Cours 7
Résolution propositionnelle

Un système formel définit des règles de calcul entre formules qui simulent le raisonnement, ce qui permet de déduire de nouvelles formules. Il est défini par :

— le langage et les formules sur ce langage,
— les axiomes : formules ou schémas de formules supposées vraies,
— les règles d'inférence : règles permettant de déduire une nouvelles formule (conclusion) à partir d'un ensemble de formules (prémisses ou hypothèses).

On note $\vdash \phi$ si ϕ peut se déduire dans le système formel considéré. Deux questions fondamentales sont systématiquement posées pour relier le calcul formel(le système de preuve syntaxique) et la sémantique :

1. Le système est-il correct : une formule déduite est-elle une tautologie ?

2. Le système est-il complet : toute tautologie peut-elle se déduire ?

7.1 Méthode de résolution (Robinson)

La méthode de résolution est un système formel qui utilise des formules sous formes de clauses, n'a pas d'axiomes, et comporte une règle d'inférence (deux si on ajoute la règle de simplification). La méthode est réfutationnelle : pour montrer qu'on peut prouver ϕ on suppose $\neg\phi$ et on montre que cela conduit à une absurdité. La méthode a été introduite par Robinson dans le cadre plus général du calcul de prédicat.
Nous notons $s(A)$ l'ensemble des littéraux de la clause A. Par convention $\perp$ est la clause vide et $s(\perp) = \emptyset$.

Notons L^c le littéral complémentaire d'un littéral L. Par exemple :$x^c = \bar{x}$ et $\bar{x}^c = x$.
Nous présentons le calcul de résolvant entre 2 clauses, qui constitue l'unique règle du système de résolution.

Définition 1 (Résolvant)**.** *Soient A et B deux clauses. La clause C est un résolvant de A et B ssi il y a un littéral L tel que $L \in s(A), L^c \in s(B)$ et $s(C) = (s(A) - L) \cup (s(B) - L^c)$ On note*

$$\frac{A \quad B}{C}.$$

On dit que C est engendrée par A et B, et que A et B sont les parents de C.

Exemple

1. $p + r$ et $q + \bar{r}$ donne la résolvante $p + q$.

2. $p + q + r$ et $\bar{r} + s + \bar{t} + q$ donnent $p + q + s + \bar{t}$.

3. $p + q + r$ et $\bar{p} + \bar{q} + s$ donnent $q + r + \bar{q} + s$ et $p + r + \bar{p} + s$. Attention, elles ne donnent pas $r + s$.

4. $\bar{p} + q$ et p donnent q.

5. $\bar{p}$ et p donnent la clause vide notée $\perp$.

Propriété
Si l'un des parents d'un résolvant est valide, le résolvant est valide ou contient l'autre parent.

Définition 2 (Preuve)**.** *Soient Γ un ensemble de clauses et C une clause. Une preuve de C à partir de Γ est une liste de clauses qui se termine par C. Toute clause de la preuve est égale à un élément de Γ ou est un résolvant de deux clauses la précédant dans la preuve.*

La clause C est déduite de Γ, notée $\Gamma \vdash C$, s'il y a une preuve de C à partir de Γ.

Une preuve P de C à partir d'un ensemble de clauses Γ est de taille n si elle contient n lignes.

Proposition 1. *(Monotonie et composition)*
Soient Γ, Δ deux ensembles de clauses et A, B deux clauses.

1. Monotonie de la déduction *: Si $\Gamma \vdash A$ et si Γ est inclus dans Δ alors $\Delta \vdash A$.*

2. Composition des déductions *: Si $\Gamma \vdash A$ et $\Gamma \vdash B$ et si C est un résolvant et A et B alors $\Gamma \vdash C$.*

Théorème 1. *(Correction de la règle de résolution) Si C est un résolvant de A et B alors $A, B \vDash C$.*

Théorème 2. *(Correction de la déduction) Soient Γ un ensemble de clauses et C une clause.*

Si $\Gamma \vdash C$ alors $\Gamma \vDash C$.

7.2 Complétude pour la réfutation

La méthode de résolution n'est pas complète pour la déduction. Par contre, elle est complète par réfutation. C'est à dire si Γ est contradictoire alors il y a une preuve de la clause vide à partir de Γ.

Si $\Gamma \vDash \bot$ alors $\Gamma \vdash \bot$.

7.3 Réduction d'un ensemble de clauses

Définition 3. *Réduire un ensemble de clauses, c'est lui enlever les clauses valides et les clauses contenant une autre clause de l'ensemble. Un ensemble de clauses est réduit s'il n'est plus réductible.*

Proposition 2. *Un ensemble de clauses est équivalent à l'ensemble de clauses obtenu par réduction.*

Preuve. Enlever une clause valide donc valant 1, c'est appliquer l'identité $x.1 = x$.
Enlever une clause incluant une autre clause, c'est appliquer l'identité $x(x + y) = x$.

Exemple 1. *L'ensemble de clauses $\{p + q + \bar{p}, p + r, p + r + \bar{s}, r + q\}$ est équivalent à l'ensemble réduit $\{p + r, r + q\}$.*

7.4 Stratégies de résolutions

7.4.1 Résolution positive

On n'autorise la Résolution qu'entre deux clauses dont l'une est positive (c-à-d n'a aucun littéral négatif).

Remarque 1. *Il y a au moins une clause positive dans un ensemble contradictoire. (Sinon l'ensemble serait satisfait par l'assignation qui rend toutes les variables fausses.)*

Propriété : La Résolution positive est complète pour la réfutation.
Exemple pour la résolution positive. De l'ensemble de clauses

(1) $\bar{p} + \bar{q} + r$		**(5)** $\bar{q} + r$	**(1)** et **(3)**
(2) $\bar{p} + q$		**(6)** q	**(2)** et **(3)**
(3) p	on obtient successivement	**(7)** r	**(5)** et **(6)**
(4) $\bar{r}$		**(8)** $\bot$	**(4)** et **(7)**

7.4.2 Résolution négative

On n'autorise la Résolution qu'entre deux clauses dont l'une est négative(c-à-d n'a aucun littéral positif).

Remarque 2. *Il y a au moins une clause négative dans un ensemble contradictoire. (Sinon l'ensemble serait satisfait par l'assignation qui rend vraie toutes les variables.*

Propriété : La Résolution négative est complète pour la réfutation.
Exemple pour la résolution négative. De l'ensemble de clauses

(1) $\bar{p} + \bar{q} + r$		**(5)** $\bar{p} + \bar{q}$	**(1)** et **(4)**
(2) $\bar{p} + q$		**(6)** $\bar{p}$	**(2)** et **(5)**
(3) p	on obtient successivement	**(8)** $\bot$	**(3)** et **(6)**
(4) $\bar{r}$			

7.4.3 Résolution linéaire et Résolution linéaire par entrée

Une preuve par résolution linéaire d'une clause F à partir d'un ensemble $\mathcal{C}$ de clauses est une suite $C_0, C_1, \ldots, C_n$ telle que $C_0 \in \mathcal{C}$ et pour tout i, C_i est une résolvante de C_{i-1} et d'une clause de C ou d'un C_j précédent.

La preuve est linéaire par entrée si on prend toujours une clause de $\mathcal{C}$ (clause d'entrée).
Exemple pour la résolution linéaire par entrée. De l'ensemble de clauses

(1) $\bar{p}+\bar{q}+r$			**(5)** q	**(2)** et **(3)**
(2) $\bar{p}+q$			**(6)** $\bar{p}+r$	**(5)** et **(1)**
(3) p		on obtient successivement	**(7)** r	**(6)** et **(3)**
(4) $\bar{r}$			**(8)** $\bot$	**(7)** et **(4)**

Propriété : La Résolution linéaire est complète pour la réfutation. La résolution linéaire par entrée n'est pas complète.

Contre-exemple :

	(1) $p+q$		**(5)** p	**(1)** et **(2)**
	(2) $p+\bar{q}$	on obtient par	**(6)** q	**(5)** et **(3)**
De l'ensemble de clauses	**(3)** $\bar{p}+q$	résolution linéaire :	**(7)** $\bar{p}$	**(6)** et **(4)**
	(4) $\bar{p}+\bar{q}$		**(8)** $\bot$	**(7)** et **(5)**

On ne peut pas avoir de réfutation linéaire par entrée car les clauses d'entrée ont toutes deux littéraux et ne peuvent donner directement la clause vide.

Une **Clause de Horn** est une clause qui a au plus un littéral positif. Les clauses négatives sont des clauses de Horn.

Propriété : La résolution linéaire par entrée est complète pour la réfutation des clauses de Horn.

Les clauses du premier exemples étaient des clauses de Horn, on était donc assuré de l'existence d'une réfutation linéaire par entrée. Celles du deuxième exemple n'étaient pas toutes des clauses de Horn. Le langage de programmation Prolog est basé sur les clauses de Horn (étendues en calcul des prédicats).

7.4.4 Stratégie de l'ensemble support

Soit T un sous ensemble de $\mathcal{C}$ tel que $C\backslash T$ soit satisfaisable et qui sera appelé *l'ensemble support*. Dans cette stratégie de l'ensemble support, on interdit la Résolution entre deux clauses de $C\backslash T$, c'est-à-dire on n'autorise la Résolution qu'entre deux clauses dont l'une au moins est une clause de T ou une descendante d'une clause de T.

Exemple de stratégie de l'ensemble support, avec $T = \{p, \neg r\}$

	(1) $\bar{p}+\bar{q}+r$		**(5)** q	**(3)** et **(2)**
	(2) $\bar{p}+q$	on obtient par	**(6)** $\bar{p}+\bar{q}$	**(4)** et **(1)**
De l'ensemble de clauses	**(3)** p	résolution à support :	**(7)** $\bar{p}$	**(5)** et **(6)**
	(4) $\bar{r}$		**(8)** $\bot$	**(3)** et **(7)**

Propriété. La stratégie de l'ensemble support est complète pour la réfutation.

Intérêt : Elle permet de se focaliser sur un sous-ensemble de clauses qui joue un rôle particulier.

Application Pour montrer qu'une formule F est valide, on cherchera une réfutation de $\mathcal{C}(\neg F)$. En effet : F valide ssi $\neg F$ insatisfaisable ssi $\mathcal{C}(\neg F)$ est contradictoire ssi $\mathcal{C}(\neg F) \vdash \bot$.

Soit $F = (p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$.

On vérifie $\mathcal{C}(\neg F) = \{\bar{p}+\bar{q}+r, \bar{p}+q, p, \bar{r}\}$. C'est l'exemple étudié précédemment, où on a vu que $\mathcal{C}(\neg F) \vdash \bot$ donc F est une tautologie(valide).

Exercice 1. *En utilisant la méthode de résolution, montrer que* $\{\mathbf{a} \Rightarrow \mathbf{b}, \mathbf{b} \Rightarrow \mathbf{c}, \mathbf{a} \vee \neg\mathbf{c}\} \vDash (\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}) \vee (\neg\mathbf{a} \wedge \neg\mathbf{b} \wedge \neg\mathbf{c})$.

On a $\{\mathbf{a} \Rightarrow \mathbf{b}, \mathbf{b} \Rightarrow \mathbf{c}, \mathbf{a} \vee \neg\mathbf{c}\} \vDash (\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}) \vee (\neg\mathbf{a} \wedge \neg\mathbf{b} \wedge \neg\mathbf{c})$ ssi $\{\mathbf{a} \Rightarrow \mathbf{b}, \mathbf{b} \Rightarrow \mathbf{c}, \mathbf{a} \vee \neg\mathbf{c}, \neg\big((\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}) \vee (\neg\mathbf{a} \wedge \neg\mathbf{b} \wedge \neg\mathbf{c})\big)\} \vDash \bot$.

Et par les équivalences remarquables , ssi $\{\bar{\mathbf{a}} + \mathbf{b}, \bar{\mathbf{b}} + \mathbf{c}, \mathbf{a} + \bar{\mathbf{c}}, \bar{\mathbf{a}} + \bar{\mathbf{b}} + \bar{\mathbf{c}}, \mathbf{a} + \mathbf{b} + \mathbf{c}\} \vDash \bot$.

(1)	$a+b+c$	Hyp
(2)	$\bar{a}+b$	Hyp
(3)	$\bar{b}+c$	Hyp
(4)	$\bar{c}+a$	Hyp
(5)	$\bar{a}+\bar{b}+\bar{c}$	Hyp
(6)	$b+c$	Res (1,2)
(7)	$a+c$	Res (1,3)
(8)	$a+b$	Res (1,4)
(9)	a	Res (4,7)
(10)	$\bar{b}+\bar{c}$	Res (5,9)
(11)	b	Res (2,8)
(12)	c	Res (11,3)
(13)	$\bar{b}$	Res (10,12)
(14)	$\bot$	Res (11,13)

On a prouvé que : $\{\bar{a}+b, \bar{b}+c, a+\bar{c}, \bar{a}+\bar{b}+\bar{c}, a+b+c\} \vdash \bot$. Par le théorème de correction de la résolution on déduit alors que $\{\bar{a}+b, \bar{b}+c, a+\bar{c}, \bar{a}+\bar{b}+\bar{c}, a+b+c\} \vDash \bot$.

7.5 Algorithme de Davis-Putnam

L'algorithme de Davis et Putnam sert à savoir si un ensemble de clauses est satisfaisable.
Cet algorithme est bien meilleur que ceux que nous avons déjà étudiés (table de vérité, transformation en somme de monômes ou produit de sommes,arbre sémantique, algorithme de Quine, ...) pour des raisons que nous présentons de suite.

Il y a deux types de transformations de formules :

1. celles qui préservent le sens, c'est-à-dire transforment une formule en une autre formule équivalente
2. celles qui préservent seulement la satisfaisabilité, c'est-à-dire transforment une formule satisfaisable en une autre formule satisfaisable

L'efficacité de l'algorithme de Davis et Putnam vient de l'utilisation des transformations **préservant seulement la satisfaisabilité**, car ces transformations sont moins coûteuses que celles préservant le sens.

7.6 Suppression des clauses qui ont des littéraux isolés

Définition 4 (Littéral isolé)**.** *Soit* Γ *un ensemble de clauses. L un littéral élément de* Γ*. est isolé (relativement à* Γ*), si aucune clause de* Γ *ne comporte de littéral complémentaire de L.*

Lemme 1.
La suppression des clauses qui comportent un littéral isolé, préserve la satisfaisabilité.

Exemple 2. *(1)* $p+q+r$ *(2)* $\bar{q}+\bar{r}$ *(3)* $q+s$ *(4)* $\bar{x}+t$

Les littéraux p et t sont isolés. Donc on peut supprimer les clauses (1) et (4). Après cette suppression, on obtient l'ensemble Θ : *(2)* $\bar{q}+\bar{r}$ *(3)* $q+s$

Les littéraux $\bar{r}$ *et s sont à leur tour isolés pour l'ensemble* Θ*. En enlevant les clauses qui contiennent ces littéraux, on obtient l'ensemble vide, dont toute assignation est modèle. Le lemme précédent nous permet de dire que* Γ *a un modèle.*

7.7 Résolution unitaire

Lemme 2. *Soit* Γ *un ensemble de clauses et L l'ensemble des littéraux des clauses unitaires de* Γ*.*
Soit Θ *l'ensemble de clauses ainsi obtenu à partir de* Γ

- *si L comporte deux littéraux complémentaires, alors* $\Theta = \{\bot\}$*.*
- *sinon* Θ *est obtenue ainsi*
  - *on enlève les clauses qui comportent un élément de L*
  - *on enlève des clauses restantes les complémentaires des éléments de L*

Γ ***a un modèle si et seulement si*** Θ ***en a un.***

Exemple 3.

- *Soit* Γ *l'ensemble de clauses* $a+b+\bar{d}, \bar{a}+c+\bar{d}, \bar{b}, d, \bar{c}$*. Par une première résolution unitaire, on obtient :* $a, \bar{a}$*. Par une deuxième résolution unitaire, on obtient la clause vide, donc* Γ *n'a pas de modèle.*
- *Soit* Γ *l'ensemble de clauses :* $p, q, p+r, \bar{p}+r, q+\bar{r}, \bar{q}+s$*. Par résolution unitaire, on obtient :*r, s*. Cet ensemble de clauses a un modèle donc* Γ *en a un.*

Algorithme de Davis et Putnam

bool fonction DP(Γ ensemble de clauses non valides) La fonction retourne vrai si et seulement si Γ est satisfaisable

1. Si $\bot \in \Gamma$, retourner(faux). Si $\Gamma = \emptyset$, retourner (vrai).
2. Réduire Γ (**RE**) : il suffit d'enlever toute clause contenant une autre clause.
3. Enlever de Γ les clauses comportant des littéraux isolés (**RI**). Si l'ensemble Γ a été modifié, aller en 1.
4. Appliquer à Γ la résolution unitaire(**RU**). Si l'ensemble Γ a été modifié, aller en 1.
5. Soit x une variable quelconque de Γ retourner ($DP(G[x := 0])$ ou $DP(G[x := 1])$)

Théorème 3.
L'algorithme de Davis et Putnam est correct et se termine.

Le choix de la prochaine variable à affecter a un impact important sur la taille de l'arbre de recherche développé par DP. Citant la plus simple. **NOMS :**(pour Maximum Occurences in Clauses of Minimum Size) sélectionner la variable ayant le plus d'occurrences dans les clauses les plus courtes.