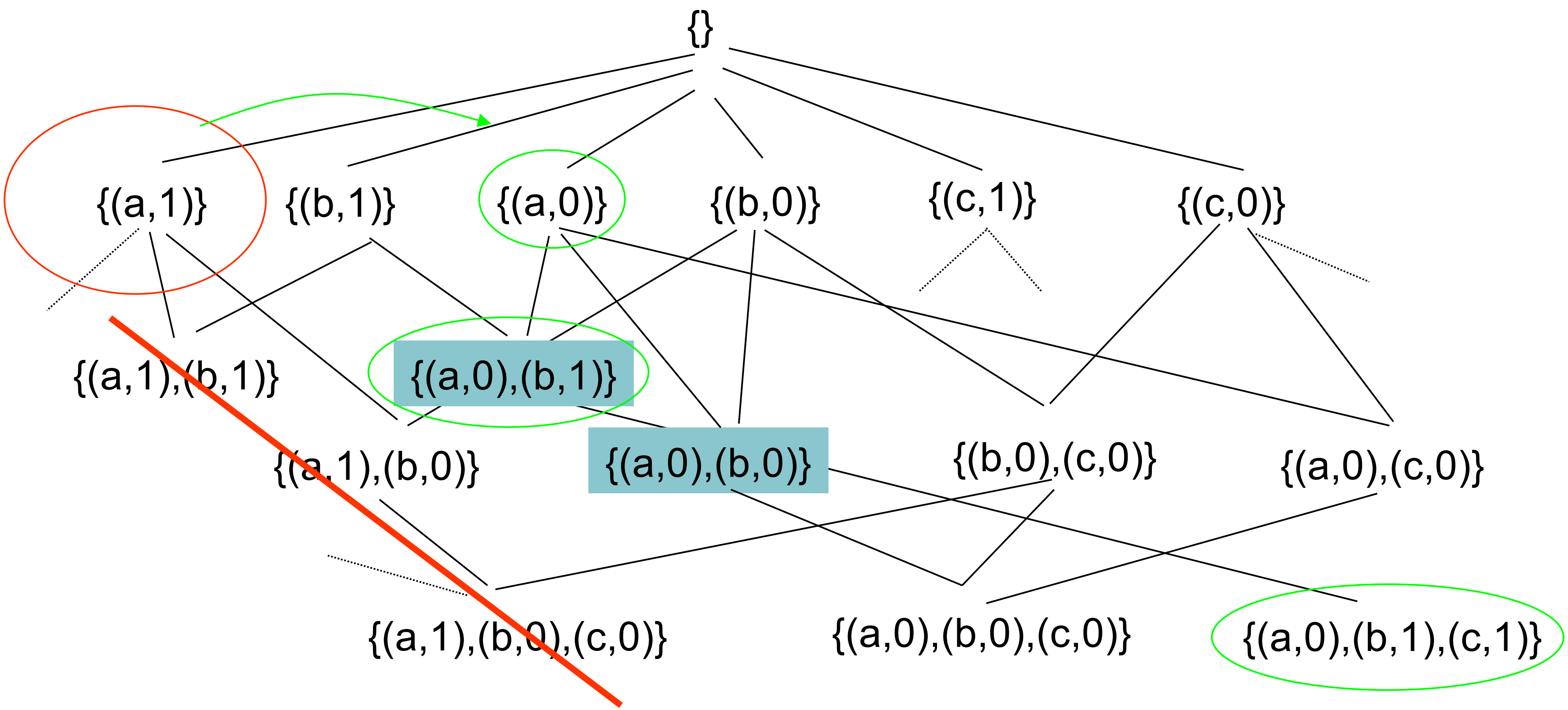
Programme

- Introduction
- Le langage de la LP (syntaxe)
- La sémantique de la LP
- Conséquence logique
- Modélisation
- Méthode des séquents
- Formes normales et clausale
- Méthode de résolution
- Méthode de Davis et Putnam

Algorithme de Davis-Putnam-Logemann-Loveland (1962)

- DPLL attaque le problème de la **satisfiabilité d'une fbf sous forme clausale**
- Il s'agit d'explorer le plus judicieusement possible l'espace des interprétations partielles à la recherche d'un modèle
  - Par opposition à la méthode « brute-force » :
    - Générer une interprétation complète
    - Tester si la valeur de vérité est vraie

Espace des interprétations partielles



Exemple : $(a \vee b) \wedge (\neg b \vee c) \wedge (\neg a)$

Recherche DPLL

Idée principale : inutile de produire une interprétation complète si une partielle est suffisante pour conclure

- Idée 1 : Backtrack - procéder par essai erreur

=> On choisit un symbole s et l'on essaye avec I(s)=1 et si on échoue et on essaye I(s)=0

- Idée 2 : Propagation : tirer parti de tout ce qu'apporte une interprétation partielle

⇒ On élimine les clauses contenant un littéral rendu vrai et on élimine les littéraux rendus faux des autres clauses

- Idée 3 : Heuristique – on choisit en priorité les symboles (et valeur) évitant de générer un « backtrack »

⇒ Le choix d'un littéral dans une clause unitaire ou d'un littéral pur est une bonne heuristique

Algorithme DPLL

Algo : DPLL
Données : F un ens. de clauses
Résultat : *vrai* (satisfiable) ou *faux* (insatisfiable)
Début

si F={} alors
  retourner vrai ;
si Ø appartient à F alors
  retourner faux ;
si il existe l un littéral pur dans F alors
  retourner DPLL(F[l]) ;
si il existe un littéral l dans une clause unitaire {l} de F alors
  retourner DPLL(F[l]) ;
s ← choisir un symbole dans F;
retourner DPLL(F[s]) ou DPLL(F[¬s]);

Fin

Exemple

{ {a,b} , {¬a,c,e} , {¬b,d} , {c} , {b,f} , {a,d,¬e} , {b,¬c,¬e} , {b,¬f} , {¬a,¬d} }

c dans une clause unitaire

{ {a,b} , {¬b,d} , {b,f} , {a,d,¬e} , {b, ¬e} , {b,¬f} , {¬a,¬d} }

¬e littéral pur

{ {a,b} , {¬b,d} , {b,f} , {b,¬f} , {¬a,¬d} }

soit a

{ {¬b,d} , {b,f} , {b,¬f} , {¬d} }

¬d dans une clause unitaire

{ {¬b} , {b,f} , {b,¬f} }

¬b dans une clause unitaire

{ {f} , {¬f} }

f dans une clause unitaire

{ ø }

faux

{ {b} , {¬b,d} , {b,f} , {b,¬f} }

b dans une clause unitaire

{ {d} }

d dans une clause unitaire

{ }

vrai

Exemple

{ {a,b} , {¬a,c,e} , {¬b,d} , {c} , {b,f} , {a,d,¬e} , {b,¬c,¬e} , {b,¬f} , {¬a,¬d} }

c dans une clause unitaire 1

{ {a,b} , {¬b,d} , {b,f} , {a,d,¬e} , {b, ¬e} , {b,¬f} , {¬a,¬d} }

¬e littéral pur 0

{ {a,b} , {¬b,d} , {b,f} , {b,¬f} , {¬a,¬d} }

soit a

{ {¬b,d} , {b,f} , {b,¬f} , {¬d} }

¬d dans une clause unitaire

{ {¬b} , {b,f} , {b,¬f} }

¬b dans une clause unitaire

{ {f} , {¬f} }

f dar

{ ø }

faux

soit ¬a 0

{ {b} , {¬b,d} , {b,f} , {b,¬f} }

b dans une clause unitaire 1

{ {d} }

d dans une clause unitaire 1

{ }

vrai

Lorsque DPLL retourne vrai, on peut exhiber un modèle en remontant la branche (et en choisissant n'importe quelle valeur pour les symboles non valués) :

I(d)=1, I(b)=1, I(a)=0, I(e)=0, I(c)=1 et I(f)=0 ou 1

Correction et complétude de DPLL

- Soit F une formule sous forme clausale :

  DPLL(F) retourne vrai ssi F est satisfiable

- Preuve (idée) :
  - Voir que le lemme

    F insatisfiable ssi F[l] insatisfiable et F[l] insatisfiable

    est équivalent à

    F satisfiable ssi F[l] satisfiable ou F[l] satisfiable
  - **Correction (=>)** : par récurrence sur la longueur de la branche conduisant à vrai
  - **Complétude (<=)** : par induction sur le nombre de symboles propositionnels (similaire à la complétude de la méthode de résolution)