DIC9150 Concepts fondamentaux de l'informatique cognitive

Satisfaction de contraintes I
(Constraint Satisfaction Problem, CSP)

Roger Villemaire

Département d'informatique
UQAM

26 septembre 2023

UQÀM
Université du Québec à Montréal

Plan

Plan

Problème : Disposition des invités

- On veut disposer des invités autour d'une table circulaire. Il y a bien le même nombre de places que de convives, malheureusement certaines personnes ne peuvent pas être assises côte à côte :
  - Béatrice, Charles et François ne peuvent pas être assis à côté d'Albert depuis qu'il a obtenu cette promotion tant convoitée !
  - Béatrice et Charles (ainsi que François et Élise) sont des ex. qu'il vaudrait mieux ne pas asseoir côte à côte !
  - Pour une raison pas très claire, Denis ne peut être placé au côté ni d'Élise, ni de Charles !
  - Finalement, François veut absolument être assis à côté de Béatrice pour lui parler de son tout nouveau projet et ainsi profiter de ses conseils !

Traitement algorithmique de la connaissance

- Représenter la situation, d'une façon exploitable par la machine
- Trouver une méthode algorithmique pour effectuer le traitement cognitif
  - Ceci doit être assez simple pour être efficace en pratique

Problème de satisfaction de contraintes

- La situation est représentée sous forme
  - d'un ensemble de *variables* $X, Y, Z, \ldots$ qui peuvent prendre des valeurs dans leurs domaines $X \in Dom_X, Y \in Dom_Y, Z \in Dom_Z, \ldots$
  - d'un ensemble $\mathcal{C}$ de *contraintes* que les valeurs choisies des variables doivent satisfaire
- La fonction cognitive est effectuée en trouvant une solution, c.-à-d. des valeurs précises aux variables pour que toutes les contraintes soient satisfaites.

Inférence et Recherche

- Une méthode usuelle pour trouver une solution pour un ensemble de contraintes est d'appliquer les principes suivants :
  - l'*inférence* : déduire des restrictions sur les domaines des variables, tant que c'est possible
  - la *recherche* : puis essayer toutes les combinaisons de valeurs parmi celles qu'il reste dans les domaines des variables

Disposition des invités

- On a 6 invités, donc 6 places qu'on numérote $0, 1, 2, 3, 4, 5$
- À chaque personne on doit assigner une place :
  - on a donc six variables A, B, C, D, E, F représentant la place qui sera assignée à Albert, Béatrice, Charles, Denis, Élise et François, respectivement
  - toutes nos variables ont donc *au départ* le même domaine, soit $\{0, 1, 2, 3, 4, 5\}$
- Pour ce qui est des contraintes, elles sont de trois types :
  - X et Y ne doivent pas être assis côte à côte,
  - X et Y doivent être assis côte à côte,
  - $X \neq Y$, pour X et Y des variables différentes, car deux personnes doivent être assises sur des chaises différentes.

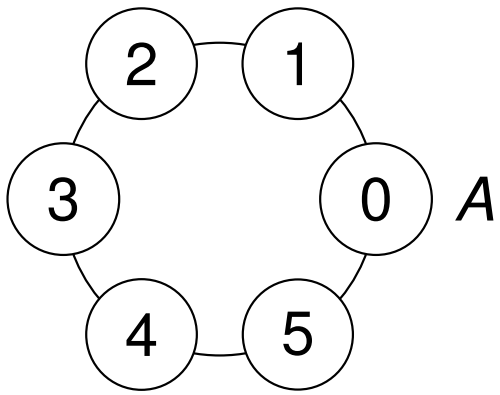
Considérations algorithmiques

- Il est souvent préférable de tenter de réduire le nombre de variables au maximum car :
  - il pourra alors être plus facile de trouver des méthodes d'inférence, réduisant le nombre de possibilités pour les valeurs des variables
  - il y aura moins de variables, donc potentiellement moins de possibilités à tenter dans la phase de recherche
- La façon de représenter un problème peut avoir un impact considérable sur la performance. En pratique il est souhaitable de tenter plusieurs représentations et de vérifier expérimentalement la performance à l'aide des meilleurs *solveurs* de contraintes

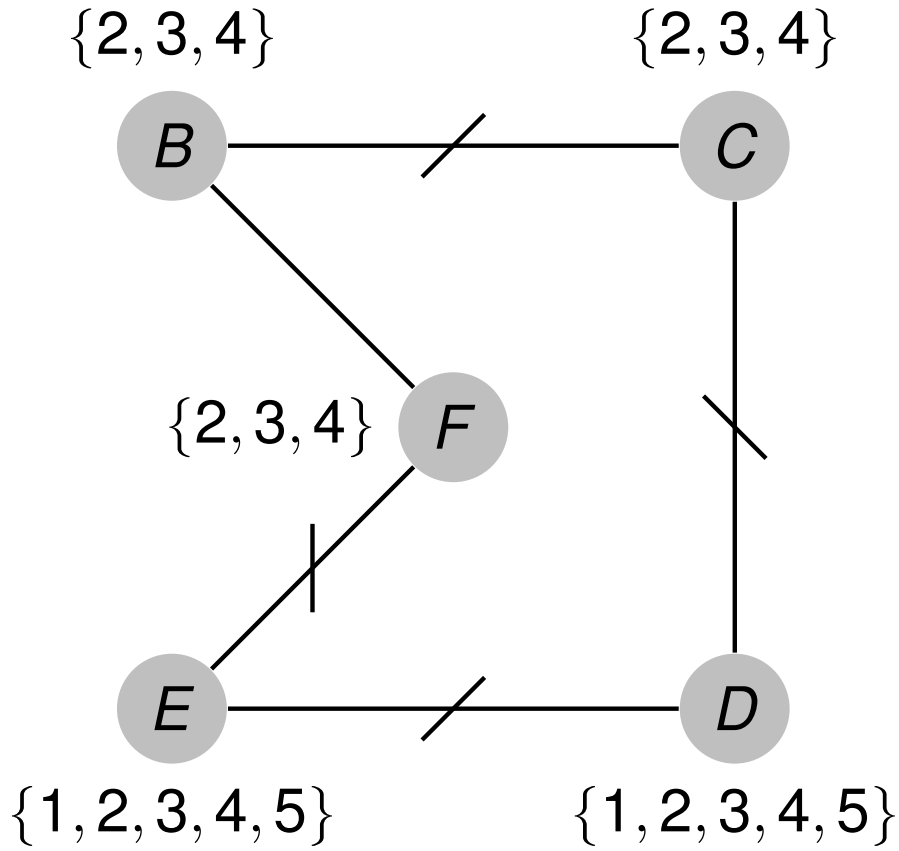
Disposition des invités : Simplification

- Comme la table est circulaire, seule la position relative des invités est importante :
  - On peut donc arbitrairement considérer que $A = 0$, sans changer le fait que le problème ait une solution ou non
  - Il nous reste donc :
    - 5 places à assigner : $1, 2, 3, 4, 5$
    - et les contraintes deviennent :
    - $Dom_B = \{2, 3, 4\}$, $Dom_C = \{2, 3, 4\}$, $Dom_F = \{2, 3, 4\}$,
    - $B + C$, $F + E$, $D + E$, $D + C$ (ne sont pas côte à côte),
    - et $F - B$ (sont côte à côte),
    - de plus toutes les variables doivent prendre des valeurs distinctes dans {1,2,3,4,5}.

Représentation Graphique



La table



Le réseau de contraintes

Résolution de contraintes : exemples d'applications

- Robotique : quels objets doivent être déplacés pour pouvoir sortir une certaine caisse de l'entrepôt ?
- Systèmes tutoriels intelligents : étant donnés les résultats obtenus jusqu'ici, quelle est la prochaine activité la plus adéquate pour un étudiant donné ?
- Analyse de données : comment peut-on regrouper les caractéristiques pour obtenir une représentation plus compacte des données ?
- Jeux : comment offrir de l'aide (indice) dans le cadre d'un jeu ?

Plan

Logique propositionnelle

- Des variables de domaine $\{0, 1\}$ ($0 = \textit{faux}$ et $1 = \textit{vrai}$)
- Les formules sont construites, à partir des variables, à l'aide de la négation $\neg$, la conjonction ("et logique") $\wedge$ et de la disjonction $\vee$.
- Exemples :
  - $\neg q \vee r$
  - $p \wedge (\neg q \vee r)$
- Une formule propositionnelle est donc une contrainte sur ses variables : on cherche des valeurs des variables qui satisfont la formule (la rendent vraie).

Problème SAT

- Trouver une solution pour une formule propositionnelle en *forme normale conjonctive* (FNC) :
  - la formule est donc une conjonction (ET) de *clauses*
  - une *clause* est une disjonction de variables et de négations de variables.

Exemple

- $(p \vee q \vee \neg r) \wedge (\neg p \vee r) \wedge (p \vee \neg q \vee r) \wedge q$ est une FNC
- Une telle formule est normalement représentée de façon suivante

$$\begin{array}{lll} p & q & \bar{r} \\ \bar{p} & r & \\ p & \bar{q} & r \\ q & & \end{array}$$

- Il s'agit donc de chercher à rendre au moins un *littéral* (une variable ou la négation d'une variable) de chaque ligne vrai.
- Évidemment une variable et sa négation doivent prendre des valeurs contraires (une est 0 et l'autre 1).

Propagation d'unités

- Lors de la recherche d'une solution pour le problème SAT, si tous les littéraux d'une clause, sauf un, sont faux, ce dernier littéral doit nécessairement être vrai.
- La procédure qui met à vrai de tels littéraux, la *propagation d'unités*, est donc une méthode d'inférence pour le problème SAT.

Algorithme DPLL

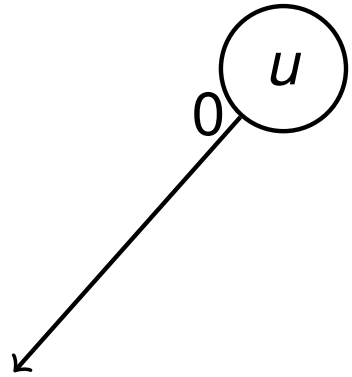
- L'algorithme de Davis, Putnam, Logemann et Loveland procède comme suit :
  - faire la propagation d'unités, si possible
  - si toute les clauses ont au moins un littéral vrai, on a trouvé une solution et la procédure se termine, sinon s'il n'y a pas de *conflit* (une clause dont tous les littéraux sont faux), on choisit une variable et une valeur pour cette variable et on retourne à l'étape précédente
  - s'il y a un conflit, on retourne à la dernière variable choisie pour laquelle il reste une valeur à tenter et on choisit cette nouvelle valeur pour cette variable et on revient à la première étape, sinon on a essayé toutes les possibilités et il n'y a pas de solution.
- C'est un algorithme d'inférence et de recherche pour le problème SAT !

Exemple DPLL

$u \quad v \quad w$
$u \quad w \quad \bar{x}$
$u \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad w$
$u \quad w \quad \bar{x}$
$\bar{v} \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad x$
$\bar{u} \quad v \quad w$
$\bar{u} \quad w \quad \bar{x}$
$\bar{u} \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$

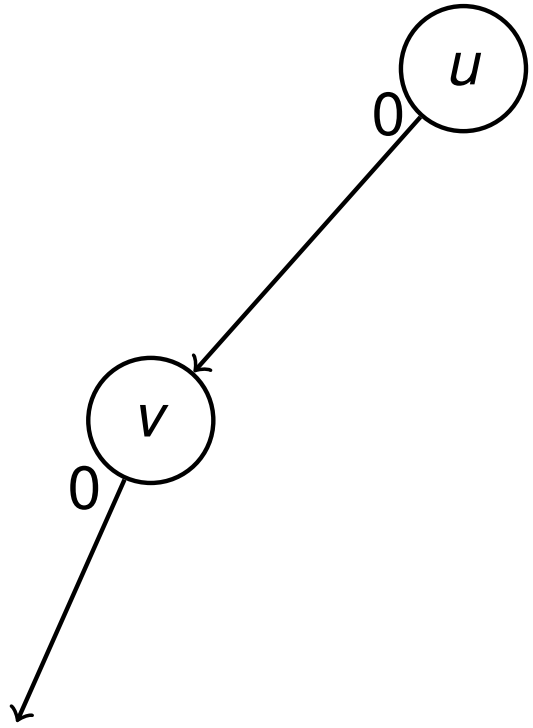
Exemple DPLL

$u \quad v \quad w$
$u \quad w \quad \bar{x}$
$u \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad w$
$u \quad w \quad \bar{x}$
$\bar{v} \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad x$
$\bar{u} \quad v \quad w$
$\bar{u} \quad w \quad \bar{x}$
$\bar{u} \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$



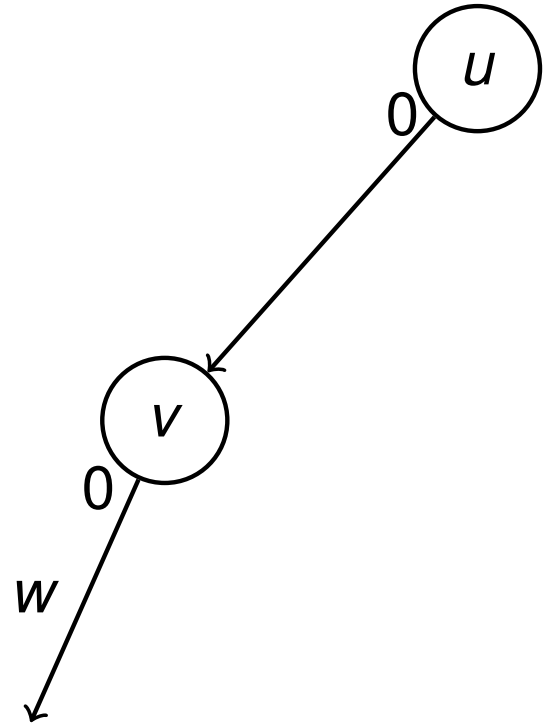
Exemple DPLL

$u \quad v \quad w$
$u \quad w \quad \bar{x}$
$u \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad w$
$u \quad w \quad \bar{x}$
$\bar{v} \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad x$
$\bar{u} \quad v \quad w$
$\bar{u} \quad w \quad \bar{x}$
$\bar{u} \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$



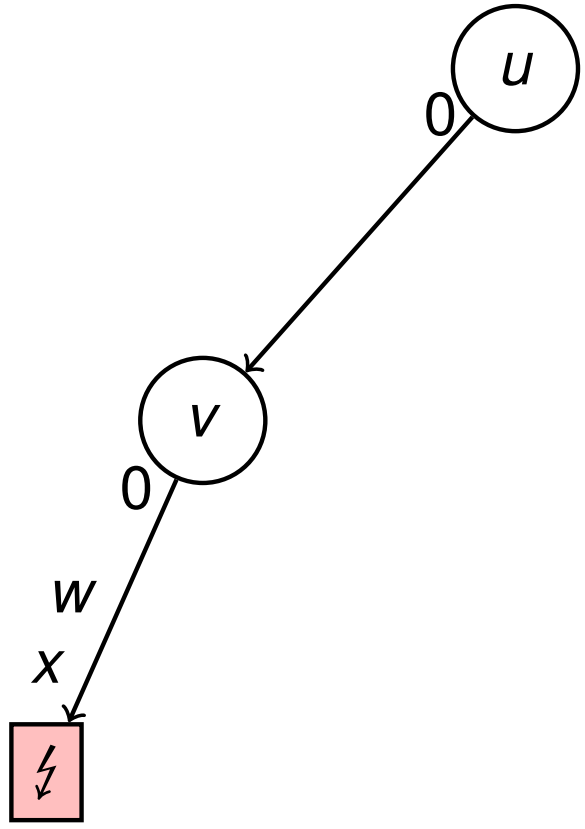
Exemple DPLL

u	v	w
u	w	$\bar{x}$
u	v	x
v	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	w
u	w	$\bar{x}$
$\bar{v}$	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	x
$\bar{u}$	v	w
$\bar{u}$	w	$\bar{x}$
$\bar{u}$	v	x
v	$\bar{w}$	$\bar{x}$



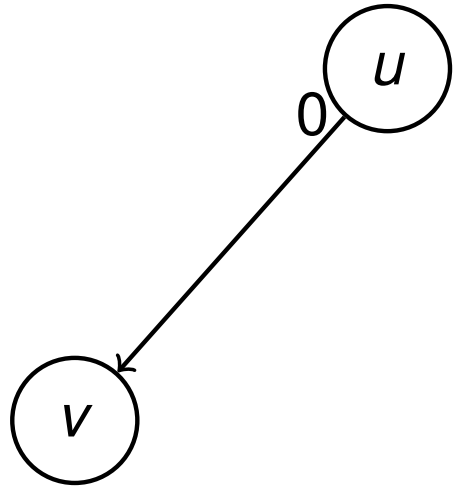
Exemple DPLL

u	v	w
u	w	$\bar{x}$
u	v	x
v	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	w
u	w	$\bar{x}$
$\bar{v}$	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	x
$\bar{u}$	v	w
$\bar{u}$	w	$\bar{x}$
$\bar{u}$	v	x
v	$\bar{w}$	$\bar{x}$



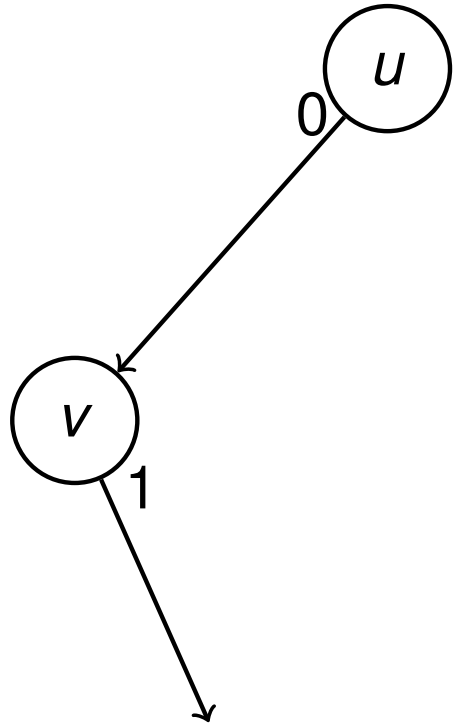
Exemple DPLL

u	v	w
u	w	$\bar{x}$
u	v	x
v	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	w
u	w	$\bar{x}$
$\bar{v}$	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	x
$\bar{u}$	v	w
$\bar{u}$	w	$\bar{x}$
$\bar{u}$	v	x
v	$\bar{w}$	$\bar{x}$



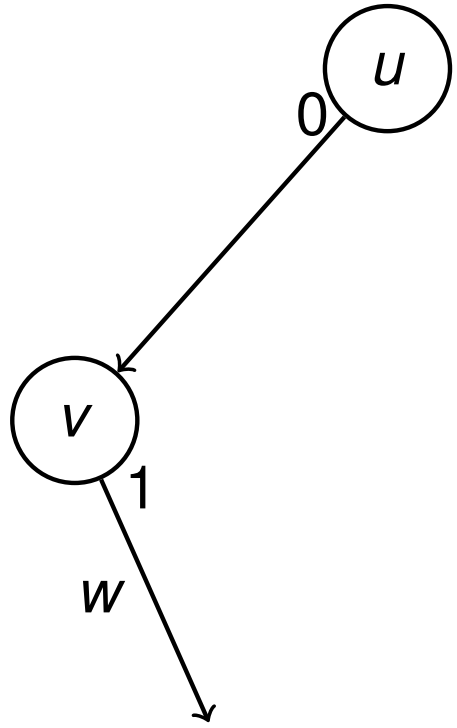
Exemple DPLL

u	v	w
u	w	$\bar{x}$
u	v	x
v	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	w
u	w	$\bar{x}$
$\bar{v}$	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	x
$\bar{u}$	v	w
$\bar{u}$	w	$\bar{x}$
$\bar{u}$	v	x
v	$\bar{w}$	$\bar{x}$



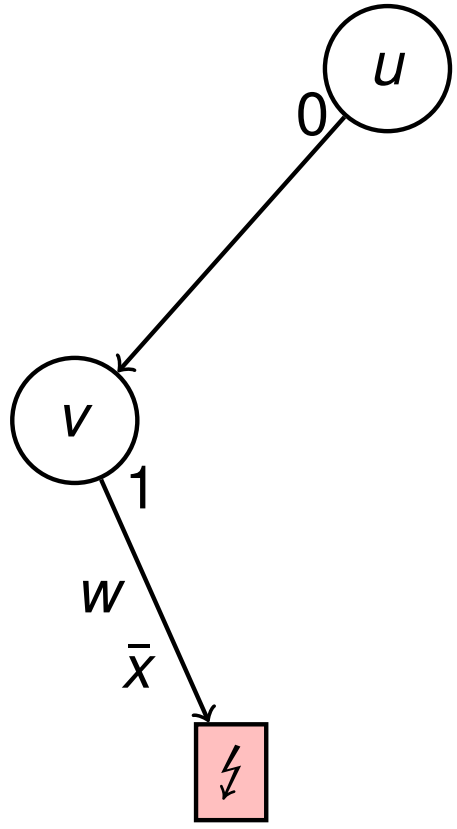
Exemple DPLL

u	v	w
u	w	$\bar{x}$
u	v	x
v	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	w
u	w	$\bar{x}$
$\bar{v}$	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	x
$\bar{u}$	v	w
$\bar{u}$	w	$\bar{x}$
$\bar{u}$	v	x
v	$\bar{w}$	$\bar{x}$



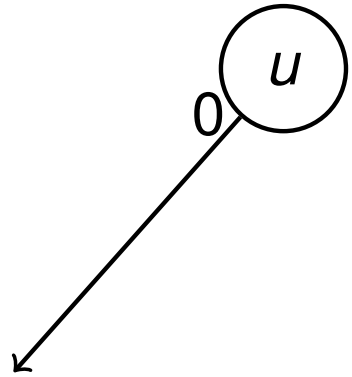
Exemple DPLL

$u \quad v \quad w$
$u \quad w \quad \bar{x}$
$u \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad w$
$u \quad w \quad \bar{x}$
$\bar{v} \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad x$
$\bar{u} \quad v \quad w$
$\bar{u} \quad w \quad \bar{x}$
$\bar{u} \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$



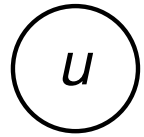
Exemple DPLL

$u \quad v \quad w$
$u \quad w \quad \bar{x}$
$u \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad w$
$u \quad w \quad \bar{x}$
$\bar{v} \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad x$
$\bar{u} \quad v \quad w$
$\bar{u} \quad w \quad \bar{x}$
$\bar{u} \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$



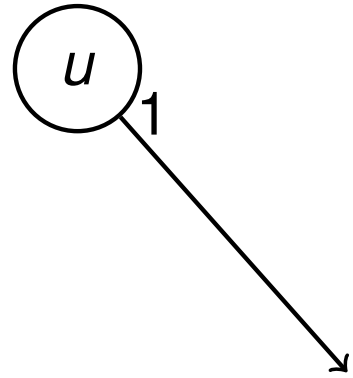
Exemple DPLL

u v w
u w $\bar{x}$
u v x
v $\bar{w}$ $\bar{x}$
u $\bar{v}$ w
u w $\bar{x}$
$\bar{v}$ $\bar{w}$ $\bar{x}$
u $\bar{v}$ x
$\bar{u}$ v w
$\bar{u}$ w $\bar{x}$
$\bar{u}$ v x
v $\bar{w}$ $\bar{x}$



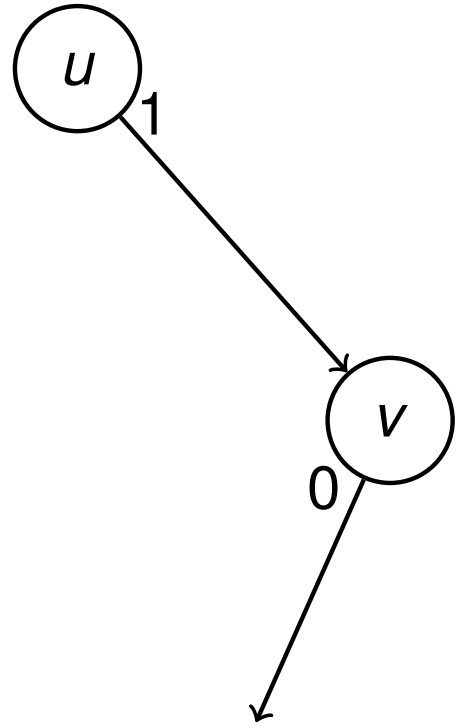
Exemple DPLL

$u \quad v \quad w$
$u \quad w \quad \bar{x}$
$u \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad w$
$u \quad w \quad \bar{x}$
$\bar{v} \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad x$
$\bar{u} \quad v \quad w$
$\bar{u} \quad w \quad \bar{x}$
$\bar{u} \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$



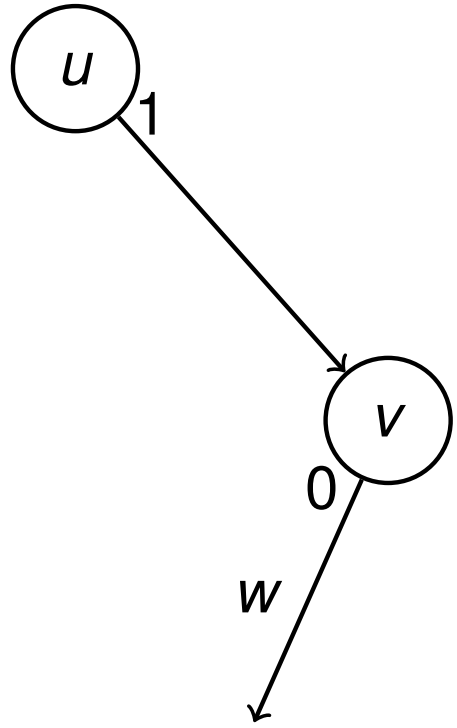
Exemple DPLL

u	v	w
u	w	$\bar{x}$
u	v	x
v	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	w
u	w	$\bar{x}$
$\bar{v}$	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	x
$\bar{u}$	v	w
$\bar{u}$	w	$\bar{x}$
$\bar{u}$	v	x
v	$\bar{w}$	$\bar{x}$



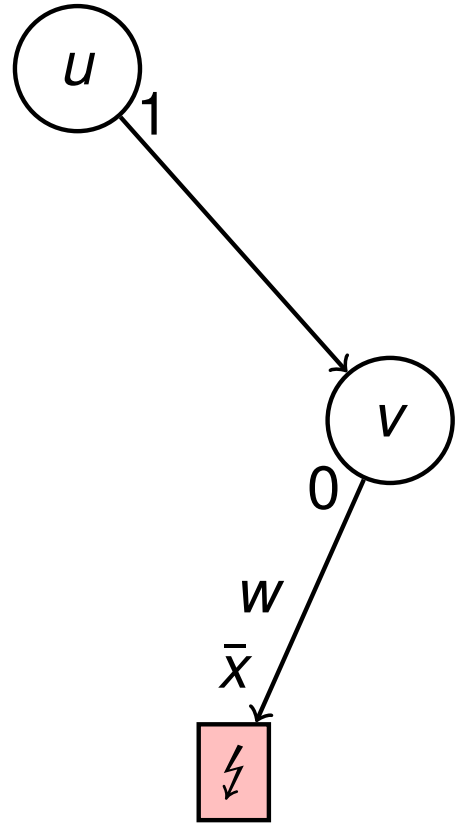
Exemple DPLL

$u \quad v \quad w$
$u \quad w \quad \bar{x}$
$u \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad w$
$u \quad w \quad \bar{x}$
$\bar{v} \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad x$
$\bar{u} \quad v \quad w$
$\bar{u} \quad w \quad \bar{x}$
$\bar{u} \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$



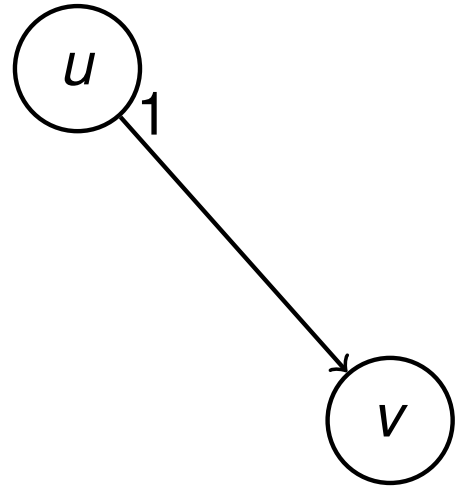
Exemple DPLL

u	v	w
u	w	$\bar{x}$
u	v	x
v	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	w
u	w	$\bar{x}$
$\bar{v}$	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	x
$\bar{u}$	v	w
$\bar{u}$	w	$\bar{x}$
$\bar{u}$	v	x
v	$\bar{w}$	$\bar{x}$



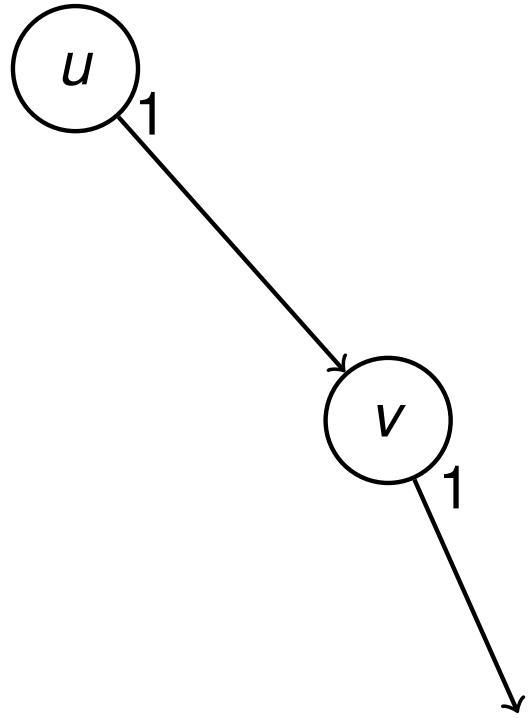
Exemple DPLL

$u \quad v \quad w$
$u \quad w \quad \bar{x}$
$u \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad w$
$u \quad w \quad \bar{x}$
$\bar{v} \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad x$
$\bar{u} \quad v \quad w$
$\bar{u} \quad w \quad \bar{x}$
$\bar{u} \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$



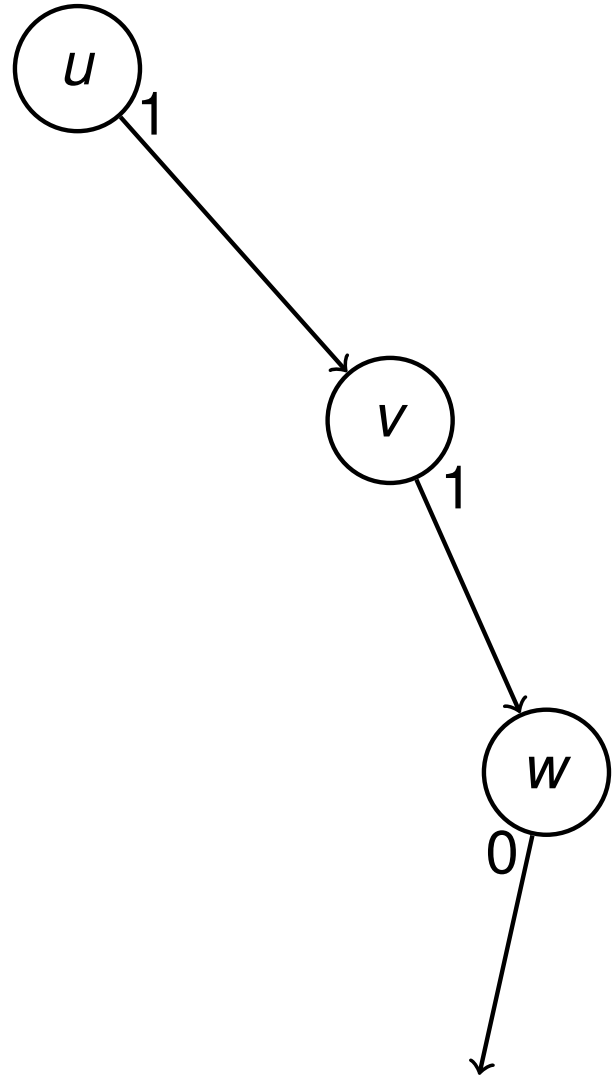
Exemple DPLL

$u \quad v \quad w$
$u \quad w \quad \bar{x}$
$u \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad w$
$u \quad w \quad \bar{x}$
$\bar{v} \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad x$
$\bar{u} \quad v \quad w$
$\bar{u} \quad w \quad \bar{x}$
$\bar{u} \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$



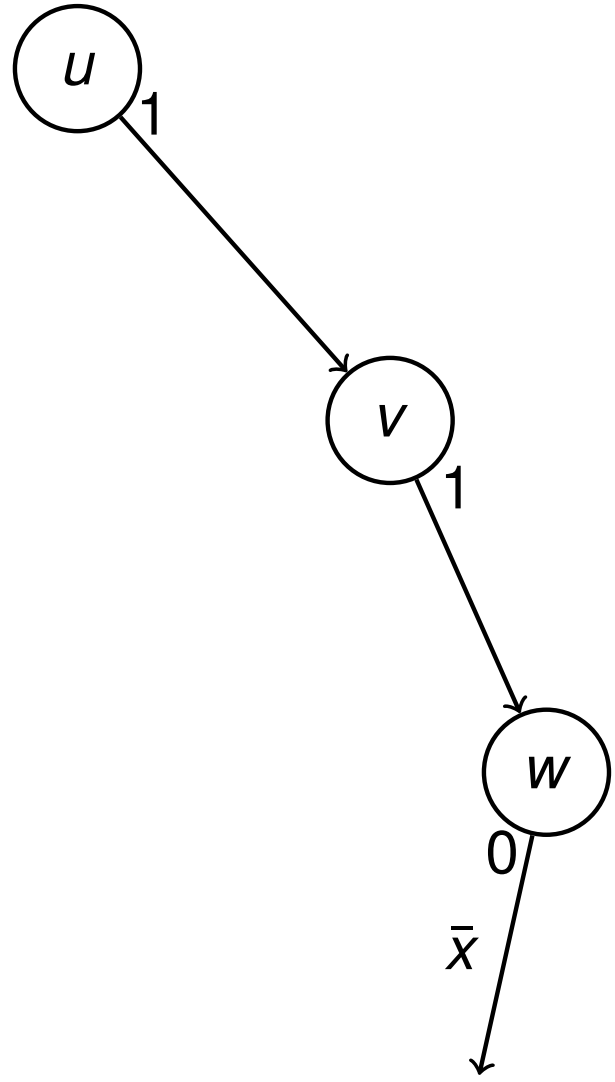
Exemple DPLL

$u \quad v \quad w$
$u \quad w \quad \bar{x}$
$u \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad w$
$u \quad w \quad \bar{x}$
$\bar{v} \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad x$
$\bar{u} \quad v \quad w$
$\bar{u} \quad w \quad \bar{x}$
$\bar{u} \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$



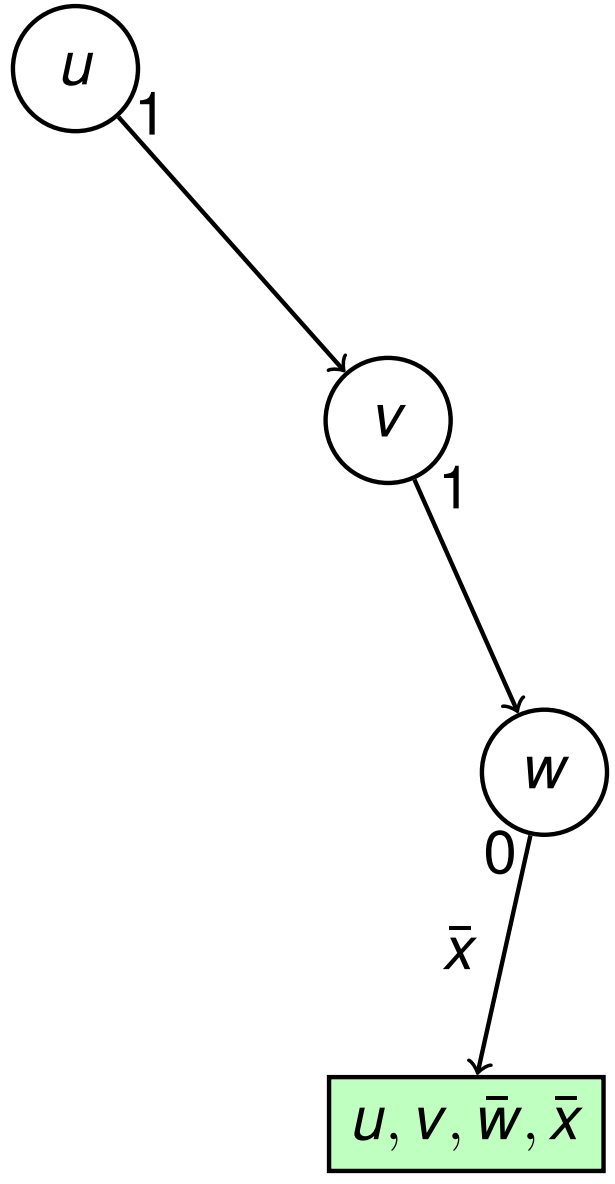
Exemple DPLL

u	v	w
u	w	$\bar{x}$
u	v	x
v	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	w
u	w	$\bar{x}$
$\bar{v}$	$\bar{w}$	$\bar{x}$
u	$\bar{v}$	x
$\bar{u}$	v	w
$\bar{u}$	w	$\bar{x}$
$\bar{u}$	v	x
v	$\bar{w}$	$\bar{x}$



Exemple DPLL

$u \quad v \quad w$
$u \quad w \quad \bar{x}$
$u \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad w$
$u \quad w \quad \bar{x}$
$\bar{v} \quad \bar{w} \quad \bar{x}$
$u \quad \bar{v} \quad x$
$\bar{u} \quad v \quad w$
$\bar{u} \quad w \quad \bar{x}$
$\bar{u} \quad v \quad x$
$v \quad \bar{w} \quad \bar{x}$



Remarques

- La propagation d'unités de l'algorithme DPLL permet d'éviter de devoir tenter toutes les valeurs possibles pour toutes les variables, ce qui serait considérablement plus long.
- Les solveurs SAT modernes utilisent néanmoins des méthodes plus sophistiquées, comme par exemple l'algorithme CDCL (Conflit Directed Clauses Learning), qui permet encore plus de réduire le nombre d'assignations de valeurs aux variables.
- En pratique de tels *solveurs* sont très efficaces et largement utilisés car tout problème fini, et certains infinis, peut être réduit à SAT.
- Ceci même si tous les algorithmes connus sont de complexité exponentielle (problème $P = NP$).

Conclusion

- La méthode *inférence et recherche* est inspirée des méthodes utilisées par les êtres humains.
- Il reste que les méthodes d'inférence utilisées peuvent être moins sophistiquées que celles des êtres humains.
- Mais la force brute de la machine permet d'effectuer beaucoup plus de recherche qu'un être humain ne pourrait le faire !
- Ceci est particulièrement utile lorsqu'il n'est pas si simple d'identifier les inférences véritablement faites par des êtres humains.
- Ultimement ce qui compte est d'obtenir un résultat pertinent dans un temps raisonnable !