcule de l'OBDD pour cette formule :*



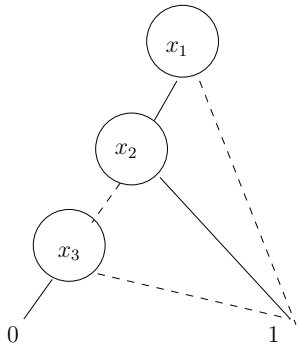
(a) $\neg(x_1 \wedge x_3)$



(b) $(x_2 \wedge x_3)$



(c) *calcule*



(d) $\neg(x_1 \wedge \neg x_2 \wedge x_3)$

Dans cet exemple **(a)** et **(b)** sont les OBDD pour les sous formules, **(c)** est le graphe d'évaluation défini par (1), et **(d)** est l'OBDD après l'optimisation par les règles $R_1 - R_3$.

L'intérêt de cette méthode est qu'un OBDD de ϕ pour un ordre $<$ sur ses variables est calculé **en temps polynomial** : $\mathcal{O}(|OBDD(\phi,<)|^2)$ par rapport à la taille $|OBDD(\phi,<)|$. **Si on devine l'ordre optimal** $<_0$, alors les problèmes $\models \phi$ et $SAT(\phi)$ deviennent décidables en temps $|OBDD(\phi,<_0)| = \mathcal{O}((|\phi|))$. On obtient alors un algorithme quadratique.

3.3. Cohérence et forme clausale.

Définition 2. *Un ensemble* Γ *de formules est cohérent s'il a un modèle.*
Γ_1 *et* Γ_2 *sont équivalents (noté* $\Gamma_1 \equiv \Gamma_2$*) s'ils ont les mêmes modèles.*
Γ_1 *et* Γ_2 *sont équicohérents (noté* $\Gamma_1 \equiv_{ec} \Gamma_2$*) quand* Γ_1 *est cohérent ssi* Γ_2 *l'est aussi.*

Proposition 4. *(Principe de cohérence) :* $\Gamma \models \phi$ *ssi* $\Gamma \cup \{\neg\phi\}$ *est non-cohérent.*

Exemple 6.
(i) $\{(\neg A \wedge \neg B \vee C \wedge D)\} \equiv \{\neg A \vee C, \neg B \vee C, \neg A \vee D, \neg B \vee D\}$ *(lois de distribution).*
(ii) $\{A \vee C, \neg B \vee C, \neg A \vee D, \neg B \vee D\} \equiv_{ec} \{C \vee D, \neg B \vee C, \neg B \vee D\}$ $(I_1(C) = I_1(D) = 1$ *satisfait les deux,* $I_2(A) = I_2(C) = 1$, $I_2(B) = I_2(D) = 0$ *satisfait seulement le deuxième).*
(iii) $\{A,\ A{\rightarrow}B,\ \neg B\}$ *est non-cohérent. Alors,* $\{A,\ A{\rightarrow}B\} \models B$.

Définition 3. *Une clause est une formule de la forme :* $l_1 \vee ... \vee l_k, k \geq 0,$ *où* l_i *sont des littéraux. La clause vide* $(k = 0)$ *est notée* $\Box$*. Elle est équivalente à* 0 *(donc, si* $\Box \in \Gamma$*, alors* Γ *est non-cohérent).*

F*orme* **C***lausale (FC) de* ϕ *est un ensemble* Γ *de clauses qui est équivalent à* ϕ *(*$\{\phi\} \equiv \Gamma$*).*

Remarque. $\bigwedge_{i=1}^{k} \phi_i \equiv \{\phi_1, ..., \phi_k\}$. Alors, FNC $\bigwedge_{i=1}^{k} D_i$ est équivalente à la FC $\{D_1, ..., D_k\}$.

Corollaire 1. *Chaque formule* ϕ *(ou un ensemble de formules) peut être transformée en sa FC équivalente.*

Exemple 7. $\Gamma = \{B, C, B \wedge C{\rightarrow}E \vee A, E \wedge B{\rightarrow}D, A \wedge B{\rightarrow}D\} \equiv$
$C(\Gamma) = \{B, C, \neg B \vee \neg C \vee E \vee A, \neg E \vee \neg B \vee D, \neg A \vee \neg B \vee D\}$.

3.4. Déduction par réfutation.

Principe de réfutation : Soit une relation binaire $\rightarrow_\alpha$ sur les ensembles des formules (une règle de preuve). Supposons que $\Gamma_1 \rightarrow_\alpha \Gamma_2$ implique $\Gamma_1 \models \Gamma_2$ pour tout Γ_1, Γ_2 (on dit dans ce cas que la règle $\rightarrow_\alpha$ est **correcte**. Pour prouver $\Gamma \models \phi$ en utilisant la règle correcte $\rightarrow_\alpha$, il suffit d'établir une suite d'ensembles de formules :

$$\Gamma \cup \{\neg\phi\} = \Gamma_0 \rightarrow_\alpha ... \rightarrow_\alpha \Gamma_n,$$

où Γ_n est non-cohérent. Alors, $\Gamma \cup \{\neg\phi\}$ l'est aussi et selon le principe de cohérence, $\Gamma \models \phi$.

Effectivement : notons $Mod(\Gamma)$ l'ensemble des modèles de Γ. Selon la correction de $\rightarrow_\alpha$, $\Gamma_i \rightarrow_\alpha \Gamma_{i+1}$ implique $Mod(\Gamma_i) \subseteq Mod(\Gamma_{i+1})$. On obtient alors :

$$Mod(\Gamma \cup \{\neg\phi\}) = Mod(\Gamma_0) \subseteq Mod(\Gamma_1) \subseteq \ldots \subseteq Mod(\Gamma_n) = \emptyset.$$

Ainsi, $Mod(\Gamma \cup \{\neg\phi\}) = \emptyset$ et $\Gamma \models \phi$ selon le principe de cohérence.

3.4.1. Méthode des tableaux.

Classification des formules.

$\wedge$-formules	membres de $< \phi_1 \wedge \phi_2 >$
$\phi_1 \wedge \phi_2$	ϕ_1, ϕ_2
$\phi_1 \leftrightarrow \phi_2$	$\phi_1 \rightarrow \phi_2, \phi_2 \rightarrow \phi_1$
$\neg(\phi_1 \vee \phi_2)$	$\neg\phi_1, \neg\phi_2$
$\neg(\phi_1 \rightarrow \phi_2)$	$\phi_1, \neg\phi_2$

$\vee$-formules	membres de $< \phi_1 \vee \phi_2 >$
$\phi_1 \vee \phi_2$	ϕ_1, ϕ_2
$\phi_1 \rightarrow \phi_2$	$\neg\phi_1, \phi_2$
$\neg(\phi_1 \wedge \phi_2)$	$\neg\phi_1, \neg\phi_2$
$\neg(\phi_1 \leftrightarrow \phi_2)$	$\neg(\phi_1 \rightarrow \phi_2), \neg(\phi_2 \rightarrow \phi_1)$

Tableau : Un ensemble de *branches* $\mathcal{T} = \{\mathcal{B}_1, ..., \mathcal{B}_k\}$.
Branche : Une liste de formules $\mathcal{B} = [\phi_1, ..., \phi_l]$.
$\mathcal{B}$ *fermée* : $\psi, \neg\psi \in \mathcal{B}$ pour une ψ.

Sémantique des tableaux : $\| \mathcal{B} \| = \bigwedge_{\phi \in \mathcal{B}} \phi, \quad \| \mathcal{T} \| = \bigvee_{\mathcal{B} \in \mathcal{T}} \| \mathcal{B} \|$

Exemple 8. *Pour* $\mathcal{T}_0 = \{[\ \neg(P \rightarrow Q), \neg(R \rightarrow S)\ ], [\ P \wedge R \rightarrow (Q \vee S)\ ]\}$,
$\| \mathcal{T}_0 \| = \neg(P \rightarrow Q) \wedge \neg(R \rightarrow S) \vee (P \wedge R \rightarrow (Q \vee S))$

Extension d'une branche dans $\mathcal{T}$:

$$\frac{\{\mathcal{T}', [\ \mathcal{B}',\ < \phi \wedge \psi >\ ]\}}{\{\mathcal{T}', [\ \mathcal{B}',\ < \phi \wedge \psi >,\ \phi,\ \psi\ ]\}}$$

Ramification d'une branche dans $\mathcal{T}$:

$$\frac{\{\mathcal{T}', [\ \mathcal{B}',\ < \phi \vee \psi >\ ]\}}{\{\mathcal{T}', [\ \mathcal{B}',\ < \phi \vee \psi >,\ \phi\ ],\ [\ \mathcal{B}',\ < \phi \vee \psi >,\ \psi\ ]\}}$$

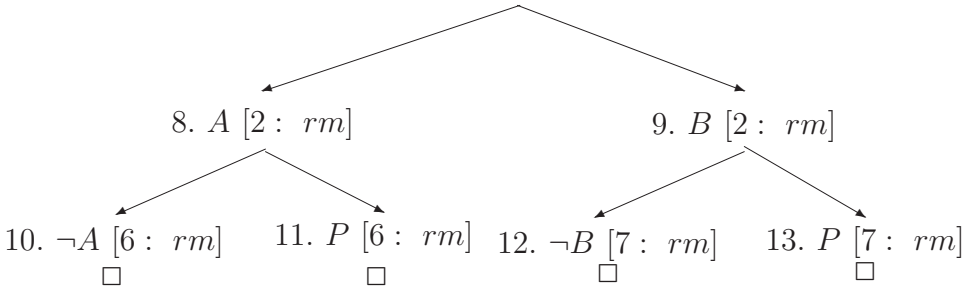
Proposition 5. *1. (Correction)*
$\| \{\mathcal{T}', [\ \mathcal{B}',\ < \phi \wedge \psi >\ ]\} \| \models \| \{\mathcal{T}', [\ \mathcal{B}',\ < \phi \wedge \psi >,\ \phi,\ \psi\ ]\} \|$,
$\| \{\mathcal{T}', [\ \mathcal{B}',\ < \phi \vee \psi >\ ]\} \| \models \| \{\mathcal{T}', [\ \mathcal{B}',\ < \phi \vee \psi >,\ \phi\ ],\ [\ \mathcal{B}',\ < \phi \vee \psi >,\ \psi\ ]\} \|$.
2. Les pas d'extension et de ramification d'une branche préservent l'ensemble des modèles de $\mathcal{T}$.

Méthode : Pour prouver $\Gamma \vdash_t \phi$:
(1) Initialiser le premier tableau : $\mathcal{T}_0 = \{\ [\Gamma, \neg\phi]\ \}$.
(2) Etablir une *dérivation* $\mathcal{T}_0 \rightarrow_t ... \rightarrow_t \mathcal{T}_n$.
Un pas $\mathcal{T}_{n-1} \rightarrow_t \mathcal{T}_n, (n \geq 1)$: c'est soit
- extension d'une branche $\mathcal{B} \in \mathcal{T}_{n-1}$ en utilisant une $\wedge$-formule $< \phi_1, \phi_2 > \in \mathcal{B}$,
soit
- ramification d'une branche $\mathcal{B} \in \mathcal{T}_{n-1}$ en utilisant une $\vee$-formule $< \phi_1, \phi_2 > \in \mathcal{B}$.

(3) $\Gamma \vdash_t \phi$: dans cette dérivation, le dernier tableau $\mathcal{T}_n$ est *fermé* (toute sa branche est fermée).

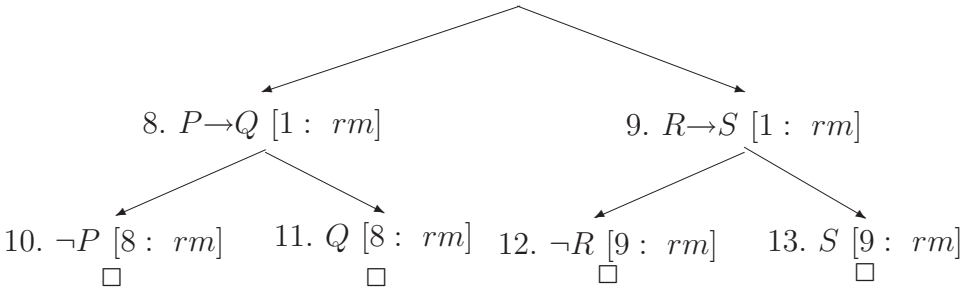
Exemple 9. $\models (A \vee B) \rightarrow ((A \rightarrow P) \wedge (B \rightarrow P) \rightarrow P)$

1. $\neg((A \vee B) \rightarrow ((A \rightarrow P) \wedge (B \rightarrow P) \rightarrow P))$
2. $A \vee B$ [1 : ex]
3. $\neg((A \rightarrow P) \wedge (B \rightarrow P) \rightarrow P)$ [1 : ex]
4. $(A \rightarrow P) \wedge (B \rightarrow P)$ [3 : ex]
5. $\neg P$ [3 : ex]
6. $A \rightarrow P$ [4 : ex]
7. $B \rightarrow P$ [4 : ex]



Exemple 10. $\{(P \rightarrow Q) \vee (R \rightarrow S), P \wedge R\} \models (Q \vee S)$

1. $(P \rightarrow Q) \vee (R \rightarrow S)$
2. $P \wedge R$
3. $\neg Q \wedge \neg S$
4. P [2 : ex]
5. R [2 : ex]
6. $\neg Q$ [3 : ex]
7. $\neg S$ [3 : ex]



Théorème 1. *(Correction et complétude).* $\Gamma \vdash_t \phi$ *ssi* $\Gamma \models \phi$.

Remarques . 1. Si on n'applique jamais une règle deux fois à une même sous formule, alors chaque tableau est transformé à un tableau terminal.
2. Si dans ce tableau terminal il y a une branche non fermée, alors elle est satisfaisable (le modèle est défini à partir des sous formules littéraux). Selon la proposition 5.2 ce modèle est celui de $\Gamma \cup \{\neg\phi\}$.

Corollaire 2. *Si le tableaux terminal n'est pas fermé, alors* $\Gamma \not\models \phi$.

3.4.2. Méthode de résolution (appliquée aux formes clausales).

Règle de résolution [A. Robinson] :
Pour deux FC : Γ_1, Γ_2, $\Gamma_1 \rightarrow_r \Gamma_2$ si
$\Gamma_1 = \Gamma \cup \{Cl_1 \vee l, Cl_2 \vee \neg l\}$ et $\Gamma_2 = \Gamma_1 \cup \{Cl_1 \vee Cl_2\}$ ($Cl_1 \vee Cl_2$: *résolvante*).
$\Gamma_0 \rightarrow_r ... \rightarrow_r \Gamma_n$ est une *preuve* si $\square \in \Gamma_n$.

Proposition 6. *(Correction). Si* $\Gamma_1 \rightarrow_r \Gamma_2$, *alors* $\Gamma_1 \models \Gamma_2$.
Corollaire 3. *Si* $\Gamma_1 \rightarrow_r ... \rightarrow_r \Gamma_n$ *est une preuve, alors* Γ_1 *est non-cohérent.*
Méthode : Pour prouver $\Gamma \vdash_r \phi$:
(1) Former la FC initiale $\Gamma_0 = \Gamma \cup \{ \neg\phi \}$.
(2) Etablir une *dérivation* $\Gamma_0 \rightarrow_r ... \rightarrow_r \Gamma_n$.
(3) $\Gamma \vdash_r \phi$, si cette dérivation est une preuve.

Exemple 11. *Prouver* $\Gamma = \{(P \rightarrow Q) \vee (R \rightarrow S), P \wedge R\} \vdash_r Q \vee S$.
La forme clausale de $\Gamma \cup \{\neg(Q \vee S)\}$ *:* $\{\neg P \vee Q \vee \neg R \vee S, P, R, \neg Q, \neg S\}$.
Une preuve :
1. $\neg P \vee Q \vee \neg R \vee S$
2. P
3. R
4. $\neg Q$
5. $\neg S$
6. $Q \vee \neg R \vee S$ $[1, 2]$
7. $Q \vee S$ $[6, 3]$
8. S $[7, 4]$
9. $\square$ $[8, 5]$

Théorème 2. *(Correction et complétude).* $\Gamma \vdash_r \phi$ *ssi* $\Gamma \models \phi$.
Exemple 12. *Pour* $\phi = (A \vee B) \rightarrow ((A \rightarrow P) \wedge (B \rightarrow P) \rightarrow P)$ *prouver* $\models \phi$.
(1) Transformation de $\neg\phi$ *en forme clausale :*
$\neg((A \vee B) \rightarrow ((A \rightarrow P) \wedge (B \rightarrow P) \rightarrow P)) \equiv$
$(A \vee B) \wedge \neg((A \rightarrow P) \wedge (B \rightarrow P) \rightarrow P)) \equiv$
$(A \vee B) \wedge (A \rightarrow P) \wedge (B \rightarrow P) \wedge \neg P))$.
$\Gamma_0 = \{A \vee B, \neg A \vee P, \neg B \vee P, \neg P\}$.
(2) Résolution.
1. $A \vee B$
2. $\neg A \vee P$
3. $\neg B \vee P$
4. $\neg P$
5. $\neg B$ $[3, 4]$
6. A $[1, 5]$
7. P $[2, 6]$
8. $\square$ $[4, 7]$

Appelons Γ_i *terminal* si on ne peut plus y rajouter une nouvelle résolvante.

Proposition 7. *1. Si* $\Gamma_1 \rightarrow_r \Gamma_2$, *alors* $\Gamma_1 \equiv \Gamma_2$ *(ils ont les mêmes modèles).*
2. Si $\Gamma_0 = FCL(\Gamma \cup \{\neg\phi\}) \rightarrow_r ... \rightarrow_r \Gamma_n$, Γ_n *est terminal et* $\square \notin \Gamma_n$, *alors* $\Gamma \not\models \phi$.