Méthode de Davis-Putnam-Logemann-Loveland

Principe des méthodes clausales par réfutation

- On n'utilise pas la formule initiale en entrée ;
- On nie la formule (on prend sa négation) ;
- On la met sous forme clausale (ensemble de clauses) ;
- Une clause est disjonction de littéraux ;
- Un littéral est un axiome ou la négation d'un axiome ;
- Un axiome est une variable (propositionnelle) ;
- Puis on cherche si la forme clausale est insatisfiable ;
- Si elle l'est alors sa négation (la formule initiale) est valide.

Mise en forme clausale

Règles de transformation

$$\neg\neg F \rightarrow F \qquad \neg\top \rightarrow \bot \qquad \neg\bot \rightarrow \top$$
$$\neg(F_1 \wedge F_2) \rightarrow \neg F_1 \vee \neg F_2 \qquad \neg(F_1 \vee F_2) \rightarrow \neg F_1 \wedge \neg F_2$$
$$F_1 \Rightarrow F_2 \rightarrow \neg F_1 \vee F_2$$
$$F_1 \wedge \top \rightarrow F_1 \qquad \top \wedge F_1 \rightarrow F_1 \qquad F_1 \wedge \bot \rightarrow \bot \qquad \bot \wedge F_1 \rightarrow \bot$$
$$F_1 \vee \top \rightarrow \top \qquad \top \vee F_1 \rightarrow \top \qquad F_1 \vee \bot \rightarrow F_1 \qquad \bot \vee F_1 \rightarrow F_1$$
$$(F_1 \wedge F_2) \vee F_3 \rightarrow (F_1 \vee F_3) \wedge (F_2 \vee F_3)$$
$$F_3 \vee (F_1 \wedge F_2) \rightarrow (F_3 \vee F_1) \wedge (F_3 \vee F_2)$$

La règle de base de DPLL : le « splitting »

Principe du « splitting »

- On considère un ensemble de clauses S et une variable A ;
- On note $S[A := \top]$ l'ensemble obtenu en enlevant de S toutes les clauses contenant le littéral A et en effaçant le littéral $\neg A$ dans les clauses restantes ;
- On note $S[A := \bot]$ l'ensemble obtenu en enlevant de S toutes les clauses contenant le littéral $\neg A$ et en effaçant le littéral A dans les clauses restantes ;
- On notera que S est satisfiable ssi $S[A := \top]$ ou $S[A := \bot]$ l'est ;
- On notera que S est insatisfiable ssi $S[A := \top]$ et $S[A := \bot]$ le sont ;
- On peut « splitter » selon toutes les variables de S, mais on retombe exactement sur la méthode naïve, qui est exponentielle ;
- Le but de DPLL est donc d'éviter à tout prix le « splitting ».

La règle de base de DPLL : le « splitting »

Simplifications implicites

- On considère des simplifications implicites lorsqu'on réalise les opérations $S[A := \top]$ et $S[A := \bot]$;
- Si $\top$ appartient à une clause alors cette clause est éliminée de S (du coup, S peut devenir vide et il sera satisfiable) ;
- Si $\bot$ appartient à une clause alors on élimine $\bot$ de la clause (du coup, on peut avoir une clause vide $\Box$ et S sera alors insatisfiable).

Une règle de simplification : la résolution unitaire

Principe de la résolution unitaire

- Une clause est unitaire si elle contient un unique littéral A ou $\neg A$;
- Si $A \in S$ alors on peut remplacer S par $S[A := \top]$;
- En effet, si $A \in S$, S est satisfiable ssi $S[A := \top]$ l'est ;
- Si $\neg A \in S$ alors on peut remplacer S par $S[A := \bot]$;
- En effet, si $\neg A \in S$, S est satisfiable ssi $S[A := \bot]$ l'est.

Une règle de simplification : les clauses pures

Principe des clauses pures

- Un atome A est pur dans S ssi il apparaît toujours avec le même signe ;
- Autrement dit, pour un atome A, soit $A \notin S$, soit $\neg A \notin S$;
- On dira que A ou $\neg A$ est un littéral pur dans S ;
- Une clause contenant un littéral pur sera dite pure ;
- Si P est le sous-ensemble des clauses pures de S, alors on peut remplacer S par $S \backslash P$;
- En effet, S est satisfiable ssi $S \backslash P$ l'est.

Une dernière simplification : les tautologies

- Une tautologie est une clause contenant à la fois A et $\neg A$;
- Si T est le sous-ensemble des tautologies de S, alors on peut remplacer S par $S \backslash T$;
- En effet, S est satisfiable ssi $S \backslash T$ l'est.

Procédure de DPLL

Algorithme

DPLL(S)=
  si $S = \emptyset$ alors retourner « satisfiable » ;
  sinon si $\Box \in S$ alors retourner « insatisfiable » ;
  sinon si S contient une tautologie C alors retourner DPLL($S \backslash C$) ;
  sinon si S contient une clause unitaire avec A (resp. $\neg A$) alors
    retourner DPLL($S[A := \top]$) (resp. DPLL($S[A := \bot]$)) ;
  sinon si A (resp. $\neg A$) est pur dans S alors
    retourner DPLL($S[A := \top]$) (resp. DPLL($S[A := \bot]$)) ;
  sinon choisir une variable A de S
    et retourner DPLL($S[A := \top]$) ou DPLL($S[A := \bot]$).

Résolution

Principe de la méthode

- Méthode clausale par réfutation (comme DPLL) :
  - On nie la proposition initiale ;
  - On la met ensuite en forme clausale.
- Règle de résolution entre deux clauses :

$$\frac{C \vee A \qquad \neg A \vee C'}{C \vee C'}$$

- Les clauses au-dessus de la barre sont les prémisses ;
- La clause en dessous est le résolvant entre les clauses prémisses.

Procédure de résolution

Algorithme

$Sat := \emptyset$;
tant que $S \neq \emptyset$ faire
    choisir $C \in S$;
    $S := S \backslash \{C\}$;
    si $C = \square$ alors retourner « insatisfiable » ;
    si C est une tautologie alors ; (* passer à la clause suivante *)
    sinon, si $C \in Sat$ alors ; (* idem *)
    sinon pour tout résolvant C_1 entre C
    et une clause de $Sat \cup \{C\}$ faire
        $S := S \cup \{C_1\}$;
    $Sat := Sat \cup \{C\}$;
retourner « satisfiable ».

Méthode des tableaux

Règles de clôture et règles analytiques

$$\frac{\bot}{\odot}\ \odot_{\bot} \qquad \frac{\neg\top}{\odot}\ \odot_{\neg\top} \qquad \frac{P \qquad \neg P}{\odot}\ \odot$$

$$\frac{\neg\neg P}{P}\ \alpha_{\neg\neg} \qquad \frac{P \Leftrightarrow Q}{\neg P, \neg Q \mid P, Q}\ \beta_{\Leftrightarrow} \qquad \frac{\neg(P \Leftrightarrow Q)}{\neg P, Q \mid P, \neg Q}\ \beta_{\neg\Leftrightarrow}$$

$$\frac{P \wedge Q}{P, Q}\ \alpha_{\wedge} \qquad \frac{\neg(P \vee Q)}{\neg P, \neg Q}\ \alpha_{\neg\vee} \qquad \frac{\neg(P \Rightarrow Q)}{P, \neg Q}\ \alpha_{\neg\Rightarrow}$$

$$\frac{P \vee Q}{P \mid Q}\ \beta_{\vee} \qquad \frac{\neg(P \wedge Q)}{\neg P \mid \neg Q}\ \beta_{\neg\wedge} \qquad \frac{P \Rightarrow Q}{\neg P \mid Q}\ \beta_{\Rightarrow}$$

Calcul des séquents intuitionniste (LJ)

Règles

$$\frac{}{\Gamma, A \vdash A}\,\text{ax} \qquad \frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\,\text{cont}$$

$$\frac{\Gamma \vdash A \qquad \Gamma, B \vdash C}{\Gamma, A \Rightarrow B \vdash C}\Rightarrow_{\text{left}} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\Rightarrow_{\text{right}}$$

$$\frac{\Gamma \vdash A \qquad \Gamma, B \vdash C}{\Gamma, A \Leftrightarrow B \vdash C}\Leftrightarrow_{\text{left1}}$$

$$\frac{\Gamma \vdash B \qquad \Gamma, A \vdash C}{\Gamma, A \Leftrightarrow B \vdash C}\Leftrightarrow_{\text{left2}} \qquad \frac{\Gamma, A \vdash B \qquad \Gamma, B \vdash A}{\Gamma \vdash A \Leftrightarrow B}\Leftrightarrow_{\text{right}}$$

Calcul des séquents intuitionniste (LJ)

Règles

$$\frac{\Gamma, A, B \vdash C}{\Gamma, A \wedge B \vdash C}\ \wedge_{\text{left}} \qquad \frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ \wedge_{\text{right}}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \vee_{\text{right1}}$$

$$\frac{\Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C}\ \vee_{\text{left}} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \vee_{\text{right2}}$$

$$\frac{\Gamma \vdash A}{\Gamma, \neg A \vdash B}\ \neg_{\text{left}} \qquad \frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\ \neg_{\text{right}}$$

$$\frac{}{\Gamma, \bot \vdash A}\ \bot_{\text{left}} \qquad \frac{}{\Gamma \vdash \top}\ \top_{\text{right}}$$

Calcul des séquents intuitionniste (LJ)

Règles

$$\frac{\Gamma, A(t) \vdash B}{\Gamma, \forall x.A(x) \vdash B}\ \forall_{\text{left}} \qquad \frac{\Gamma \vdash A(x)}{\Gamma \vdash \forall x.A(x)}\ \forall_{\text{right}},\ x \notin \Gamma$$

$$\frac{\Gamma, A(x) \vdash B}{\Gamma, \exists x.A(x) \vdash B}\ \exists_{\text{left}},\ x \notin \Gamma, B \qquad \frac{\Gamma \vdash A(t)}{\Gamma \vdash \exists x.A(x)}\ \exists_{\text{right}}$$

$$\frac{\Gamma \vdash A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}\ \text{cut}$$

Calcul des séquents classique (LJ_{em})

Règles

$$\frac{\Gamma, A(t) \vdash B}{\Gamma, \forall x.A(x) \vdash B}\ \forall_{\text{left}} \qquad \frac{\Gamma \vdash A(x)}{\Gamma \vdash \forall x.A(x)}\ \forall_{\text{right}},\ \ x \notin \Gamma$$

$$\frac{\Gamma, A(x) \vdash B}{\Gamma, \exists x.A(x) \vdash B}\ \exists_{\text{left}},\ \ x \notin \Gamma, B \qquad \frac{\Gamma \vdash A(t)}{\Gamma \vdash \exists x.A(x)}\ \exists_{\text{right}}$$

$$\frac{\Gamma \vdash A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}\ \text{cut} \qquad \frac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}\ \text{em}$$

Calcul des séquents classique (LK)

Règles

$$\frac{}{\Gamma, A \vdash \Delta, A}\;\mathrm{ax} \qquad \frac{\Gamma \vdash \Delta, A \qquad \Gamma, A \vdash \Delta, B}{\Gamma \vdash \Delta, B}\;\mathrm{cut}$$

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\;\mathrm{cont_{left}} \qquad \frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A}\;\mathrm{cont_{right}}$$

Calcul des séquents classique (LK)

Règles

$$\frac{\Gamma \vdash \Delta, A \qquad \Gamma, B \vdash \Delta}{\Gamma, A \Rightarrow B \vdash \Delta}\Rightarrow_{\text{left}} \qquad \frac{\Gamma, A \vdash \Delta, B}{\Gamma \vdash \Delta, A \Rightarrow B}\Rightarrow_{\text{right}}$$

$$\frac{\Gamma \vdash \Delta, A, B \qquad \Gamma, A, B \vdash \Delta}{\Gamma, A \Leftrightarrow B \vdash \Delta}\Leftrightarrow_{\text{left}}$$

$$\frac{\Gamma, A \vdash \Delta, B \qquad \Gamma, B \vdash \Delta, A}{\Gamma \vdash \Delta, A \Leftrightarrow B}\Leftrightarrow_{\text{right}}$$

Calcul des séquents classique (LK)

Règles

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge_{\text{left}} \qquad \frac{\Gamma \vdash \Delta, A \qquad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B}\ \wedge_{\text{right}}$$

$$\frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\ \vee_{\text{left}} \qquad \frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B}\ \vee_{\text{right}}$$

$$\frac{\Gamma \vdash \Delta, A}{\Gamma, \neg A \vdash \Delta}\ \neg_{\text{left}} \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \Delta, \neg A}\ \neg_{\text{right}}$$

$$\frac{}{\Gamma, \bot \vdash \Delta}\ \bot_{\text{left}} \qquad \frac{}{\Gamma \vdash \Delta, \top}\ \top_{\text{right}}$$

Calcul des séquents classique (LK)

Règles

$$\frac{\Gamma, A(t) \vdash \Delta}{\Gamma, \forall x.A(x) \vdash \Delta}\,\forall_{\text{left}} \qquad \frac{\Gamma \vdash \Delta, A(x)}{\Gamma \vdash \Delta, \forall x.A(x)}\,\forall_{\text{right}},\ \ x \notin \Gamma, \Delta$$

$$\frac{\Gamma, A(x) \vdash \Delta}{\Gamma, \exists x.A(x) \vdash \Delta}\,\exists_{\text{left}},\ \ x \notin \Gamma, \Delta \qquad \frac{\Gamma \vdash \Delta, A(t)}{\Gamma \vdash \Delta, \exists x.A(x)}\,\exists_{\text{right}}$$

Guide de survie du petit Coq-uin

Correspondance LK/Coq

Logique propositionnelle		Logique du premier ordre	
Règle LK	Tactique Coq	Règle LK	Tactique Coq
ax	`assumption`	$\forall_{\text{right}}$	`intro`
cut	`cut`	$\forall_{\text{left}}$	`apply`
$\Rightarrow_{\text{right}}$	`intro`	$\exists_{\text{right}}$	`exists`
$\Rightarrow_{\text{left}}$	`apply`	$\exists_{\text{left}}$	`elim`
$\Leftrightarrow_{\text{right}}$	`split`		
$\Leftrightarrow_{\text{lefti}}$	`elim`		
$\wedge_{\text{right}}$	`split`		
$\wedge_{\text{left}}$	`elim`		
$\vee_{\text{right1}}$	`left`		
$\vee_{\text{right2}}$	`right`		
$\vee_{\text{left}}$	`elim`		
$\neg_{\text{right}}$	`intro`		
$\neg_{\text{left}}$	`elimtype False` + `apply`		
$\top_{\text{right}}$, $\bot_{\text{left}}$	`auto`		

Unification

Algorithme d'unification de Robinson

- $G\{s_1 = t_1, \ldots, s_n = t_n\} \hookrightarrow \{x_1 = u_1, \ldots, x_m = u_m\}$
  où x_i sont des variables distinctes et $x_i \notin u_j$;
- Règles :
  - $G \cup \{t = t\} \hookrightarrow G$ (delete) ;
  - $G \cup \{f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)\} \hookrightarrow G \cup \{s_1 = t_1, \ldots, s_n = t_n\}$ (decompose) ;
  - $G \cup \{f(s_1, \ldots, s_n) = g(t_1, \ldots, t_m)\} \hookrightarrow \perp$, si $f \neq g$ ou $n \neq m$ (conflict) ;
  - $G \cup \{f(s_1, \ldots, s_n) = x\} \hookrightarrow G \cup \{x = f(s_1, \ldots, s_n)\}$ (swap) ;
  - $G \cup \{x = t\} \hookrightarrow G[t/x] \cup \{x = t\}$, si $x \notin t$ et $x \in G$ (eliminate) ;
  - $G \cup \{x = f(s_1, \ldots, s_n)\} \hookrightarrow \perp$, si $x \in f(s_1, \ldots, s_n)$ (check).

Méthode des tableaux

Règles de clôture et règles analytiques

$$\frac{\bot}{\odot}\ \odot_{\bot} \qquad \frac{\neg\top}{\odot}\ \odot_{\neg\top} \qquad \frac{P \qquad \neg P}{\odot}\ \odot$$

$$\frac{\neg\neg P}{P}\ \alpha_{\neg\neg} \qquad \frac{P \Leftrightarrow Q}{\neg P, \neg Q \mid P, Q}\ \beta_{\Leftrightarrow} \qquad \frac{\neg(P \Leftrightarrow Q)}{\neg P, Q \mid P, \neg Q}\ \beta_{\neg\Leftrightarrow}$$

$$\frac{P \wedge Q}{P, Q}\ \alpha_{\wedge} \qquad \frac{\neg(P \vee Q)}{\neg P, \neg Q}\ \alpha_{\neg\vee} \qquad \frac{\neg(P \Rightarrow Q)}{P, \neg Q}\ \alpha_{\neg\Rightarrow}$$

$$\frac{P \vee Q}{P \mid Q}\ \beta_{\vee} \qquad \frac{\neg(P \wedge Q)}{\neg P \mid \neg Q}\ \beta_{\neg\wedge} \qquad \frac{P \Rightarrow Q}{\neg P \mid Q}\ \beta_{\Rightarrow}$$

Méthode des tableaux (sans variable libre)

δ/γ-règles

$$\frac{\exists x.P(x)}{P(c)}\,\delta_{\exists},\ c \text{ frais} \qquad \frac{\neg\forall x.P(x)}{\neg P(c)}\,\delta_{\neg\forall},\ c \text{ frais}$$

$$\frac{\forall x.P(x)}{P(t)}\,\gamma_{\forall\text{inst}} \qquad \frac{\neg\exists x.P(x)}{\neg P(t)}\,\gamma_{\neg\exists\text{inst}}$$

Méthode des tableaux (avec variable libre, destructif)

δ/γ-règles

$$\frac{\exists x.P(x)}{P(f(X_1,\ldots,X_n))}\ \delta_\exists,\ \begin{array}{l}f \text{ frais,}\\ X_i \text{ var. lib.}\end{array} \qquad \frac{\neg\forall x.P(x)}{\neg P(f(X_1,\ldots,X_n))}\ \delta_{\neg\forall},\ \begin{array}{l}f \text{ frais,}\\ X_i \text{ var. lib.}\end{array}$$

$$\frac{\forall x.P(x)}{P(X)}\ \gamma_{\forall M} \qquad \frac{\neg\exists x.P(x)}{\neg P(X)}\ \gamma_{\neg\exists M}$$

$$\frac{\forall x.P(x)}{P(t)}\ \gamma_{\forall\text{inst}} \qquad \frac{\neg\exists x.P(x)}{\neg P(t)}\ \gamma_{\neg\exists\text{inst}}$$

$$\frac{\begin{array}{l}\text{Appliquer } \sigma \text{ à l'arbre s'il existe dans la branche}\\ \text{deux littéraux } K \text{ et } \neg L \text{ t.q. } \sigma = mgu(K,L)\end{array}}{\odot}\ \odot$$

Méthode des tableaux (avec variable libre, non destructif)

δ/γ-règles

$$\frac{\exists x.P(x)}{P(f(X_1,\ldots,X_n))}\ \delta_\exists,\ \begin{array}{l} f \text{ frais,} \\ X_i \text{ var. lib.}\end{array} \qquad \frac{\neg\forall x.P(x)}{\neg P(f(X_1,\ldots,X_n))}\ \delta_{\neg\forall},\ \begin{array}{l} f \text{ frais,} \\ X_i \text{ var. lib.}\end{array}$$

$$\frac{\forall x.P(x)}{P(X)}\ \gamma_{\forall M} \qquad \frac{\neg\exists x.P(x)}{\neg P(X)}\ \gamma_{\neg\exists M}$$

$$\frac{\forall x.P(x)}{P(t)}\ \gamma_{\forall\text{inst}} \qquad \frac{\neg\exists x.P(x)}{\neg P(t)}\ \gamma_{\neg\exists\text{inst}}$$

Méthode des tableaux (non destructif, avec ϵ-termes)

δ/γ-règles

$$\frac{\exists x.P(x)}{P(\epsilon(x).P(x))}\ \delta_{\exists} \qquad \frac{\neg\forall x.P(x)}{\neg P(\epsilon(x).\neg P(x))}\ \delta_{\neg\forall}$$

$$\frac{\forall x.P(x)}{P(X)}\ \gamma_{\forall M} \qquad \frac{\neg\exists x.P(x)}{\neg P(X)}\ \gamma_{\neg\exists M}$$

$$\frac{\forall x.P(x)}{P(t)}\ \gamma_{\forall \mathrm{inst}} \qquad \frac{\neg\exists x.P(x)}{\neg P(t)}\ \gamma_{\neg\exists \mathrm{inst}}$$

Méthode des tableaux (non destructif, avec ϵ-termes)

Exemple

$$\dfrac{\forall x\ (P(x) \vee Q(x))\ ,\ \ \neg P(a)\ ,\ \ \neg Q(a)}{\dfrac{P(X) \vee Q(X)}{\dfrac{P(X)}{\dfrac{P(a) \vee Q(a)}{\dfrac{P(a)}{\odot}\ \odot \qquad \dfrac{Q(a)}{\odot}\ \odot}\ \beta_\vee}\ \gamma_{\forall\mathrm{inst}} \qquad Q(X)}\ \beta_\vee}\ \gamma_{\forall M}$$

Skolémisation

Fonctions de skolémisation et herbrandisation

- Si Φ est atomique, $s(\Phi) = h(\Phi) = \Phi$;
- $s(\Phi \wedge \Phi') = s(\Phi) \wedge s(\Phi')$, $h(\Phi \wedge \Phi') = h(\Phi) \wedge h(\Phi')$;
- $s(\Phi \vee \Phi') = s(\Phi) \vee s(\Phi')$, $h(\Phi \vee \Phi') = h(\Phi) \vee h(\Phi')$;
- $s(\neg\Phi) = \neg h(\Phi)$, $h(\neg\Phi) = \neg s(\Phi)$;
- $s(\Phi \Rightarrow \Phi') = h(\Phi) \Rightarrow s(\Phi')$, $h(\Phi \Rightarrow \Phi') = s(\Phi) \Rightarrow h(\Phi')$;
- $s(\forall x.\Phi) = s(\Phi)$, $h(\forall x.\Phi) = h(\Phi)[f(x_1, \ldots, x_n)/x]$, où $x_1, \ldots, x_n$ sont les variables libres de $\forall x.\Phi$;
- $s(\exists x.\Phi) = s(\Phi)[f(x_1, \ldots, x_n)/x]$, où $x_1, \ldots, x_n$ sont les variables libres de $\exists x.\Phi$, $h(\exists x.\Phi) = h(\Phi)$.
- Ensuite, une fois le calcul terminé :
  - Skolémisation : $\forall x_1. \ldots .\forall x_n.s(\Phi)$, où $x_1, \ldots, x_n$ sont les variables libres de $s(\Phi)$;
  - Herbrandisation : $\exists x_1. \ldots .\exists x_n.h(\Phi)$, où $x_1, \ldots, x_n$ sont les variables libres de $h(\Phi)$.

Résolution et factorisations binaires

Résolution binaire

$$\frac{A \vee C \qquad \neg B \vee D}{\sigma(C) \vee \sigma(D)}\ \text{res}$$

où $\sigma(A) = \sigma(B)$.

Factorisations binaires

$$\frac{A \vee B \vee C}{\sigma(B) \vee \sigma(C)}\ \text{fact}^{+} \qquad \frac{\neg A \vee \neg B \vee C}{\neg\sigma(B) \vee \sigma(C)}\ \text{fact}^{-}$$

où $\sigma(A) = \sigma(B)$.

Procédure de résolution (la même qu'en propositionnel !)

Algorithme (« Given-Clause Algorithm »)

$Sat := \emptyset$;
tant que $S \neq \emptyset$ faire
    choisir $C \in S$;
    $S := S \backslash \{C\}$;
    si $C = \square$ alors retourner « insatisfiable » ;
    si C est une tautologie alors ; (* passer à la clause suivante *)
    sinon, si $C \in Sat$ alors ; (* idem *)
    sinon pour tout résolvant C_1 entre C
    et une clause de $Sat \cup \{C\}$ faire
        $S := S \cup \{C_1\}$;
    $Sat := Sat \cup \{C\}$;
retourner « satisfiable ».

Pourquoi des règles de factorisation ?

Exemple

- Ensemble de clauses : $S = \{P(x) \vee P(y), \neg P(a) \vee \neg P(z)\}$;
- S est bien insatisfiable ;
- Résolution entre $P(x) \vee P(y)$ et $\neg P(a) \vee \neg P(z)$: $P(y) \vee \neg P(z)$;
- Résolution entre $P(x) \vee P(y)$ et $P(y') \vee \neg P(z)$: $P(y) \vee P(y')$;
- Résolution entre $\neg P(a) \vee \neg P(z)$ et $P(y) \vee \neg P(z')$: $\neg P(z) \vee \neg P(z')$;
- On regénère les mêmes clauses indéfiniment ;
- Règle fact$^+$ sur $P(x) \vee P(y)$: $P(x)$;
- Règle fact$^-$ sur $\neg P(a) \vee \neg P(z)$: $\neg P(a)$;
- Résolution entre $P(x)$ et $\neg P(a)$: $\Box$.