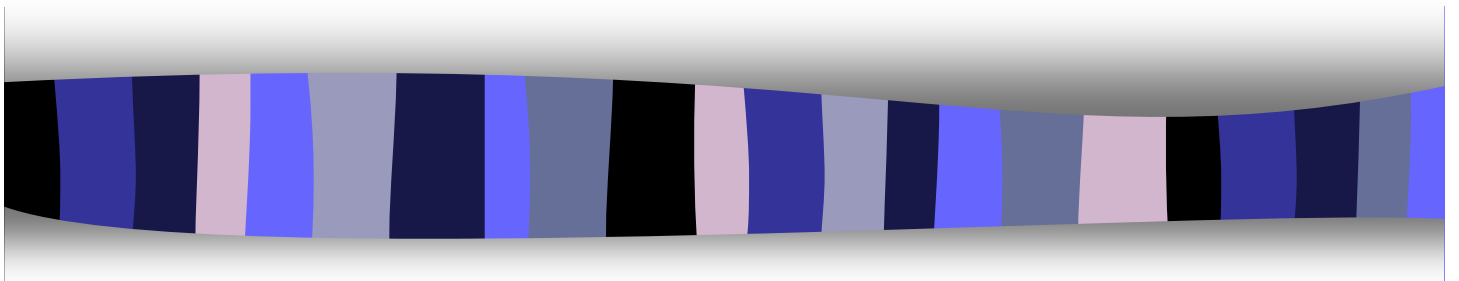


Mise en oeuvre





Plan

■ Introduction

- L'approche déductive
- Complétion d'hypothèses

■ Résolution : propriétés

- Pour la déduction
- Pour la réfutation

■ Méthode de réfutation

- Principes fondateurs
- Procédure
- Vocabulaire
- Exemples de preuve et de question

■ Stratégies de mise en oeuvre

- Possibilités théoriques
- Vocabulaire et étude des stratégies
- Stratégies en largeur et profondeur d'abord
- Stratégies de l'ensemble de support
- Stratégies linéaires, à entrées et unitaires



Introduction

Les approches

L'approche déductive

C'est l'approche suivie dans les systèmes formels : il s'agit de "dédire" la fbf recherchée, en partant des axiomes logiques et non logiques, et par un enchaînement rigoureux d'inférences

$X \rightarrow Y_1 \rightarrow Y_2 \rightarrow X \rightarrow Z_1 \rightarrow Z_2 \rightarrow Z_3 \dots \rightarrow C$

X axiome logique, Y axiome non logique

Z fbf inférée, C fbf déduite

- ➡ Cette approche syntaxique est possible, mais suppose une axiomatique "lourde" et est difficile à mettre en œuvre
- ➡ En outre, la règle de résolution, à elle seule, ne permet pas de "tout" déduire !

Les approches

Complétion d'hypothèses

Par un raisonnement par cas :

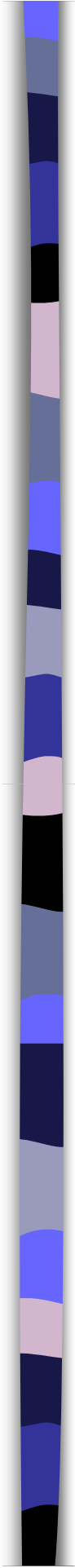
- On considère successivement les différentes valeurs que peut prendre un paramètre
- On envisage chacune de ces valeurs comme une hypothèse supplémentaire

Par un raisonnement par l'absurde :

- On ajoute la négation du but aux hypothèses du problème
- On met en évidence une contradiction

Approche en démonstration automatique :

L'approche est réfutative, car la règle de résolution à elle seule la rend possible !!!
La méthode de réfutation est due à Al. Robinson (1965)



Mise en œuvre de la méthode de réfutation par résolution

Résolution : propriétés

Pour la déduction

- Une règle d'inférence est “saine” si, pour tout ensemble E de fbfs, toute fbf G déductible de E par application de cette règle est une conséquence logique de E

La résolution est saine pour la déduction

- Un groupe de règles d'inférence est “complet pour la déduction” si, pour tout ensemble E de fbfs, toute fbf G conséquence logique de E est déductible de E par application des règles de ce groupe

La résolution n'est pas complète pour la déduction

Résolution : propriétés

Pour la réfutation

Une réfutation d'un ensemble de clauses E est une déduction par résolution de $\{\}$ à partir de E

- S'il existe une réfutation de E , alors E est insatisfiable

La résolution est saine pour la réfutation

- Si E est insatisfiable, alors il existe une réfutation de E

La résolution est complète pour la réfutation

- Théorème de complétude généralisé :

Un ensemble de clauses est insatisfiable ssi il en existe une réfutation par résolution

Méthode de réfutation

Principes fondateurs

Le théorème de Skolem plus celui de complétude permettent de fonder la méthode de réfutation par résolution. En partant du problème de déduction, les énoncés suivants sont équivalents :

G est conséquence logique de H_1, H_2, \dots, H_n

$(H_1 \wedge H_2 \wedge \dots \wedge H_n \Rightarrow G)$ est une fbf valide

$\{ H_1 ; H_2 ; \dots ; H_n ; \neg G \}$ est insatisfiable

L'ensemble des clauses issues de H_1, H_2, \dots, H_n et $\neg G$ est insatisfiable

Il existe une réfutation par résolution des clauses issues de H_1, H_2, \dots, H_n et $\neg G$

Méthode de réfutation

Procédure

1. Ajouter la négation du but G aux n hypothèses H_1, H_2, \dots, H_n
2. Etablir l'ensemble des clauses issues de H_1, H_2, \dots, H_n et $\neg G$
3. Tant que la clause vide n'est pas obtenue et tant qu'il est possible de créer de nouveaux résolvants :
 - choisir deux clauses parentes
 - former le résolvant
 - renommer ses variables
 - l'ajouter à l'ensemble des clauses

Remarques :

- ☞ Cette procédure est non déterministe : comment choisir les parents ?
- ☞ Les inférences sont irrévocables : le mécanisme est monotone

Méthode de réfutation

Vocabulaire

- *Graphe de dérivation*

Graphe de "tous" les résolvents qui peuvent être dérivés des clauses initiales

- *Graphe de recherche*

Partie du graphe de dérivation effectivement produite pendant la recherche, en application d'une stratégie

- *Graphe de réfutation*

Partie du graphe de dérivation qui, contenant la clause vide, réfute $\neg G$

- *Lien avec le problème de déduction*

Si le graphe de recherche contient un graphe de réfutation, alors il est prouvé que G est conséquence logique de H_1, H_2, \dots, H_n

Méthode de réfutation

Un premier exemple

Tous les élèves d'une classe pratiquent l'anglais ou l'allemand. Aucun ne fait à la fois de l'anglais et du latin. Seuls ceux qui font du latin font du grec. Paul fait du grec".

Problème de type "preuve" : montrer que Paul fait de l'allemand (G)

Représentation :

Phrase 1 : $\forall x (anglais(x) \vee allemand(x))$
soit $\{anglais(x) \vee allemand(x)\}$

Phrase 2 : $\neg \exists x (anglais(x) \wedge latin(x))$
soit $\{\neg anglais(y) \vee \neg latin(y)\}$

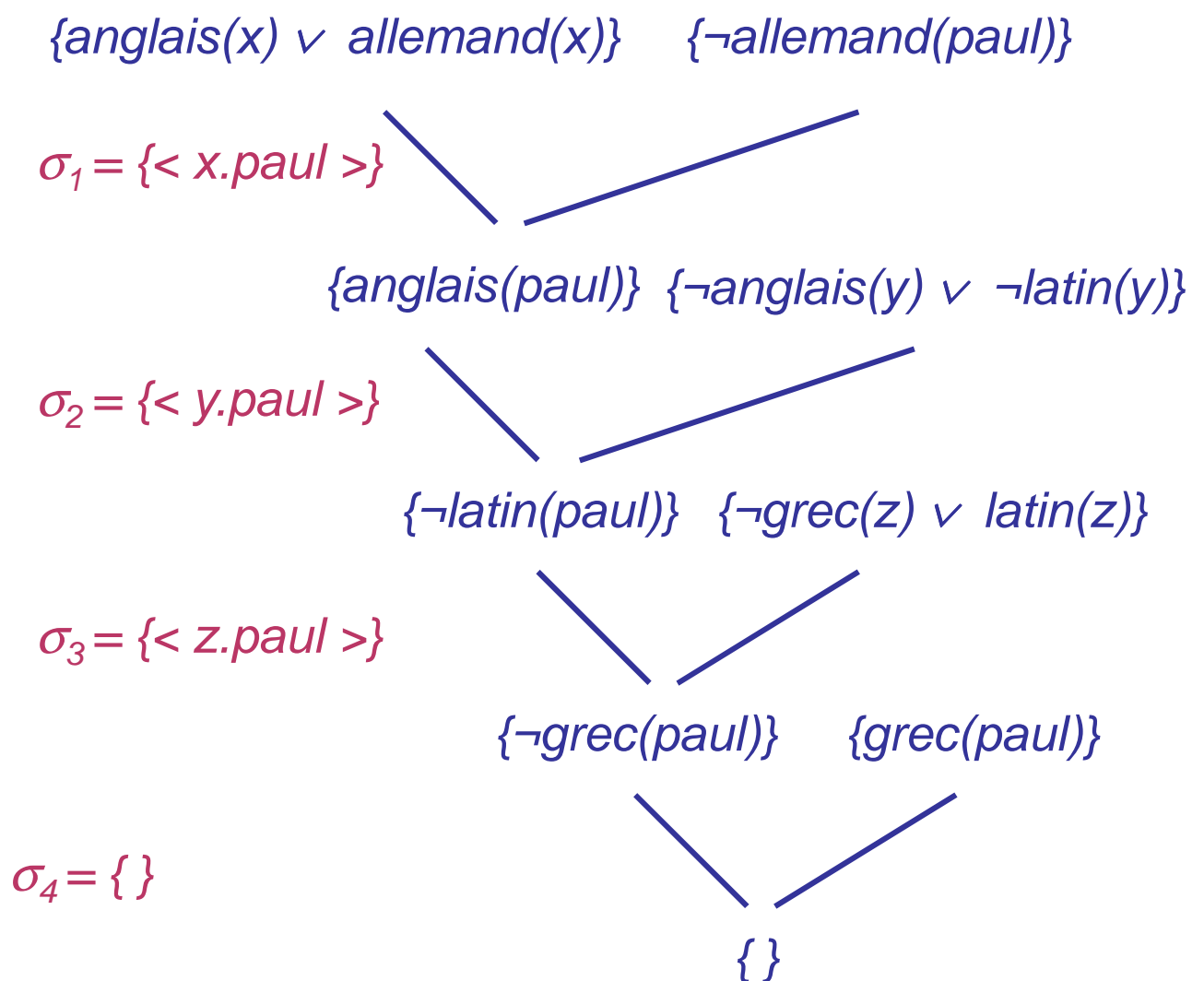
Phrase 3 : $\forall x (grec(x) \Rightarrow latin(x))$
soit $\{\neg grec(z) \vee latin(z)\}$

Phrase 4 : $grec(paul)$ soit $\{grec(paul)\}$

$\neg G$: $\neg allemand(paul)$ soit $\{\neg allemand(paul)\}$

Méthode de réfutation

Preuve : « Paul fait de l'allemand »



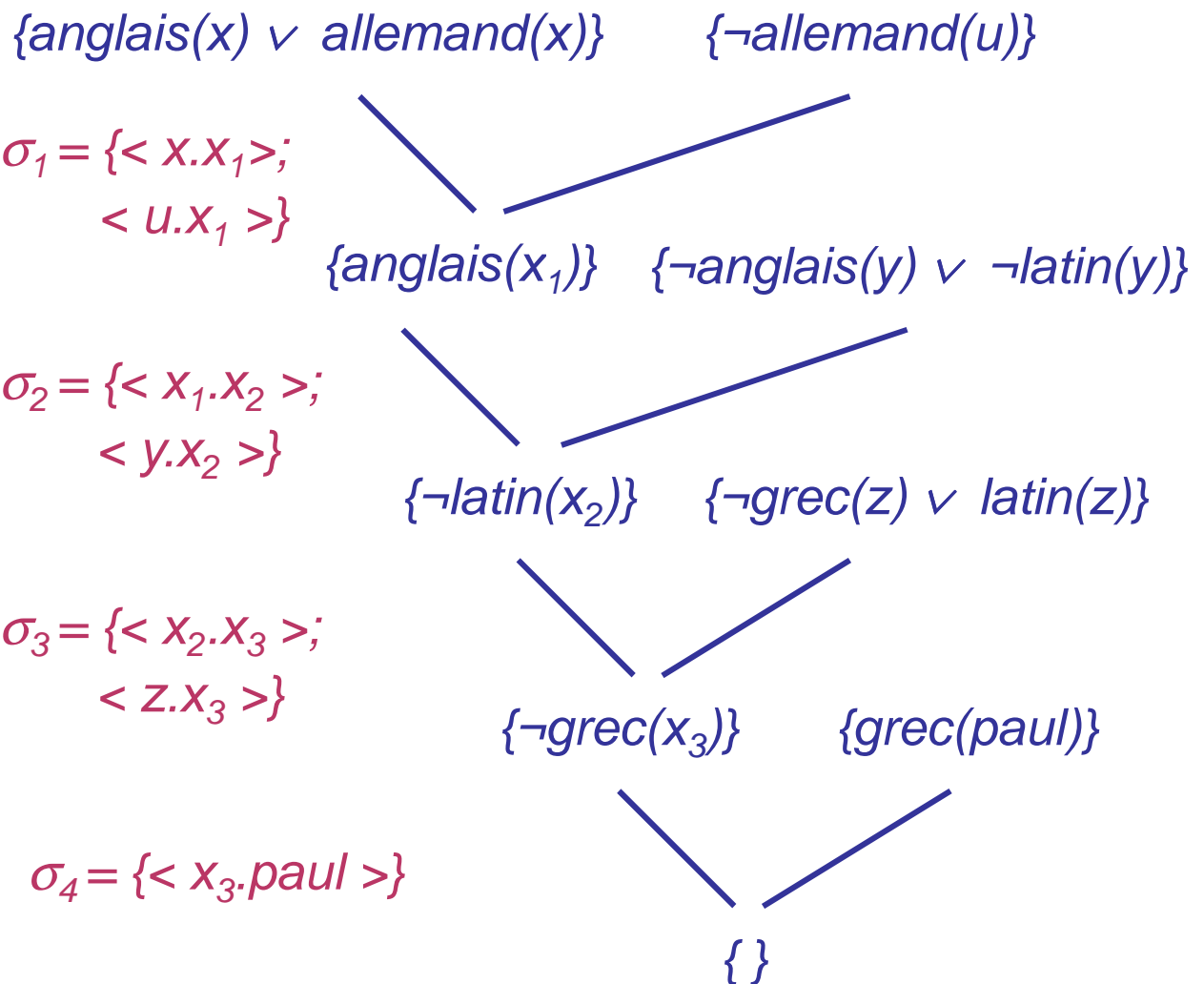
G est conséquence logique de H_1, H_2, H_3, H_4

Méthode de réfutation

Question : « qui fait de l'allemand ? »

Soit G cette question : $\exists x \text{ allemand}(x)$

Forme clausale de sa négation : $\{\neg \text{allemand}(u)\}$



Par composition des substitutions : $\langle u, \text{Paul} \rangle$



Stratégies de mise en oeuvre

Possibilités théoriques

Puissance de la méthode

- *Rappel* : Soit E un ensemble de fbfs et G une fbf de $L1$. Existe-t'il une procédure capable de décider, en un temps fini, si G est une conséquence logique de E ?
- *Question* : la Méthode de Réfutation par Résolution est-elle une procédure de décision ? On a vu précédemment que : « Le Calcul des Prédicats est indécidable »
- *Conséquence* : la Méthode de Réfutation par Résolution n'est pas une procédure de décision, *quelle que soit la stratégie de choix des parents*
- *Semi-décidabilité* : il existe des stratégies pour lesquelles cette procédure devient une procédure de semi-décision

Stratégies

Vocabulaire pour les stratégies

Une stratégie est *complète* s'il existe au moins un graphe de réfutation qui satisfait les critères de choix de cette stratégie

Remarques :

- ➡ à ce stade, la méthode de réfutation n'est pas encore déterministe
- ➡ rien ne garantit encore qu'un graphe de réfutation soit effectivement trouvé !
- ➡ la méthode de réfutation est alors elle-même dite complète

Une stratégie est *directement complète* dès lors que l'application effective de cette stratégie conduit à l'obtention d'une réfutation en un nombre fini de pas

Etude des stratégies

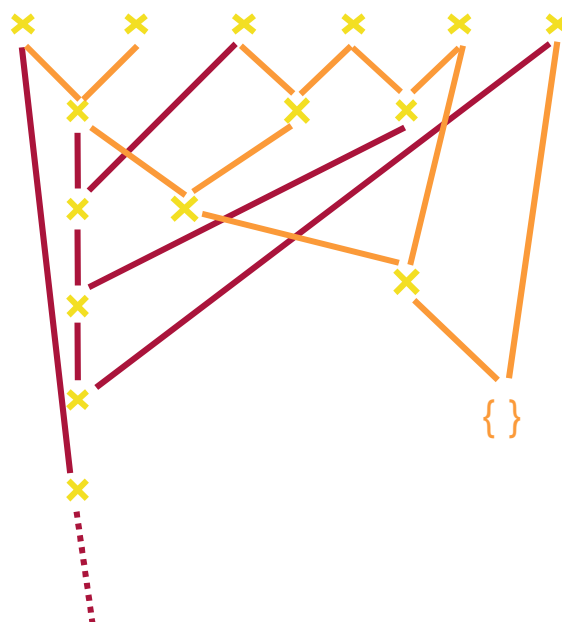
Cadre de l'étude

Soit G conséquence logique de E

On sait donc *au préalable* qu'il existe au moins un graphe de réfutation de G . Reste à le trouver effectivement en un nombre fini de pas

Problématique : étant donné une stratégie S , quelle est sa capacité à permettre le développement d'un graphe de réfutation ?

☞ risque encouru : "rater la clause vide"



Etude des stratégies

■ Problématique

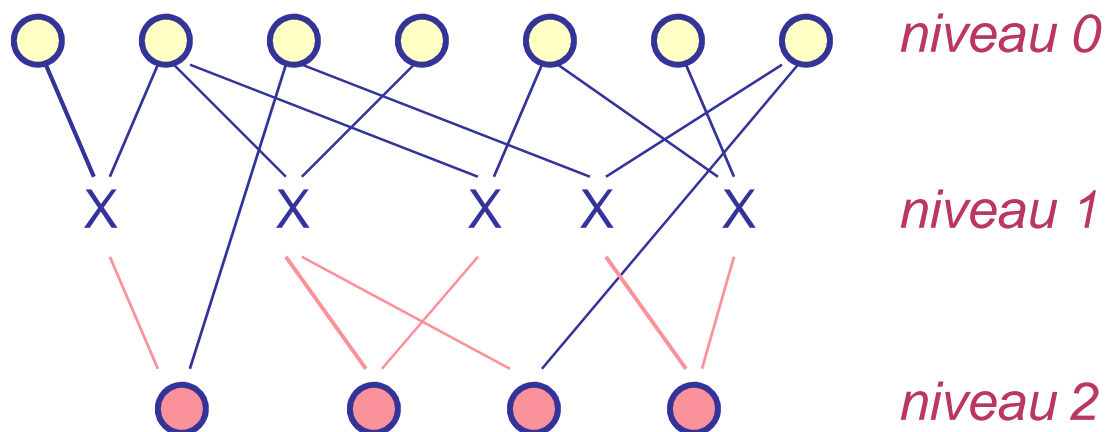
- Deux types de choix :
 - ☞ celui des clauses parentes : de quelles clauses partir ? comment continuer ? quid des clauses issues de $\neg G$?
 - ☞ celui des atomes à unifier
- Peut-on garantir l'obtention d'une preuve ? de la preuve la plus courte ? être efficace ?

■ Types de stratégies

- *Les stratégies générales* (sans volonté de réduire le nombre des clauses) : en largeur ou en profondeur d'abord
- *Les stratégies gestionnaires des arbres de déduction* : à ensemble de support, linéaires, à ancêtres filtrés, ordonnées,

En largeur d'abord

Tous les résolvants de niveau 1 sont produits, puis tous ceux de niveau 2, etc...

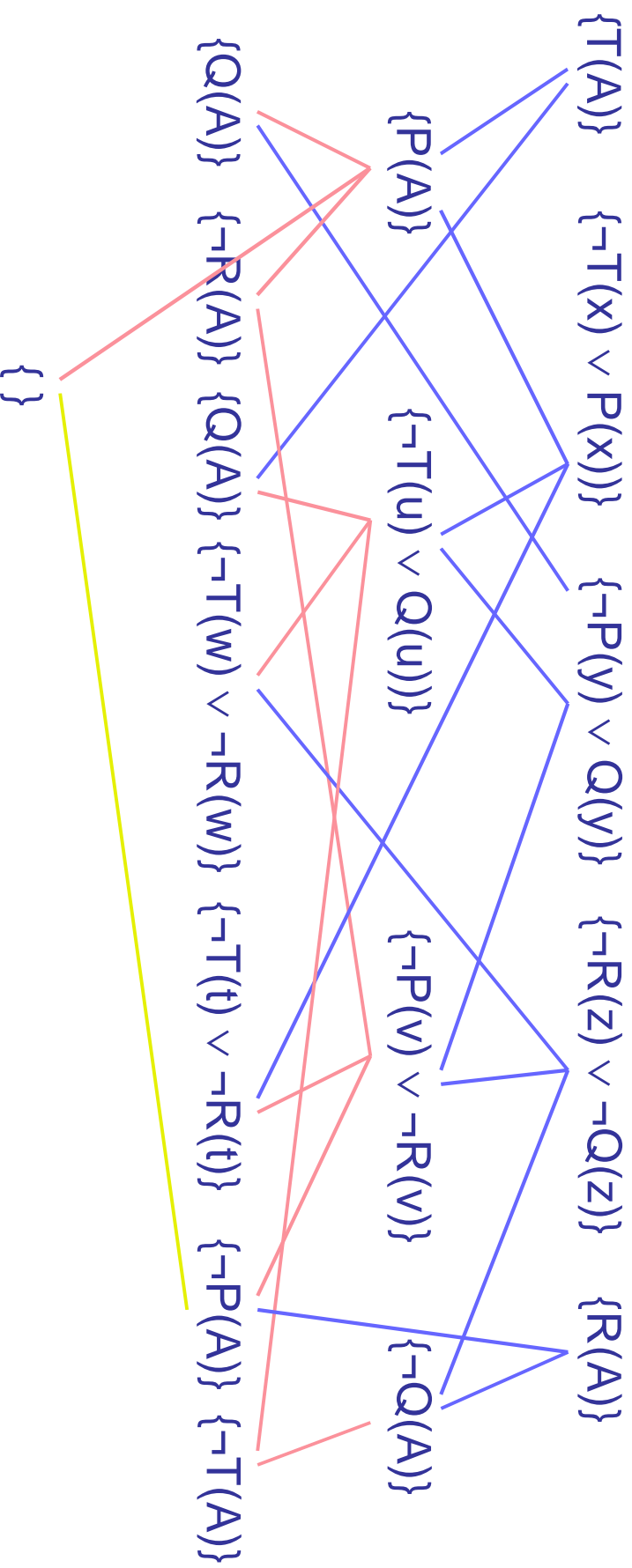


Les stratégies en largeur d'abord sont directement complètes, donnent la plus courte preuve, mais sont très inefficaces



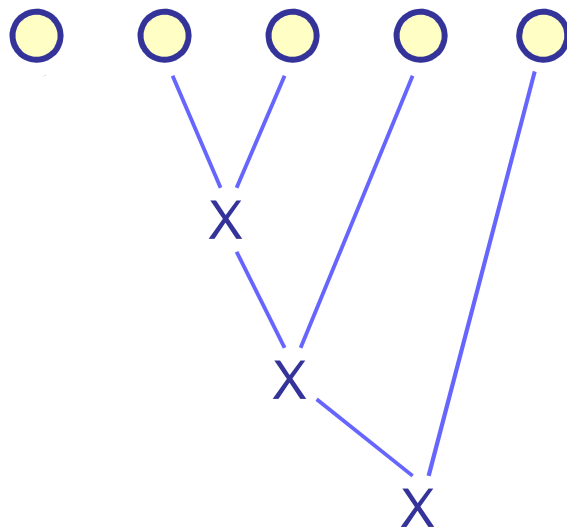
En largeur d'abord

127



En profondeur d'abord

L'un des parents est le dernier résolvant produit, et si ce n'est pas possible, alors on rebrousse chemin (backtracking)



En profondeur d'abord

■ Caractéristiques

Avantages :

- Possibilité, éventuellement, de trouver une clause vide profonde plus rapidement
- Meilleure gestion de la mémoire

Inconvénients :

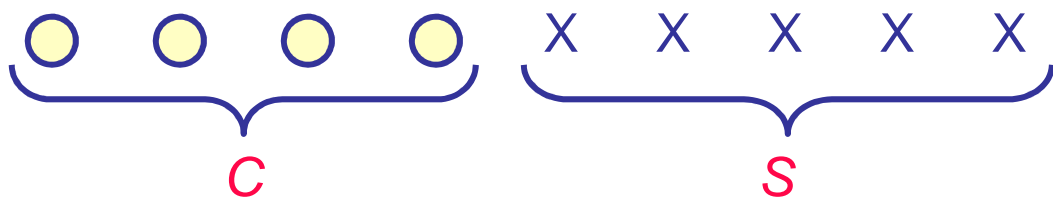
- La stratégie est complète, mais n'est pas directement complète, en raison du risque de se perdre dans une branche infinie
- A priori réservée à l'exploration des arbres finis

■ L'approfondissement itéré

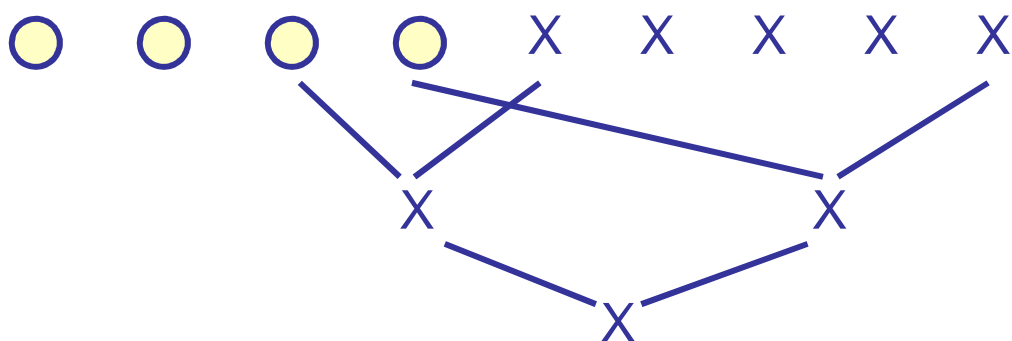
- En profondeur d'abord à un niveau k
- Si échec, en profondeur à un niveau $k+1$, etc....

Ensemble de support

L'un des parents est choisi parmi les clauses d'un ensemble de support S ou parmi les clauses descendantes de S



Exemple :



Ensemble de support

■ Théorèmes

Les stratégies de l'ensemble de support sont complètes ssi l'ensemble C est satisfiable

Si cette condition est vérifiée, toute stratégie de l'ensemble de support en largeur d'abord est directement complète

■ Comment rendre C satisfiable ?

Plusieurs façons pour constituer l'ensemble de support S

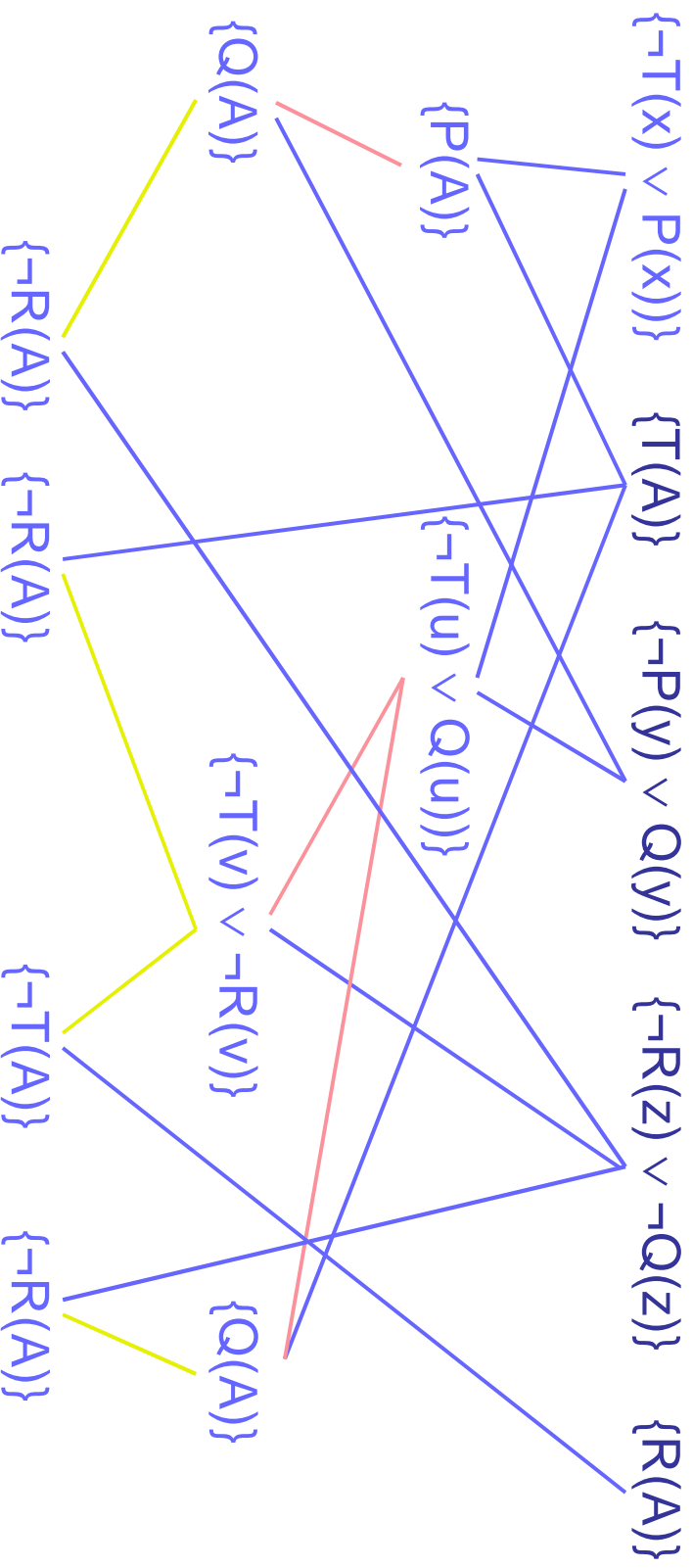
- S ensemble de clauses à littéraux positifs
- S ensemble de clauses à littéraux négatifs
- S ensemble des clauses issues de $\sim G$ (*sous certaines conditions*)



Ensemble de support

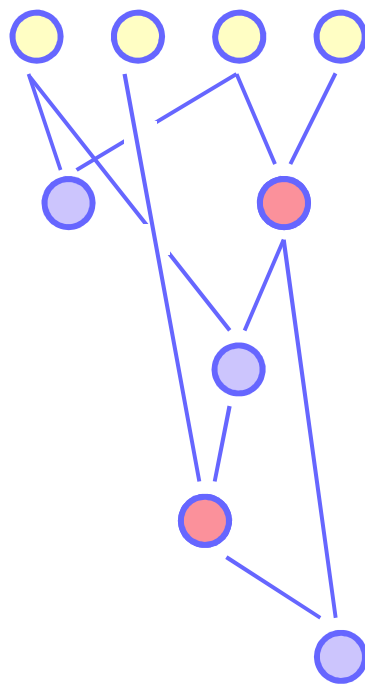
132

Exemple d'application d'une stratégie ensemble de support avec largeur d'abord :



Ancêtres filtrés

Chaque résolvant a un parent qui est une clause initiale ou qui est ancêtre de l'autre parent



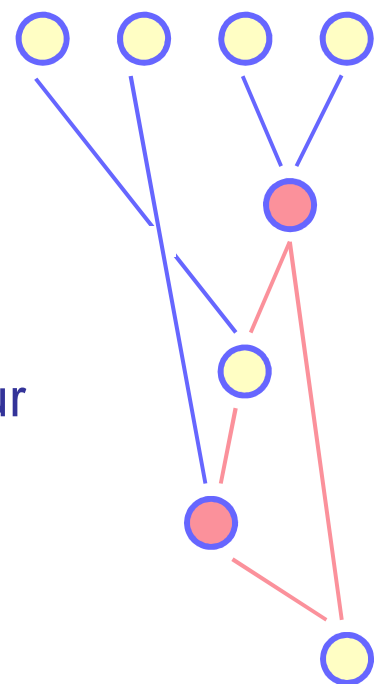
Une stratégie qui serait capable de produire tous les arbres de preuve de ce type serait complète

Stratégies linéaires

Soit un ensemble S de clauses d'entrées. Une stratégie linéaire dérive des clauses C_0, \dots, C_i telles que $C_0 \in S$ et pour tout $i \geq 0$, C_{i+1} est un résolvant de C_i avec une clause de $S \cup \{C_1, \dots, C_i\}$

Caractéristiques

- Stratégies de type "ancêtres filtrés"
- On impose en plus que l'autre parent soit le dernier résolvant produit, avec retour en arrière si nécessaire
- Stratégies complètes, mais pas directement complètes



Stratégies à entrées

Une stratégie à entrées impose que l'un des parents au moins appartienne à l'ensemble de clauses initial

Caractéristiques :

- De telles stratégies ne sont pas complètes
- Exemple :

$$C = \{A \vee B ; A \vee \neg B ; \neg A \vee B ; \neg A \vee \neg B\}$$

Stratégies linéaires avec entrées :

- A la fois linéaires et à entrées. On impose en plus que l'autre parent soit le dernier résolvant produit, avec retour en arrière
- A fortiori non complètes, mais utilisées sous certaines restrictions

Stratégies unitaires

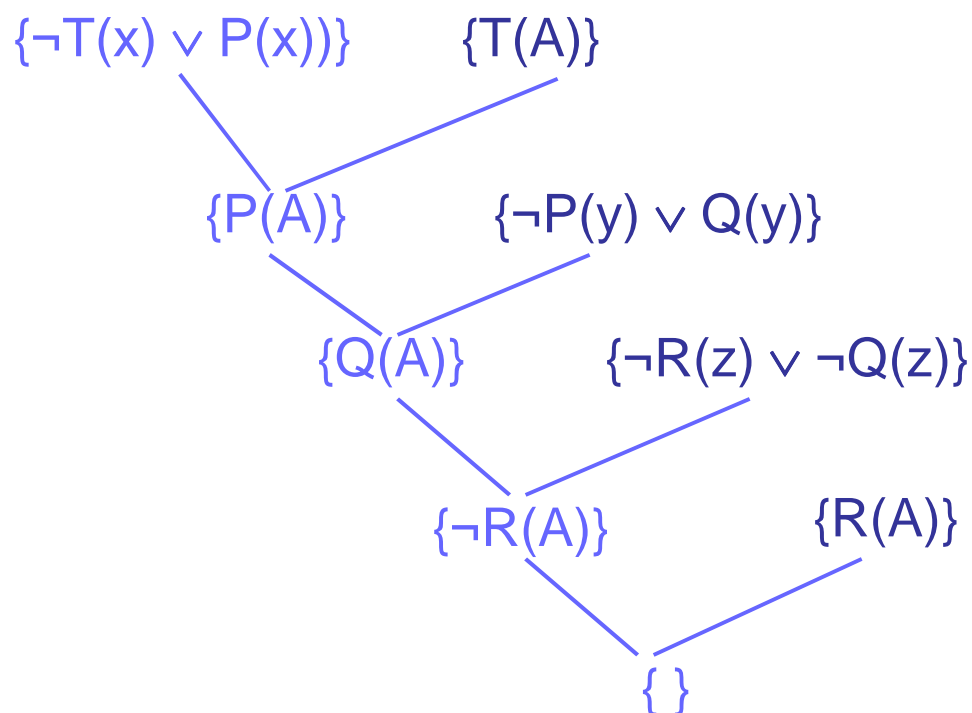
Une stratégie unitaire impose que l'un au moins des deux parents soit une clause unitaire, (i.e.) possédant un et un seul littéral

Caractéristiques

- Les stratégies unitaires sont incomplètes

Ex. : $C = \{A \vee B ; A \vee \neg B ; \neg A \vee B ; \neg A \vee \neg B\}$

- Variante : unitaire + ensemble-de-support



Stratégies d'élimination

Elimination de clauses inutiles

- **Elimination de tautologies :**

Il s'agit d'éliminer toute clause contenant un littéral et sa négation

Exemple : $\{P(u) \vee Q(v) \vee \neg P(u)\}$

- **Elimination de clauses pures :**

Il s'agit d'éliminer toute clause contenant un littéral pur, (ie) un littéral qui ne s'unifie avec aucun littéral opposé d'une autre clause

- **Elimination de clauses subsumées :**

Une clause $\{L_1 \vee \dots \vee L_n\}$ subsume une clause $\{M_1 \vee \dots \vee M_p\}$ s'il existe une substitution σ telle que $\{\sigma(L_1), \dots, \sigma(L_n)\}$ soit inclus dans $\{M_1, \dots, M_p\}$

Exemple : $\{P(x)\}$ subsume $\{P(a) \vee Q(v)\}$ qui peut être éliminée

Stratégies d'élimination

Attachements procéduraux

Prédicats calculables :

Evaluation des atomes concernés plutôt que recherche dans la base de clauses

- ➡ pas d'unification
- ➡ appel d'une procédure pour $PG(5,2)$, puis élimination de la clause $\{Q(x) \vee PG(5,2)\}$
- ➡ appel d'une procédure pour $PG(2,5)$, puis élimination du littéral dans $\{Q(u) \vee PG(2,5)\}$

Fonctions calculables :

Utilisation de fonctions prédéfinies plutôt qu'un calcul résultant d'une déduction ou d'une réfutation

- ➡ $PG(\text{plus}(4,3),2)$

