Chapitre 6
Modélisation en logique propositionnelle

Table de Matières

Introduction

Exemple 1 : Colorier une carte

Exemple 2 : Des mariages heureux

Contraintes de comptage

Remarques finales

Résumé du cours précedent

Formule en	DNF	CNF
Satisfaisabilité :	trivial	DPLL
Validité :	DPLL dual	trivial

Modéliser et résoudre

- Outils efficaces (en pratique) pour trouver des solutions des formules en forme CNF : DPLL, et plus récemment les SAT-solveurs.

Modéliser et résoudre

- Outils efficaces (en pratique) pour trouver des solutions des formules en forme CNF : DPLL, et plus récemment les SAT-solveurs.
- Étant donné un problème combinatoire, comment construire la formule correspondante en forme CNF ?

Modéliser et résoudre

- Outils efficaces (en pratique) pour trouver des solutions des formules en forme CNF : DPLL, et plus récemment les SAT-solveurs.
- Étant donné un problème combinatoire, comment construire la formule correspondante en forme CNF ?
- Comment travailler avec un SAT-solveur ? (ne sera pas traité)

Pourquoi construire la formule en CNF

. . . et pas en DNF ?

Pourquoi construire la formule en CNF

. . . et pas en DNF ?

- Pour beaucoup d'applications c'est plus naturel :

Pourquoi construire la formule en CNF

... et pas en DNF ?

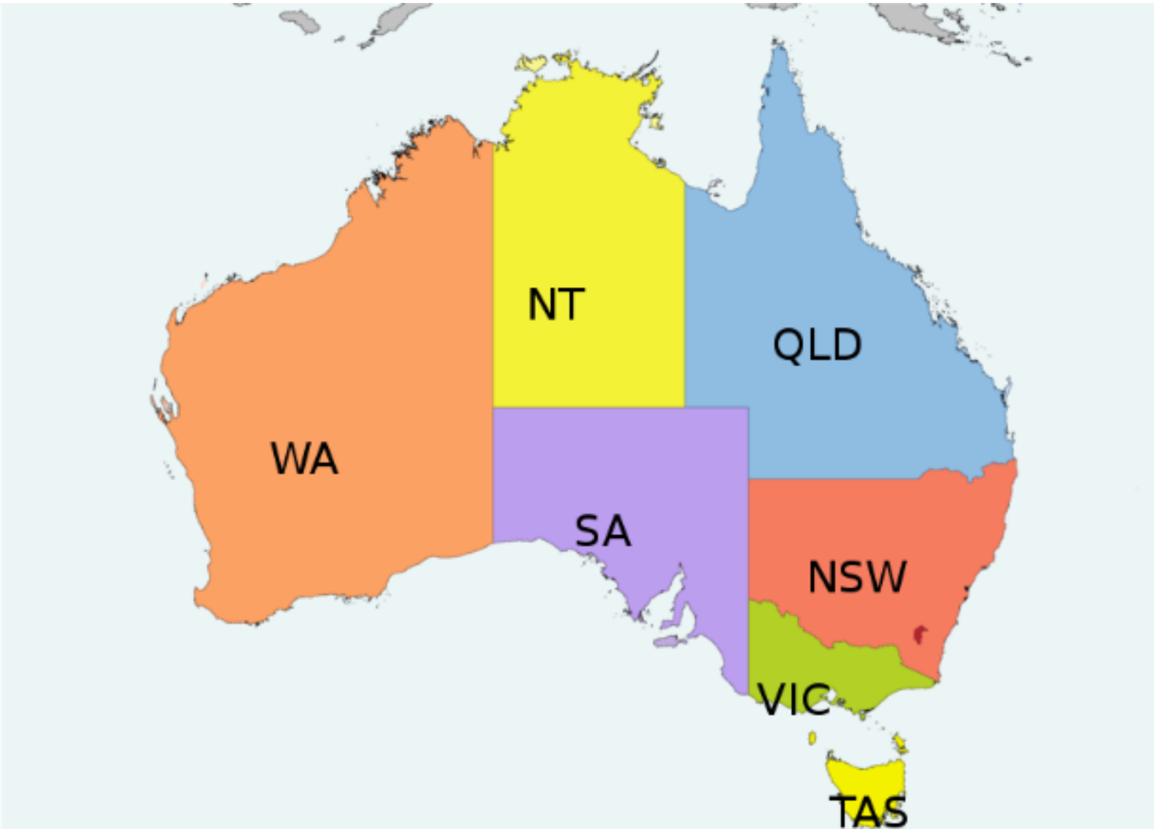
- Pour beaucoup d'applications c'est plus naturel :
- On a plusieurs contraintes à respecter (donc une conjonction)

Pourquoi construire la formule en CNF

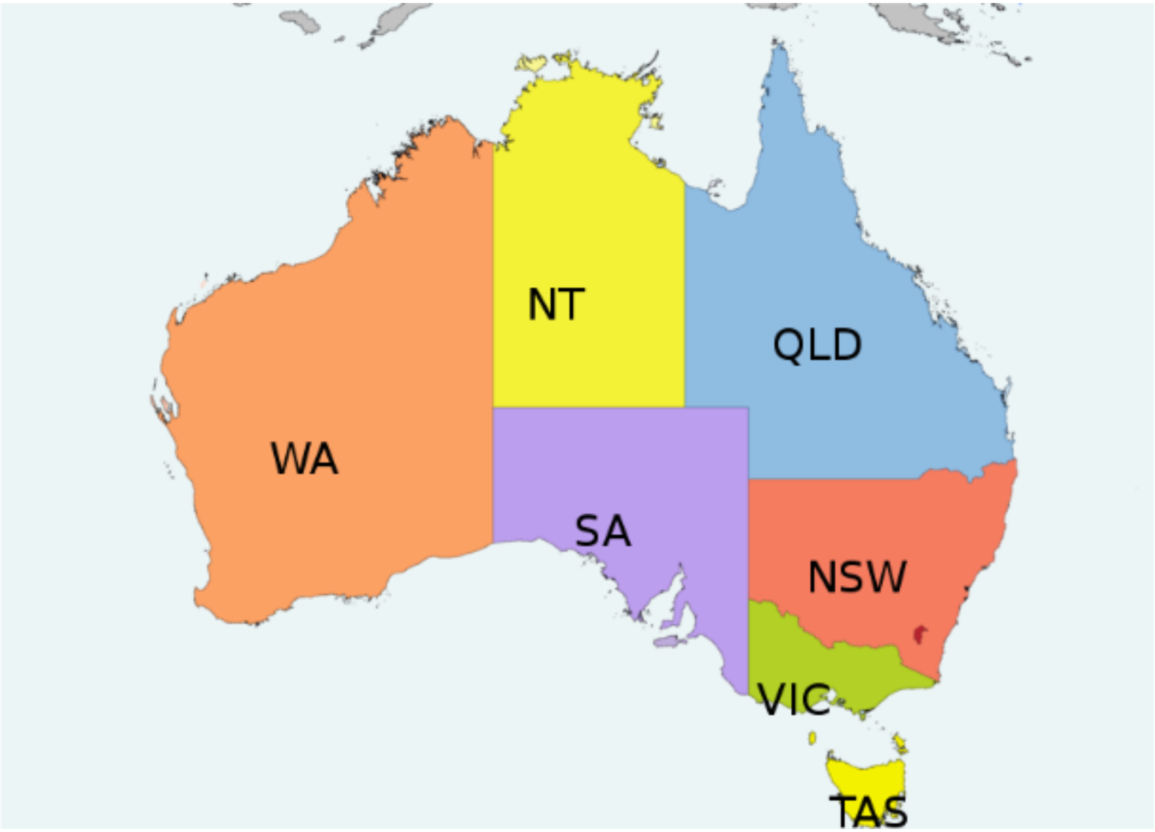
... et pas en DNF ?

- Pour beaucoup d'applications c'est plus naturel :
- On a plusieurs contraintes à respecter (donc une conjonction)
- Chaque contrainte représente un choix entre plusieurs alternatives (donc une disjonction)

Colorier la carte de l'Australie



Colorier la carte de l'Australie



- Colorier cette carte avec 3 couleurs (rouge,bleu,vert) seulement !

Coloration d'une carte

- Donnée une carte avec plusieurs régions, et n couleurs

Coloration d'une carte

- Donnée une carte avec plusieurs régions, et n couleurs
- Associer à chaque région une couleur, . . .

Coloration d'une carte

- Donnée une carte avec plusieurs régions, et n couleurs
- Associer à chaque région une couleur, ...
- ... telle que deux régions adjacentes (ayant un segment de frontière commun, pas seulement un point commun) ne portent pas la même couleur.

Coloration d'une carte

- Donnée une carte avec plusieurs régions, et n couleurs
- Associer à chaque région une couleur, ...
- ... telle que deux régions adjacentes (ayant un segment de frontière commun, pas seulement un point commun) ne portent pas la même couleur.
- Avec 4 couleurs c'est toujours possible (pour une carte planaire, ou un globe) : C'est le célèbre *théorème des quatre couleurs*.

Formulation mathématique du problème

On cherche une fonction

$$couleur\colon \{WA, NT, SA, QLD, NSW, VIC, TAS\} \to \{r, b, v\}$$

telle que

$$couleur(x) \neq couleur(y) \quad \text{si } x \text{ et } y \text{ sont adjacents}$$

Vers une modélisation

On coupe la construction de la formule en deux parties :

Vers une modélisation

On coupe la construction de la formule en deux parties :

1. Modélisation des fonctions

$$couleur\colon \{WA, NT, SA, QLD, NSW, VIC, TAS\} \to \{r, b, v\}$$

Vers une modélisation

On coupe la construction de la formule en deux parties :

1. Modélisation des fonctions

$$couleur : \{WA, NT, SA, QLD, NSW, VIC, TAS\} \rightarrow \{r, b, v\}$$

2. Modélisation de la contrainte :

$$couleur(x) \neq couleur(y) \quad \text{si } x \text{ et } y \text{ sont adjacents}$$

Modélisation des colorations

- Si on avait seulement deux couleurs (blanc et noir, par exemple) :

Modélisation des colorations

- Si on avait seulement deux couleurs (blanc et noir, par exemple) :
- On pourrait définir une variable propositionnelle par région, avec la convention que 0 représente noir, et 1 représente blanc.

Modélisation des colorations

- Si on avait seulement deux couleurs (blanc et noir, par exemple) :
- On pourrait définir une variable propositionnelle par région, avec la convention que 0 représente noir, et 1 représente blanc.
- Dans ce cas on aurait une correspondance parfaite entre affectations et colorations : toute affectation des variables correspond à une coloration de la carte, et inversement.

Modélisation des colorations

- Comment faire avec n couleurs ($n \geq 3$) ?

Modélisation des colorations

- Comment faire avec n couleurs ($n \geq 3$) ?
- La solution la plus simple : choisir une variable `[x,y]` pour toute région x et toute couleur y.

Modélisation des colorations

- Comment faire avec n couleurs ($n \geq 3$) ?
- La solution la plus simple : choisir une variable `[x,y]` pour toute région x et toute couleur y.
- Donc, pour 7 régions et 3 couleurs on a 21 variables :

$$\texttt{[WA,r]}, \texttt{[WA,b]}, \texttt{[WA,v]}, \texttt{[NT,r]}, \ldots, \texttt{[TAS, v]}$$

Modélisation des colorations

Pour toute coloration c de la carte il y a une affectation α qui lui correspond :

$$\alpha([x,y]) = \begin{cases} 1 & \text{si } c(x) = y \\ 0 & \text{si } c(x) \neq y \end{cases}$$

Modélisation des colorations

Problème : L'inverse n'est pas vrai. Il y a des affectations des variables qui ne correspondent pas à une coloration !

Modélisation des colorations

Problème : L'inverse n'est pas vrai. Il y a des affectations des variables qui ne correspondent pas à une coloration !

- Si
$$\alpha([WA, r]) = 0, \alpha([WA, b]) = 0, \alpha([WA, v]) = 0$$
alors on aurait aucune couleur pour WA.

Modélisation des colorations

Problème : L'inverse n'est pas vrai. Il y a des affectations des variables qui ne correspondent pas à une coloration !

- Si
$$\alpha([WA, r]) = 0, \alpha([WA, b]) = 0, \alpha([WA, v]) = 0$$
alors on aurait aucune couleur pour WA.
- Si
$$\alpha([WA, r]) = 1, \alpha([WA, b]) = 1$$
alors on aurait plusieurs couleurs pour WA.

Modélisation des colorations

Exprimer que toute région a au moins une couleur :

$$
\begin{aligned}
P_1 := \quad & (\texttt{[WA,r]} \vee \texttt{[WA,b]} \vee \texttt{[WA,v]}) \\
\wedge \quad & (\texttt{[NT,r]} \vee \texttt{[NT,b]} \vee \texttt{[NT,v]}) \\
\wedge \quad & (\texttt{[SA,r]} \vee \texttt{[SA,b]} \vee \texttt{[SA,v]}) \\
\wedge \quad & (\texttt{[QLD,r]} \vee \texttt{[QLD,b]} \vee \texttt{[QLD,v]}) \\
\wedge \quad & (\texttt{[NSW,r]} \vee \texttt{[NSW,b]} \vee \texttt{[NSW,v]}) \\
\wedge \quad & (\texttt{[VIC,r]} \vee \texttt{[VIC,b]} \vee \texttt{[VIC,v]}) \\
\wedge \quad & (\texttt{[TAS,r]} \vee \texttt{[TAS,b]} \vee \texttt{[TAS,v]})
\end{aligned}
$$

Modélisation des colorations

Exprimer que toute région a au plus une couleur :

$$
\begin{array}{lllll}
P_2 & := & & & \\
 & (\neg\texttt{[WA,r]} \vee \neg\texttt{[WA,b]}) & \wedge & (\neg\texttt{[WA,r]} \vee \neg\texttt{[WA,v]}) & \wedge \quad (\neg\texttt{[WA,b]} \vee \neg\texttt{[WA,v]}) \\
\wedge & (\neg\texttt{[NT,r]} \vee \neg\texttt{[NT,b]}) & \wedge & (\neg\texttt{[NT,r]} \vee \neg\texttt{[NT,v]}) & \wedge \quad (\neg\texttt{[NT,b]} \vee \neg\texttt{[NT,v]}) \\
\wedge & (\neg\texttt{[SA,r]} \vee \neg\texttt{[SA,b]}) & \wedge & (\neg\texttt{[SA,r]} \vee \neg\texttt{[SA,v]}) & \wedge \quad (\neg\texttt{[SA,b]} \vee \neg\texttt{[SA,v]}) \\
\wedge & (\neg\texttt{[QLD,r]} \vee \neg\texttt{[QLD,b]}) & \wedge & (\neg\texttt{[QLD,r]} \vee \neg\texttt{[QLD,v]}) & \wedge \quad (\neg\texttt{[QLD,b]} \vee \neg\texttt{[QLD,v]}) \\
\wedge & (\neg\texttt{[NSW,r]} \vee \neg\texttt{[NSW,b]}) & \wedge & (\neg\texttt{[NSW,r]} \vee \neg\texttt{[NSW,v]}) & \wedge \quad (\neg\texttt{[NSW,b]} \vee \neg\texttt{[NSW,v]}) \\
\wedge & (\neg\texttt{[VIC,r]} \vee \neg\texttt{[VIC,b]}) & \wedge & (\neg\texttt{[VIC,r]} \vee \neg\texttt{[VIC,v]}) & \wedge \quad (\neg\texttt{[VIC,b]} \vee \neg\texttt{[VIC,v]}) \\
\wedge & (\neg\texttt{[TAS,r]} \vee \neg\texttt{[TAS,b]}) & \wedge & (\neg\texttt{[TAS,r]} \vee \neg\texttt{[TAS,v]}) & \wedge \quad (\neg\texttt{[TAS,b]} \vee \neg\texttt{[TAS,v]})
\end{array}
$$

Modéliser la contrainte spécifique

Exclure toutes les affectations qui donnent la même couleur à deux régions adjacentes :

$P_3 :=$

$$
\begin{array}{llllll}
 & (\neg\texttt{[WA,r]} \vee \neg\texttt{[NT,r]}) & \wedge & (\neg\texttt{[WA,b]} \vee \neg\texttt{[NT,b]}) & \wedge & (\neg\texttt{[WA,v]} \vee \neg\texttt{[NT,v]}) \\
\wedge & (\neg\texttt{[WA,r]} \vee \neg\texttt{[SA,r]}) & \wedge & (\neg\texttt{[WA,b]} \vee \neg\texttt{[SA,b]}) & \wedge & (\neg\texttt{[WA,v]} \vee \neg\texttt{[SA,v]}) \\
\wedge & (\neg\texttt{[NT,r]} \vee \neg\texttt{[SA,r]}) & \wedge & (\neg\texttt{[NT,b]} \vee \neg\texttt{[SA,b]}) & \wedge & (\neg\texttt{[NT,v]} \vee \neg\texttt{[SA,v]}) \\
\wedge & (\neg\texttt{[NT,r]} \vee \neg\texttt{[QLD,r]}) & \wedge & (\neg\texttt{[NT,b]} \vee \neg\texttt{[QLD,b]}) & \wedge & (\neg\texttt{[NT,v]} \vee \neg\texttt{[QLD,v]}) \\
\wedge & (\neg\texttt{[SA,r]} \vee \neg\texttt{[QLD,r]}) & \wedge & (\neg\texttt{[SA,b]} \vee \neg\texttt{[QLD,b]}) & \wedge & (\neg\texttt{[SA,v]} \vee \neg\texttt{[QLD,v]}) \\
\wedge & (\neg\texttt{[SA,r]} \vee \neg\texttt{[NSW,r]}) & \wedge & (\neg\texttt{[SA,b]} \vee \neg\texttt{[NSW,b]}) & \wedge & (\neg\texttt{[SA,v]} \vee \neg\texttt{[NSW,v]}) \\
\wedge & (\neg\texttt{[SA,r]} \vee \neg\texttt{[VIC,r]}) & \wedge & (\neg\texttt{[SA,b]} \vee \neg\texttt{[VIC,b]}) & \wedge & (\neg\texttt{[SA,v]} \vee \neg\texttt{[VIC,v]})
\end{array}
$$

Modéliser la contrainte spécifique

Exclure toutes les affectations qui donnent la même couleur à deux régions adjacentes :

$$
\begin{array}{llllll}
P_3 := & & & & & \\
 & (\neg\texttt{[WA,r]} \lor \neg\texttt{[NT,r]}) & \land & (\neg\texttt{[WA,b]} \lor \neg\texttt{[NT,b]}) & \land & (\neg\texttt{[WA,v]} \lor \neg\texttt{[NT,v]}) \\
\land & (\neg\texttt{[WA,r]} \lor \neg\texttt{[SA,r]}) & \land & (\neg\texttt{[WA,b]} \lor \neg\texttt{[SA,b]}) & \land & (\neg\texttt{[WA,v]} \lor \neg\texttt{[SA,v]}) \\
\land & (\neg\texttt{[NT,r]} \lor \neg\texttt{[SA,r]}) & \land & (\neg\texttt{[NT,b]} \lor \neg\texttt{[SA,b]}) & \land & (\neg\texttt{[NT,v]} \lor \neg\texttt{[SA,v]}) \\
\land & (\neg\texttt{[NT,r]} \lor \neg\texttt{[QLD,r]}) & \land & (\neg\texttt{[NT,b]} \lor \neg\texttt{[QLD,b]}) & \land & (\neg\texttt{[NT,v]} \lor \neg\texttt{[QLD,v]}) \\
\land & (\neg\texttt{[SA,r]} \lor \neg\texttt{[QLD,r]}) & \land & (\neg\texttt{[SA,b]} \lor \neg\texttt{[QLD,b]}) & \land & (\neg\texttt{[SA,v]} \lor \neg\texttt{[QLD,v]}) \\
\land & (\neg\texttt{[SA,r]} \lor \neg\texttt{[NSW,r]}) & \land & (\neg\texttt{[SA,b]} \lor \neg\texttt{[NSW,b]}) & \land & (\neg\texttt{[SA,v]} \lor \neg\texttt{[NSW,v]}) \\
\land & (\neg\texttt{[SA,r]} \lor \neg\texttt{[VIC,r]}) & \land & (\neg\texttt{[SA,b]} \lor \neg\texttt{[VIC,b]}) & \land & (\neg\texttt{[SA,v]} \lor \neg\texttt{[VIC,v]})
\end{array}
$$

La Tasmanie n'y figure pas car c'est une île.

- La formule complète : $P_1 \wedge P_2 \wedge P_3$, elle est déjà en forme CNF.

- La formule complète : $P_1 \wedge P_2 \wedge P_3$, elle est déjà en forme CNF.
- La première étape (modélisation des colorations) était indépendante de la carte spécifique (à part du nombre de pays).

- La formule complète : $P_1 \wedge P_2 \wedge P_3$, elle est déjà en forme CNF.
- La première étape (modélisation des colorations) était indépendante de la carte spécifique (à part du nombre de pays).
- La deuxième étape était spécifique à la carte donnée, même si le *principe* de construction s'applique à n'importe quelle carte.

Généralisation

Donnés :

- un ensemble de pays $1, \ldots, n$
- un ensemble de couleurs $1, \ldots, m$
- une fonction $A \colon \{1, \ldots, n\} \times \{1, \ldots, n\} \to \{true, false\}$ qui indique si deux pays sont adjacents.

Modélisation dans le cas général

- L'ensemble des variables propositionnelles est

$$\{[i,j] \mid 1 \leq i \leq n, 1 \leq j \leq m\}$$

Modélisation dans le cas général

- L'ensemble des variables propositionnelles est

$$\{[i,j] \mid 1 \leq i \leq n, 1 \leq j \leq m\}$$

- $$P_1 = \bigwedge_{1 \leq i \leq n} \Big(\bigvee_{1 \leq j \leq m} [i,j] \Big)$$

Modélisation dans le cas général

- L'ensemble des variables propositionnelles est

$$\{[i,j] \mid 1 \leq i \leq n, 1 \leq j \leq m\}$$

- $$P_1 = \bigwedge_{1\leq i\leq n} \Big(\bigvee_{1\leq j\leq m} [i,j] \Big)$$

- $$P_2 = \bigwedge_{1\leq i\leq n} \bigwedge_{1\leq j\leq m} \bigwedge_{j<k\leq m} (\neg[i,j] \vee \neg[i,k])$$

Modélisation dans le cas général

- L'ensemble des variables propositionnelles est

$$\{[i,j] \mid 1 \leq i \leq n, 1 \leq j \leq m\}$$

- $$P_1 = \bigwedge_{1\leq i\leq n} \Big(\bigvee_{1\leq j\leq m} [i,j] \Big)$$

- $$P_2 = \bigwedge_{1\leq i\leq n} \bigwedge_{1\leq j\leq m} \bigwedge_{j<k\leq m} (\neg[i,j] \vee \neg[i,k])$$

- $$P_3 = \bigwedge_{1\leq i\leq n} \bigwedge_{\substack{i<j\leq n \\ A(i,j)=true}} \bigwedge_{1\leq k\leq m} (\neg[i,k] \vee \neg[j,k])$$

Le programme complet

On peut maintenant facilement écrire un programme qui

1. lit la carte (dans une représentation appropriée) et le nombre de couleurs

Le programme complet

On peut maintenant facilement écrire un programme qui

1. lit la carte (dans une représentation appropriée) et le nombre de couleurs
2. construit la formule propositionnelle comme décrit au-dessus

Le programme complet

On peut maintenant facilement écrire un programme qui

1. lit la carte (dans une représentation appropriée) et le nombre de couleurs
2. construit la formule propositionnelle comme décrit au-dessus
3. lance un SAT-solveur externe sur cette formule

Le programme complet

On peut maintenant facilement écrire un programme qui

1. lit la carte (dans une représentation appropriée) et le nombre de couleurs
2. construit la formule propositionnelle comme décrit au-dessus
3. lance un SAT-solveur externe sur cette formule
4. analyse l'affectation trouvée par le SAT-solveur (quand une solution existe) pour afficher une coloration de la carte.

Comportement d'un SAT-solveur pour ce problème

- Initialement il n'y a ni de clause unitaire, ni de variable en une seule polarité.

Comportement d'un SAT-solveur pour ce problème

- Initialement il n'y a ni de clause unitaire, ni de variable en une seule polarité.
- On choisi donc une variable, disons `[WA,r]`, et essaye de la mettre à la valeur 1, sinon à 0.

Comportement d'un SAT-solveur pour ce problème

- Initialement il n'y a ni de clause unitaire, ni de variable en une seule polarité.
- On choisi donc une variable, disons `[WA,r]`, et essaye de la mettre à la valeur 1, sinon à 0.
- Quand la variable `[WA,r]` est mise à 1, toutes les clauses binaires qui contiennent le littéral négatif $\neg$`[WA,r]` vont se transformer en clause unitaire, et donc déclencher un grand nombre de résolutions unitaires.

Comportement d'un SAT-solveur pour ce problème

- Initialement il n'y a ni de clause unitaire, ni de variable en une seule polarité.
- On choisi donc une variable, disons `[WA,r]`, et essaye de la mettre à la valeur 1, sinon à 0.
- Quand la variable `[WA,r]` est mise à 1, toutes les clauses binaires qui contiennent le littéral négatif $\neg$`[WA,r]` vont se transformer en clause unitaire, et donc déclencher un grand nombre de résolutions unitaires.
- En particulier, le choix de la couleur rouge va être enlevé pour tous les pays qui sont adjacents avec WA.

Le problème

- Trois étudiants : Adam, Bruno, Chen.
- Trois étudiantes : Xanthia, Ylenia, Zoé.

Le problème

- Trois étudiants : Adam, Bruno, Chen.
- Trois étudiantes : Xanthia, Ylenia, Zoé.
- Adam aime Xanthia et Zoé, Bruno aime Ylenia, Chen aime Ylenia et Xanthia, tandis que Xanthia aime Bruno et Chen, Ylenia aime Adam et Bruno, Zoé aime Adam et Chen.

Le problème

- Trois étudiants : Adam, Bruno, Chen.
- Trois étudiantes : Xanthia, Ylenia, Zoé.
- Adam aime Xanthia et Zoé, Bruno aime Ylenia, Chen aime Ylenia et Xanthia, tandis que Xanthia aime Bruno et Chen, Ylenia aime Adam et Bruno, Zoé aime Adam et Chen.
- Est-il possible de faire trois mariages parmi eux, tel que chacun(e) se marie avec une personne qu'il (qu'elle) aime ?

Comment définir les variables propositionnelles?

- Tentative : variable $\langle X, Y \rangle$, où X et Y sont des personnes, qui dénotent le fait que X aime Y.

Comment définir les variables propositionnelles?

- Tentative : variable $\langle X, Y \rangle$, où X et Y sont des personnes, qui dénotent le fait que X aime Y.
- Ça serait une mauvaise approche. Pourquoi ?

Comment définir les variables propositionnelles?

- Tentative : variable $\langle X, Y \rangle$, où X et Y sont des personnes, qui dénotent le fait que X aime Y.
- Ça serait une mauvaise approche. Pourquoi ?
- Les (affectations à des) variables doivent modéliser des solutions, la formule doit modéliser le problème !

Comment définir les variables propositionnelles?

- Variables $[X, Y]$ pour toutes personnes X, Y, exprimant le fait que X se marie avec Y ($6 * 6 = 36$ variables).

Comment définir les variables propositionnelles?

- Variables $[X, Y]$ pour toutes personnes X, Y, exprimant le fait que X se marie avec Y ($6 * 6 = 36$ variables).
- Cela nous obligera d'exprimer par une formule le fait que chaque mariage est entre un garçon et une fille, et en plus qu'un mariage entre X et Y est la même chose qu'un mariage entre Y et X.

Comment définir les variables propositionnelles?

- Variables $[X, Y]$ pour toutes personnes X, Y, exprimant le fait que X se marie avec Y ($6 * 6 = 36$ variables).
- Cela nous obligera d'exprimer par une formule le fait que chaque mariage est entre un garçon et une fille, et en plus qu'un mariage entre X et Y est la même chose qu'un mariage entre Y et X.
- Choisir mieux les variables : on choisit des variables $[G, F]$ où G est un garçon, et F une fille.

Comment définir les variables propositionnelles?

- Variables $[X, Y]$ pour toutes personnes X, Y, exprimant le fait que X se marie avec Y ($6 * 6 = 36$ variables).
- Cela nous obligera d'exprimer par une formule le fait que chaque mariage est entre un garçon et une fille, et en plus qu'un mariage entre X et Y est la même chose qu'un mariage entre Y et X.
- Choisir mieux les variables : on choisit des variables $[G, F]$ où G est un garçon, et F une fille.
- Seulement $3 * 3 = 9$ variables :

$$[A, X], [A, Y], [A, Z], [B, X], [B, Y], [B, Z], [C, X], [C, Y], [C, Z]$$

Qu'est-ce qu'il reste à faire ?

- Modéliser les mariages entre toutes les personnes :

Qu'est-ce qu'il reste à faire ?

- Modéliser les mariages entre toutes les personnes :
  - pas de célibat

Qu'est-ce qu'il reste à faire ?

- Modéliser les mariages entre toutes les personnes :
  - pas de célibat
  - pas de bigamie

Qu'est-ce qu'il reste à faire ?

- Modéliser les mariages entre toutes les personnes :
  - pas de célibat
  - pas de bigamie
- Modéliser la condition que tout mariage est un mariage d'amour.

Absence de célibat

$$
\begin{aligned}
P_1 :=& \quad (\texttt{[A,X]} \vee \texttt{[A,Y]} \vee \texttt{[A,Z]}) \\
&\wedge \ (\texttt{[B,X]} \vee \texttt{[B,Y]} \vee \texttt{[B,Z]}) \\
&\wedge \ (\texttt{[C,X]} \vee \texttt{[C,Y]} \vee \texttt{[C,Z]}) \\
P_1' :=& \quad (\texttt{[A,X]} \vee \texttt{[B,X]} \vee \texttt{[C,X]}) \\
&\wedge \ (\texttt{[A,Y]} \vee \texttt{[B,Y]} \vee \texttt{[C,Y]}) \\
&\wedge \ (\texttt{[A,Z]} \vee \texttt{[B,Z]} \vee \texttt{[C,Z]})
\end{aligned}
$$

Absence de bigamie pour les garçons

$$
\begin{array}{lccccc}
P_2 := & & & & & \\
 & (\neg\texttt{[A,X]} \lor \neg\texttt{[A,Y]}) & \land & (\neg\texttt{[A,X]} \lor \neg\texttt{[A,Z]}) & \land & (\neg\texttt{[A,Y]} \lor \neg\texttt{[A,Z]}) \\
\land & (\neg\texttt{[B,X]} \lor \neg\texttt{[B,Y]}) & \land & (\neg\texttt{[B,X]} \lor \neg\texttt{[B,Z]}) & \land & (\neg\texttt{[B,Y]} \lor \neg\texttt{[B,Z]}) \\
\land & (\neg\texttt{[C,X]} \lor \neg\texttt{[C,Y]}) & \land & (\neg\texttt{[C,X]} \lor \neg\texttt{[C,Z]}) & \land & (\neg\texttt{[C,Y]} \lor \neg\texttt{[C,Z]})
\end{array}
$$

Absence de bigamie pour les filles

$$
\begin{array}{ll}
P_2' := & \\
 & (\neg \texttt{[A,X]} \vee \neg \texttt{[B,X]}) \quad \wedge \quad (\neg \texttt{[A,X]} \vee \neg \texttt{[C,X]}) \quad \wedge \quad (\neg \texttt{[B,X]} \vee \neg \texttt{[C,X]}) \\
\wedge & (\neg \texttt{[A,Y]} \vee \neg \texttt{[B,Y]}) \quad \wedge \quad (\neg \texttt{[A,Y]} \vee \neg \texttt{[C,Y]}) \quad \wedge \quad (\neg \texttt{[B,Y]} \vee \neg \texttt{[C,Y]}) \\
\wedge & (\neg \texttt{[A,Z]} \vee \neg \texttt{[B,Z]}) \quad \wedge \quad (\neg \texttt{[A,Z]} \vee \neg \texttt{[C,Z]}) \quad \wedge \quad (\neg \texttt{[B,Z]} \vee \neg \texttt{[C,Z]})
\end{array}
$$

Modéliser les mariages

- Condition : $P_1 \wedge P'_1 \wedge P_2 \wedge P'_2$

Modéliser les mariages

- Condition : $P_1 \wedge P_1' \wedge P_2 \wedge P_2'$
- Il aurait suffit de poser : $P_1 \wedge P_2'$ (ou $P_1' \wedge P_2$).

Modéliser les mariages

- Condition : $P_1 \wedge P'_1 \wedge P_2 \wedge P'_2$
- Il aurait suffit de poser : $P_1 \wedge P'_2$ (ou $P'_1 \wedge P_2$).
- Avec la formule complète, un solveur a plus de chances de tomber sur des clauses unaires, ce qui peut donner la solution plus rapidement.

Modéliser les mariages

- Condition : $P_1 \wedge P_1' \wedge P_2 \wedge P_2'$
- Il aurait suffit de poser : $P_1 \wedge P_2'$ (ou $P_1' \wedge P_2$).
- Avec la formule complète, un solveur a plus de chances de tomber sur des clauses unaires, ce qui peut donner la solution plus rapidement.
- Des clauses qui sont logiquement redondantes peuvent être bénéfiques si elles permet des raccourcis dans le calcul.

Deuxième étape : exprimer la contrainte spécifique

Il reste maintenant à exprimer que chaque garçon se marie avec une fille qu'il aime, et que chaque fille se marie avec un garçon qu'elle aime.

$$\begin{array}{rcl} P_3 & = & ([A, X] \vee [A, Z]) \\ & \wedge & [B, Y] \\ & \wedge & ([C, X] \vee [C, Y]) \\ & \wedge & ([B, X] \vee [C, X]) \\ & \wedge & ([A, Y] \vee [B, Y]) \\ & \wedge & ([A, Z] \vee [C, Z]) \end{array}$$

La formule entière

La formule entière est :

$$P_1 \wedge P_1' \wedge P_2 \wedge P_2' \wedge P_3$$

Subsomption

- Chacune des clauses de P_1 et de P_1' sont subsumées par des clauses en P_3.

Subsomption

- Chacune des clauses de P_1 et de P_1' sont subsumées par des clauses en P_3.
- P_1 consiste en des clauses comme

$$(\texttt{[A,X]} \vee \texttt{[A,Y]} \vee \texttt{[A,Z]})$$

(Adam se marie avec une fille)

Subsomption

- Chacune des clauses de P_1 et de P_1' sont subsumées par des clauses en P_3.
- P_1 consiste en des clauses comme

$$(\texttt{[A,X]} \vee \texttt{[A,Y]} \vee \texttt{[A,Z]})$$

(Adam se marie avec une fille)

- P_3 consiste en des clauses comme

$$([A, X] \vee [A, Z])$$

(Adam se marie avec une fille qu'il aime)

Clauses subsumées

- Des clauses redondantes sont bénéfiques seulement quand elles permettent des déductions plus rapides (avec moins de choix)

Clauses subsumées

- Des clauses redondantes sont bénéfiques seulement quand elles permettent des déductions plus rapides (avec moins de choix)
- Les clauses subsumées par des autres clauses sont redondantes, et en plus ne donnent aucun avantage dans l'algorithme DPLL.

Clauses subsumées

- Des clauses redondantes sont bénéfiques seulement quand elles permettent des déductions plus rapides (avec moins de choix)
- Les clauses subsumées par des autres clauses sont redondantes, et en plus ne donnent aucun avantage dans l'algorithme DPLL.
- On fera mieux de supprimer les clauses redondantes :

$$P_2 \wedge P_2' \wedge P_3$$

Contraintes comptage

Pour un sous-ensemble $Y \subseteq X$ de variables et un entier n une des trois formes :

1. au moins n des variables dans Y sont vraies;
2. au plus n des variables dans Y sont vraies;
3. exactement n des variables dans Y sont vraies.

Contraintes comptage

Pour un sous-ensemble $Y \subseteq X$ de variables et un entier n une des trois formes :

1. au moins n des variables dans Y sont vraies;
2. au plus n des variables dans Y sont vraies;
3. exactement n des variables dans Y sont vraies.

Exemple : Coloration de la carte d'Australie

Précisément 2 des régions sont colorées en rouges.
On a $n = 2$ et

$$Y = \{\texttt{[WA,r]}, \texttt{[NT,r]}, \texttt{[SA,r]}, \texttt{[QLD,r]}, \texttt{[NSW,r]}, \texttt{[VIC,r]}, \texttt{[TAS,r]}\}$$

Comment coder les contraintes de comptage ?

La contrainte *exactement* n *parmi les variables de* Y *sont vraies* est équivalente à la conjonction de

- au moins n parmi les variables de Y sont vraies;
- au plus n parmi les variables de Y sont vraies.

Supposons que $Y = \{y_1, \ldots, y_m\}$.

Au moins n parmi les variables de Y sont vraies

Cas $n = 1$

$$y_1 \vee y_2 \vee \ldots \vee y_m$$

Au moins n parmi les variables de Y sont vraies

Cas $n = 1$

au moins 1 est vraie

$$y_1 \vee y_2 \vee \ldots \vee y_m$$

Cas $n = 2$

engendrer toutes les paires de deux variables différentes de Y, et exprimer le fait que au moins une de ces paires est entièrement colorée en rouge :

au moins 2 sont vraies !

$$(y_1 \wedge y_2) \vee \ldots \vee (y_1 \wedge y_m) \vee (y_2 \wedge y_3) \vee \ldots \vee (y_2 \wedge y_m) \vee \ldots \vee (y_{m-1} \wedge y_m)$$

Plus précisément :

$$\bigvee_{\substack{i=1,\ldots,m-1 \\ j=i+1,\ldots,m}} (y_i \wedge y_j)$$

Comment coder les contraintes de comptage ?

Cette construction se généralise à un nombre n quelconque :

$$\bigvee_{\substack{i_1,\ldots,i_n\in\{1,\ldots,m\}\\ i_1<\ldots<i_n}} \bigwedge_{j=1,\ldots,n} y_{i_j}$$

Comment coder les contraintes de comptage ?

Cette construction se généralise à un nombre n quelconque :

$$\bigvee_{\substack{i_1,\ldots,i_n\in\{1,\ldots,m\}\\ i_1<\ldots<i_n}} \bigwedge_{j=1,\ldots,n} y_{i_j}$$

Comment construire une formule équivalente en forme CNF ?

Construction directe de la CNF équivalente

Sont équivalents :

- au moins n parmi les variables de Y sont vraies
- pour n'importe quelle sélection de $m - n + 1$ variables de Y, au moins une dans cette sélection est vraie.

Construction directe de la CNF équivalente

Sont équivalents :

- au moins n parmi les variables de Y sont vraies
- pour n'importe quelle sélection de $m - n + 1$ variables de Y, au moins une dans cette sélection est vraie.

La formule :

$$\bigwedge_{\substack{i_1,\ldots,i_{m-n+1} \\ i_1<\ldots<i_{m-n+1}}} \bigvee_{j=1,\ldots,m-n+1} y_{i_j}$$

Au moins deux régions de l'Australie sont rouges :

$$
\begin{array}{ll}
 & \big(\texttt{[WA,r]} \lor \texttt{[NT,r]} \lor \texttt{[SA,r]} \lor \texttt{[QLD,r]} \lor \texttt{[NSW,r]} \lor \texttt{[VIC,r]}\big) \\
\land & \big(\texttt{[WA,r]} \lor \texttt{[NT,r]} \lor \texttt{[SA,r]} \lor \texttt{[QLD,r]} \lor \texttt{[NSW,r]} \lor \texttt{[TAS,r]}\big) \\
\land & \big(\texttt{[WA,r]} \lor \texttt{[NT,r]} \lor \texttt{[SA,r]} \lor \texttt{[QLD,r]} \lor \texttt{[VIC,r]} \lor \texttt{[TAS,r]}\big) \\
\land & \big(\texttt{[WA,r]} \lor \texttt{[NT,r]} \lor \texttt{[SA,r]} \lor \texttt{[NSW,r]} \lor \texttt{[VIC,r]} \lor \texttt{[TAS,r]}\big) \\
\land & \big(\texttt{[WA,r]} \lor \texttt{[NT,r]} \lor \texttt{[QLD,r]} \lor \texttt{[NSW,r]} \lor \texttt{[VIC,r]} \lor \texttt{[TAS,r]}\big) \\
\land & \big(\texttt{[WA,r]} \lor \texttt{[SA,r]} \lor \texttt{[QLD,r]} \lor \texttt{[NSW,r]} \lor \texttt{[VIC,r]} \lor \texttt{[TAS,r]}\big) \\
\land & \big(\texttt{[NT,r]} \lor \texttt{[SA,r]} \lor \texttt{[QLD,r]} \lor \texttt{[NSW,r]} \lor \texttt{[VIC,r]} \lor \texttt{[TAS,r]}\big)
\end{array}
$$

au plus n des variables dans Y sont vraies

Équivalent à :

- *au moins $m - n$ des variables dans Y sont fausses*

au plus n des variables dans Y sont vraies

Équivalent à :

- *au moins $m - n$ des variables dans Y sont fausses*
- Dans toute sélection de $m - (m - n) + 1 = n + 1$ variables de Y il y a une qui est fausse

au plus n des variables dans Y sont vraies

Équivalent à :

- *au moins $m - n$ des variables dans Y sont fausses*
- Dans toute sélection de $m - (m - n) + 1 = n + 1$ variables de Y il y a une qui est fausse

La formule

$$\bigwedge_{\substack{i_1,\ldots,i_{n+1} \\ i_1 < \ldots < i_{n+1}}} \bigvee_{j=1,\ldots,n+1} \neg y_{i_j}$$

Au plus deux régions de l'Australie sont rouges :

$$
\begin{array}{ll}
 & (\neg\,\texttt{[WA,r]} \vee \neg\,\texttt{[NT,r]} \vee \neg\,\texttt{[SA,r]}) \\
\wedge & (\neg\,\texttt{[WA,r]} \vee \neg\,\texttt{[NT,r]} \vee \neg\,\texttt{[QLD,r]}) \\
\wedge & (\neg\,\texttt{[WA,r]} \vee \neg\,\texttt{[NT,r]} \vee \neg\,\texttt{[NSW,r]}) \\
\vdots & \qquad\qquad\qquad\qquad \vdots \\
\wedge & (\neg\,\texttt{[NSW,r]} \vee \neg\,\texttt{[VIC,r]} \vee \neg\,\texttt{[TAS,r]})
\end{array}
$$

Remarques finales

- En logique propositionnelle, les variables ne peuvent prendre que deux valeurs différentes.

Remarques finales

- En logique propositionnelle, les variables ne peuvent prendre que deux valeurs différentes.
- D'où le besoin d'encodages avec plusieurs variables.

Remarques finales

- En logique propositionnelle, les variables ne peuvent prendre que deux valeurs différentes.
- D'où le besoin d'encodages avec plusieurs variables.
- Il y a aussi une technique utilisant des variables qui peuvent prendre plusieurs valeurs différentes (appelées des *variables à domaine fini*).

Remarques finales

- En logique propositionnelle, les variables ne peuvent prendre que deux valeurs différentes.
- D'où le besoin d'encodages avec plusieurs variables.
- Il y a aussi une technique utilisant des variables qui peuvent prendre plusieurs valeurs différentes (appelées des *variables à domaine fini*).
- Résolution de contraintes avec des variables de domaine fini : Voir les cours de programmation logique (et par contrainte) en M1 et M2.