

Représentation des connaissances en logique

Approche déclarative bien éprouvée qui permet de disposer de méthodes de représentation et de déduction solides basées sur des années d'expérience des mathématiciens

- ▶ logique des propositions
 - méthode de preuve efficace
 - trop restreint car il n'y a pas de quantificateurs
- ▶ logique des prédicats
 - dans le cas général, indécidable
 - méthodes qui s'arrêteront si c'est un théorème mais risquent de ne pas s'arrêter si on l'applique à une formule non valide (semi-décidable)

Logique des propositions

Définitions:

- ▶ T et F sont des propositions
- ▶ identificateur est une proposition associé à T ou F
- ▶ si B est une proposition alors $\neg B$ est une proposition
- ▶ si B et C sont des propositions alors
 $B \wedge C, B \vee C, B \Rightarrow C, B \Leftrightarrow C$ sont des propositions

Remarque:

$B \Rightarrow C$ n'est F que si $B = T$ et $C = F$

Par exemple: "s'il pleut" \Rightarrow "pas de pique-nique"
s'il ne pleut pas alors on ne peut rien dire ...

Exemples de traduction d'énoncés en logique des propositions

pleut

pique-nique contremandé

être mouillé

rester à la maison

s'il pleut, mais je reste à la maison
alors je ne serai pas mouillé

je serai mouillé s'il pleut

s'il pleut et que le pique-nique n'est
pas contremandé ou que je ne reste
pas à la maison, je serai mouillé

que le pique-nique soit contremandé
ou non, je reste à la maison s'il pleut
soit qu'il pleuve pas ou que je reste
à la maison

p

pc

m

rm

$(p \wedge rm) \Rightarrow \neg m$

$p \Rightarrow m$

$((p \wedge \neg pc) \vee \neg rm) \Rightarrow m$

ou

$(p \wedge (\neg pc \vee \neg rm)) \Rightarrow m$

$pc \vee \neg pc \wedge p \Rightarrow rm$

$\neg p \vee rm$

Logique des prédicats

- ▶ identificateurs peuvent être associés à d'autres valeurs (entiers, caractères, ensembles, ...)
- ▶ un identificateur peut être remplacé par une expression ($x < y$)
- ▶ quantificateurs $\exists var, \forall var$
- ▶ prédicats ($p(x)$)
- ▶ fonctions

Interprétation sur un domaine D non-vide

- ▶ chaque item est affecté à un élément de D
- ▶ chaque variable est affectée à un sous-ensemble non-vide de D
- ▶ chaque prédicat p d'arité n définit une projection de $D^n \rightarrow \{T, F\}$
- ▶ chaque fonction f d'arité m définit une projection $D^m \rightarrow D$

Avec une interprétation, on peut trouver le “sens” d’une expression (une affectation d’une valeur de vérité prise de $\{T, F\}$)

- ▶ interprétation “directe” pour T, F , constantes, variables, fonctions, prédicats.
- ▶ règles “habituelles” pour $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- ▶ variable x dans une expression E

$\forall x E$ T si $E = T$ pour toutes les affectations de x

$\exists x E$ T si $E = T$ pour au moins une affectation de x

Exemples de traduction en logique des prédicats

- | | | |
|---|--|---|
| 1 | Marcus était un homme | $homme(marcus)$ |
| 2 | Marcus était un pompéien | $pompeien(marcus)$ |
| 3 | Tous les pompéiens étaient des Romains | $\forall Xpompeien(X) \Rightarrow romain(X)$ |
| 4 | César était un dirigeant | $dirigeant(cesar)$ |
| 5 | Tous les romains étaient soit loyaux à César ou l'haïssaient | $\forall Xromain(X) \Rightarrow loyala(X, cesar) \vee hair(X, cesar)$ |
| 6 | Tout le monde est loyal à quelqu'un | $\forall X\exists Yloyala(X, Y)$
ou
$\exists Y\forall Xloyala(X, Y)$ |
| 7 | Les personnes essaient seulement d'assassiner les dirigeants auxquels ils ne sont pas loyaux | $\forall X\forall Ypersonne(X) \wedge dirigeants(Y) \wedge essaiedassassiner(X, Y) \Rightarrow \neg loyala(X, Y)$ |
| 8 | Marcus a essayé d'assassiner César | $essaiedassassiner(marcus, cesar)$ |

Marcus est-il loyal à César?

Preuve en “reculant” car le facteur de branchement est plus petit

$$\neg \text{loyal}(\text{marcus}, \text{cesar})$$

(7)

$$\text{personne}(\text{marcus}) \wedge \text{dirigeant}(\text{cesar}) \wedge \text{essaiedassassiner}(\text{marcus}, \text{cesar})$$

(4)

$$\text{personne}(\text{marcus}) \wedge \text{essaiedassassiner}(\text{marcus}, \text{cesar})$$

(8)

$$\text{personne}(\text{marcus})$$

Il nous manque donc le fait qu’un “homme est une personne”

$$\forall X \text{homme}(X) \Rightarrow \text{personne}(X)$$

ce qui nous permet de continuer avec

$$\text{homme}(\text{marcus})$$

Marcus est-il loyal à César?

Preuve en “reculant” car le facteur de branchement est plus petit

$$\neg \text{loyal}(\text{marcus}, \text{cesar})$$

(7)

$$\text{personne}(\text{marcus}) \wedge \text{dirigeant}(\text{cesar}) \wedge \text{essaiedassassiner}(\text{marcus}, \text{cesar})$$

(4)

$$\text{personne}(\text{marcus}) \wedge \text{essaiedassassiner}(\text{marcus}, \text{cesar})$$

(8)

$$\text{personne}(\text{marcus})$$

Il nous manque donc le fait qu’un “homme est une personne”

$$\forall X \text{homme}(X) \Rightarrow \text{personne}(X)$$

ce qui nous permet de continuer avec

$$\text{homme}(\text{marcus})$$

- ▶ phrases en langue naturelle peuvent être ambiguës
- ▶ plusieurs choix de représentation
- ▶ doit ajouter des “évidences”

Extension de la logique avec des fonctions évaluable

$gt(1, 0)$ $lt(0, 1)$

$gt(2, 1)$ $lt(1, 2)$

...

...

1	Marcus était un homme	$homme(marcus)$
2	Marcus était un pompéien	$pompeien(marcus)$
3	Marcus est né en 40 après J.C.	$nait(marcus, 40)$
4	Tous les hommes sont mortels	$\forall X homme(X) \Rightarrow mortel(X)$
5,6	Tous les pompéiens moururent lors de l'éruption du volcan en 79	$eruption(volcan, 79)$ $\forall X pompeien(X) \Rightarrow meurt(X, 79)$
7	Aucun mortel ne vit plus de 150 ans	$\forall X \forall T1 \forall T2 mortel(X) \wedge$ $nait(X, T1) \wedge gt(T2 - T1, 150)$ $\Rightarrow mort(X, T2)$
8	Nous sommes maintenant en 2009	$maintenant = 2009$

Est-ce que Marcus est vivant?

Il y a deux approches:

1. il aurait plus de 150 ans
2. il est mort à cause de l'éruption du volcan

Il faut maintenant ajouter deux autres règles:

9 un vivant n'est pas mort

10 un mort reste mort !!!

$$\forall X \forall T \text{vivant}(X, T)$$

$$\Leftrightarrow \neg \text{mort}(X, T)$$

$$\forall X \forall T1 \forall T2$$

$$\text{meurt}(X, T1) \wedge \text{gt}(T2, T1)$$

$$\Rightarrow \text{mort}(X, T2)$$

Marcus est-il vivant? (n'a pas survécu à l'éruption)

\neg *vivant*(*marcus*, *maintenant*)

(9)

mort(*marcus*, *maintenant*)

(10)

meurt(*marcus*, *T1*) \wedge *gt*(*maintenant*, *T1*)

(5)

pompeien(*marcus*) \wedge *gt*(*maintenant*, 79)

(2)

gt(*maintenant*, 79)

(8)

gt(2009, 79)

(calcul de *gt*)

OK

Marcus est-il vivant? (trop vieux maintenant)

$\neg \text{vivant}(\text{marcus}, \text{maintenant})$

(9)

$\text{mort}(\text{marcus}, \text{maintenant})$

(7)

$\text{mortel}(\text{marcus}) \wedge \text{nait}(\text{marcus}, T1) \wedge \text{gt}(\text{maintenant} - T1, 150)$

(4)

$\text{homme}(\text{marcus}) \wedge \text{nait}(\text{marcus}, T1) \wedge \text{gt}(\text{maintenant} - T1, 150)$

(1)

$\text{nait}(\text{marcus}, T1) \wedge \text{gt}(\text{maintenant} - T1, 150)$

(3)

$\text{gt}(\text{maintenant} - 40, 150)$

(8)

$\text{gt}(2009 - 40, 79)$

(calcul de $-$)

$\text{gt}(1969, 79)$

(calcul de gt)

OK

Procédure d'inférence

- ▶ processus de dérivation de conclusions à partir de prémisses
- ▶ en une ou plusieurs étapes

Exemples de règles d'inférences:

- ▶ “modus ponens”: $p \Rightarrow q$

$$\frac{p}{q}$$

- ▶ “modus tollens”: $p \Rightarrow q$

$$\frac{\neg q}{\neg p}$$

- ▶ élimination des “et” (\wedge): $p \wedge q$

$$\frac{p}{q}$$

- ▶ introduction des “et” (\wedge): p

$$\frac{q}{p \wedge q}$$

- ▶ instanciation universelle: $\forall X p(X)$

$$\frac{\forall X p(X)}{p(a), \text{ si } a \in \text{domaine de } X}$$

Ces règles sont pûrement “mécaniques” : elles permettent de passer d’une conclusion à une autre par application de règles générales (dérivation). Toutefois, il y a souvent des choix à faire entre plusieurs dérivations et on cherche alors une méthode systématique qui soit:

- ▶ valide
- ▶ complète
- ▶ efficace

Résolution

- ▶ applique toujours la même opération
- ▶ travaille sur une forme standardisée

On prouve qu'une chose est vraie en montrant que sa négation amène une contradiction avec les énoncés connus. Ce n'est pas très intuitif, mais c'est systématique.

Etapes de preuve:

- ▶ transforme en forme clausale
- ▶ ajoute la négation de ce qu'on veut prouver à l'ensemble des axiomes
- ▶ "résout" ces clauses en produisant de nouvelles clauses
- ▶ produit une contradiction en générant une clause vide
- ▶ les substitutions qui ont produit la contradiction donnent "la réponse"

Transformation en forme clausale (1)

Une **clause** est une disjonction de **littéraux** positifs ou négatifs.

$$r(Z, a, f(X)) \vee \neg t(U) \vee \neg v(b, k(c))$$

On peut transformer toute formule logique des prédicats en une clause.

$$\forall X \{ p(X) \Rightarrow \{ \forall Y \{ p(Y) \Rightarrow p(f(X, Y)) \} \wedge \neg \forall Y \{ q(X, Y) \Rightarrow p(Y) \} \} \}$$

1. éliminer les implications ($a \Rightarrow b \equiv \neg a \vee b$)

$$\forall X \{ \neg p(X) \vee \{ \forall Y \{ \neg p(Y) \vee p(f(X, Y)) \} \wedge \neg \forall Y \{ \neg q(X, Y) \vee p(Y) \} \} \}$$

Transformation en forme clausale (2)

2. réduire la portée des négations

$$\neg(a \wedge b) \equiv \neg a \vee \neg b$$

$$\neg(a \vee b) \equiv \neg a \wedge \neg b$$

$$\neg\neg a \equiv a$$

$$\neg(\forall X a) \equiv \exists X(\neg a)$$

$$\neg(\exists X a) \equiv \forall X(\neg a)$$

$$\forall X\{\neg p(X) \vee \{\forall Y\{\neg p(Y) \vee p(f(X, Y))\} \wedge \exists Y\{q(X, Y) \wedge \neg p(Y)\}\}\}$$

3. “standardiser” les variables pour les rendre uniques

$$\forall X\{\neg p(X) \vee \{\forall Y\{\neg p(Y) \vee p(f(X, Y))\} \wedge \exists W\{q(X, W) \wedge \neg p(W)\}\}\}$$

Transformation en forme clausale (3)

4. éliminer les quantificateurs existentiels (skolemisation)

$$\forall Y \exists X p(X, Y) \equiv \forall Y p(g(Y), Y)$$

si \exists n'est pas dans la portée d'un \forall alors on crée une constante $\exists X p(X) \equiv P(a)$

$$\forall X \{ \neg p(X) \vee \{ \forall Y \{ \neg p(Y) \vee p(f(X, Y)) \} \wedge \{ q(X, g(X)) \wedge \neg p(g(X)) \} \} \}$$

5. convertir en forme "prenex" (tous les quantificateurs au début)

$$\forall X \forall Y \{ \neg p(X) \vee \{ \{ \neg p(Y) \vee p(f(X, Y)) \} \wedge \{ q(X, g(X)) \wedge \neg p(g(X)) \} \} \}$$

Transformation en forme clauseale (4)

6. transformer la matrice en forme conjonctive normale
(conjonction de disjonctions) $a \vee \{b \wedge c\} \equiv \{a \vee b\} \wedge \{a \vee c\}$

$$\forall X \forall Y \{ \{ \neg p(X) \vee \neg p(Y) \vee p(f(X, Y)) \} \\ \wedge \{ \neg p(X) \vee q(X, g(X)) \} \wedge \{ \neg p(X) \vee \neg p(g(X)) \} \}$$

7. éliminer les quantificateurs universels
8. éliminer les \wedge pour obtenir une suite de disjonctions (clauses)

$$\{ \neg p(X) \vee \neg p(Y) \vee p(f(X, Y)) \} \\ \{ \neg p(X) \vee q(X, g(X)) \} \\ \{ \neg p(X) \vee \neg p(g(X)) \}$$

On peut prouver que si un ensemble de clauses est impossible à satisfaire, il en est de même pour le système original.

Principe de résolution

$$\frac{a \vee b}{\frac{\neg a \vee c}{b \vee c}}$$

car a et $\neg a$ ne peuvent être vrais en même temps, il faut donc que b ou c soient vrais pour conserver la véracité du système. On peut aussi le comprendre en sachant que la deuxième clause équivaut à $a \Rightarrow c$.

Résolution en logique des propositions

$$\begin{array}{lll} p & p & (1) \\ p \wedge q \Rightarrow r & \neg p \vee \neg q \vee r & (2) \\ s \vee t \Rightarrow q & \neg s \vee q & (3) \\ & \neg t \vee q & (4) \\ t & t & (5) \end{array}$$

on veut prouver r , donc on ajoute $\neg r$ aux clauses existantes.

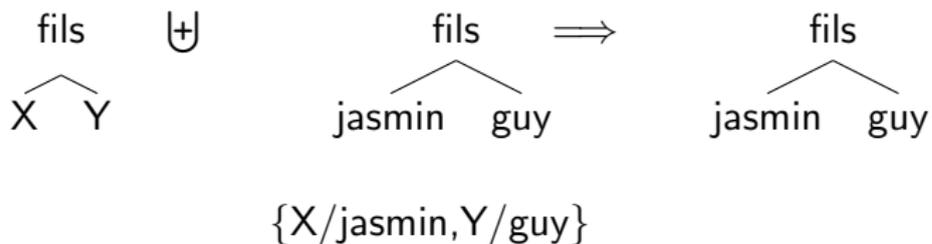
$$\begin{array}{r} \neg r \\ \neg p \vee \neg q \vee r \quad (2) \\ \hline \neg p \vee \neg q \\ p \quad (1) \\ \hline \neg q \\ q \vee \neg t \quad (4) \\ \hline \neg t \\ t \quad (5) \\ \hline \square \end{array}$$

Unification

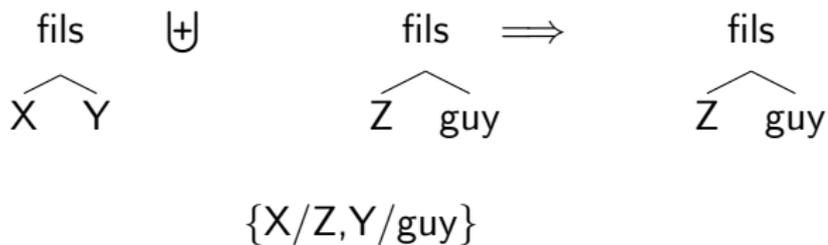
En logique des propositions, l'appariement des clauses est direct mais en logique des prédicats, c'est plus complexe à cause des variables.

L'unification est le processus de détermination de la possibilité de rendre deux expressions identiques au moyen de substitution de variables.

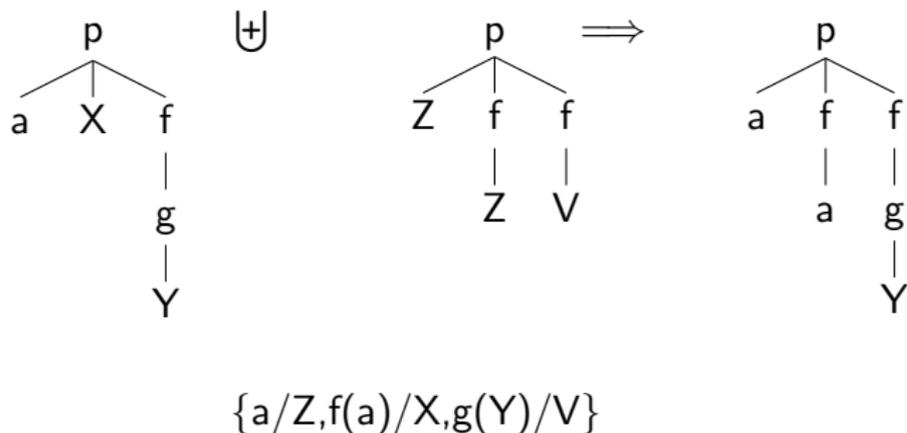
Exemples d'unification d'arbres



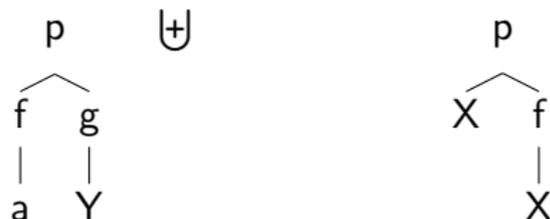
Exemples d'unification d'arbres



Exemples d'unification d'arbres

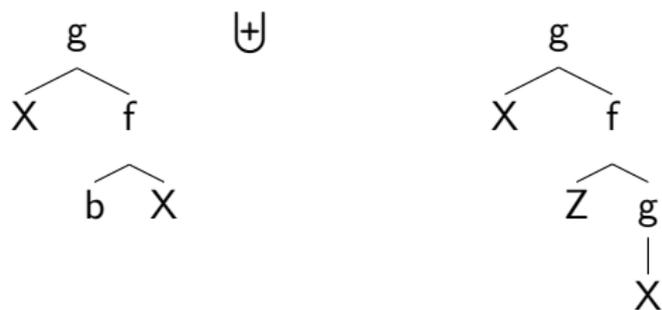


Exemples d'unification d'arbres



non unifiable car $f(f(a)) \neq g(Y)$

Exemples d'unification d'arbres



non-unifiable car $X \neq g(X)$

Procédure d'unification

```
procedure UNIFIER(X,Y)
  if X=Y then
    retourner ()
  else if VARIABLE(X) then
    retourner(UNIFIERVER(X,Y))
  else if VARIABLE(Y) then
    retourner(UNIFIERVER(Y,X))
  else if CONSTANT(X) ou CONSTANT(Y) then
    retourner(echec)
  else if LONGUEUR(X)  $\neq$  LONGUEUR(Y) then
    retourner(echec)
  else
    G := []
    for I := 1 to LONGUEUR(X) do
      S := UNIFIER(X[I],Y[I])
      if S = echec then
        retourner(echec)
      end if
      G := COMPOSER(G,S)
      X := SUBSTITUER(X,G)
      Y := SUBSTITUER(Y,G)
    end for
  end if
end if

procedure UNIFIERVER(X,Y)
  if INCLUS(X,Y) then
    retourner(echec)
  else
    retourner([X/Y])
  end if
end if
```

Exemples de traduction en logique des prédicats sous forme de clauses

- | | | |
|---|---|--|
| 1 | Marcus était un homme | $homme(marcus)$ |
| 2 | Marcus était un pompéien | $pompeien(marcus)$ |
| 3 | Tous les pompéiens étaient des Romains | $\neg pompeien(X1) \vee romain(X1)$ |
| 4 | César était un dirigeant | $dirigeant(cesar)$ |
| 5 | Tous les romains étaient soit loyaux à César ou l'haïssaient | $\neg romain(X2) \vee$
$loyala(X2, cesar) \vee hair(X2, cesar)$ |
| 6 | Tout le monde est loyal à quelqu'un | $loyala(X3, f(X3))$ |
| 7 | Les hommes essaient seulement d'assassiner les dirigeants auxquels ils ne sont pas loyaux | $\neg homme(X4) \vee$
$\neg dirigeant(Y1) \vee$
$\neg essaiedassassiner(X4, Y1) \vee$
$\neg loyala(X4, Y1)$ |
| 8 | Marcus a essayé d'assassiner César | $essaiedassassiner(marcus, cesar)$ |

Preuve par résolution: Marcus haït-il César?

$\neg \text{hair}(\text{marcus}, \text{cesar})$
 $\neg \text{romain}(X2) \vee \text{loyala}(X2, \text{cesar})$ marcus/X2
 $\vee \text{hair}(X2, \text{cesar})$

$\neg \text{romain}(\text{marcus}) \vee \text{loyala}(\text{marcus}, \text{cesar})$
 $\neg \text{pompeien}(X1) \vee \text{romain}(X1)$ marcus/X1

$\neg \text{pompeien}(\text{marcus}) \vee \text{loyala}(\text{marcus}, \text{cesar})$
 $\text{pompeien}(\text{marcus})$

$\text{loyala}(\text{marcus}, \text{cesar})$
 $\neg \text{homme}(X4) \vee \neg \text{dirigeant}(Y1) \vee$ marcus/X4,
 $\neg \text{essaiedassassiner}(X4, Y1) \vee$ cesar/Y1
 $\neg \text{loyala}(X4, Y1)$

$\neg \text{homme}(\text{marcus}) \vee \neg \text{dirigeant}(\text{cesar}) \vee$
 $\neg \text{essaiedassassiner}(\text{marcus}, \text{cesar})$
 $\text{homme}(\text{marcus})$

$\neg \text{dirigeant}(\text{cesar}) \vee \neg \text{essaiedassassiner}(\text{marcus}, \text{cesar})$
 $\text{dirigeant}(\text{cesar})$

$\neg \text{essaiedassassiner}(\text{marcus}, \text{cesar})$
 $\text{essaiedassassiner}(\text{marcus}, \text{cesar})$



Traduction en clauses avec fonctions évaluable

1	Marcus était un homme	$homme(marcus)$
2	Marcus était un pompéien	$pompeien(marcus)$
3	Marcus est né en 40 après J.C.	$nait(marcus, 40)$
4	Tous les hommes sont mortels	$\neg homme(X1) \vee mortel(X1)$
5,6	Tous les pompéiens moururent lors de l'éruption du volcan en 79	$eruption(volcan, 79)$ $\neg pompeien(X2) \vee meurt(X2, 79)$
7	Aucun mortel ne vit plus de 150 ans	$\neg mortel(X3) \vee nait(X3, T1) \vee$ $gt(T2 - T1, 150) \vee$ $mort(X3, T2)$
8	Nous sommes maintenant en 2009	$maintenant = 2009$
9a,b	un vivant n'est pas mort	$\neg vivant(X4, T3) \vee \neg mort(X4, T3)$ $mort(X5, T4) \vee vivant(X5, T4)$
10	un mort reste mort !!!	$\neg meurt(X6, T5) \vee \neg gt(T6, T5) \vee$ $mort(X6, T6)$

Est-ce que Marcus est vivant?

Preuve par résolution (avec réduction et égalité)

$vivant(marcus, maintenant)$

$\neg vivant(X4, T3) \vee \neg mort(X4, T3)$

$marcus/X4,$
 $maintenant = T3$

$\neg mort(marcus, maintenant)$

$\neg meurt(X6, T5) \vee gt(T6, T5) \vee$

$mort(X6, T6)$

$marcus/X6,$
 $maintenant/T6$

$\neg meurt(marcus, T5) \vee \neg gt(maintenant, T5)$

$\neg pompeien(X2) \vee meurt(X2, 79)$

$marcus/X2, 79/T5$

$\neg pompeien(marcus) \vee \neg gt(maintenant, 79)$

$maintenant = 2009$

$\neg pompeien(marcus) \vee \neg gt(2009, 79)$

réduction de $2009 > 79$

$\neg pompeien(marcus)$

$pompeien(marcus)$

□

Stratégies de preuve

La procédure de résolution nous dit comment combiner les clauses mais il reste à déterminer l'ordre dans lequel faire les combinaisons.

Exemples de stratégies:

- ▶ en largeur
 - + complète
 - explosion combinatoire
- ▶ ensemble de support: part de la contradiction comme ensemble de départ et on ne considère que les clauses pouvant se résoudre avec une de cet ensemble qu'on "élargit" peu à peu
- ▶ "unit preference": lorsque c'est possible on résout avec des clauses unitaires de façon à diminuer le nombre de littéraux dans les clauses résultantes
- ▶ "linear input form": à chaque étape on résout la clause obtenue le plus récemment. C'est plus efficace mais risque de ne plus être complet.

Logiques non-classiques

La logique des prédicats est quelque peu restrictive pour décrire toutes les situations.

- ▶ logique non-monotone
- ▶ logique des défauts
- ▶ logique modale (nécessité, obligation, savoir, croyance ...)
- ▶ logique temporelle
- ▶ logique floue
- ▶ logique probabiliste