Chapitre 6
Modélisation en logique propositionnelle
Cours 1
Premiers Exemples

Table de Matières

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.
- Étant donné un problème combinatoire, comment construire la formule correspondante en forme CNF ?
- Comment travailler avec un SAT-solveur ?

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 ?

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 ?

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)

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)

Pourquoi construire la formule en CNF

... et pas en DNF ?

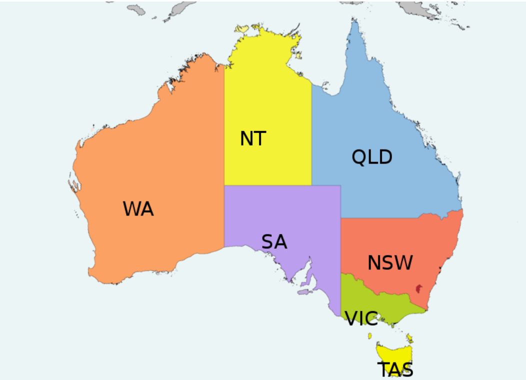
- Pour beaucoup d'applications c'est plus naturel :
- On a plusieurs contraintes à respecter (donc une conjonction)
- Chaque contrainte représente un choix entre plusieurs alternatives (donc une disjonction)

Pourquoi construire la formule en CNF

... et pas en DNF ?

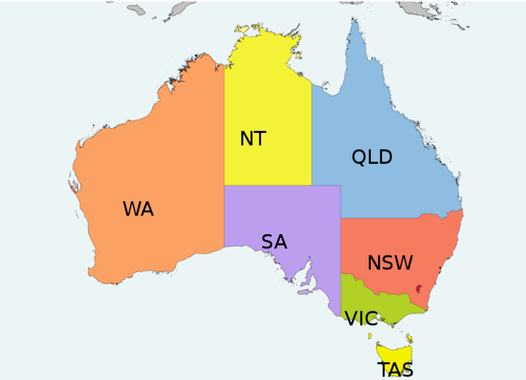
- Pour beaucoup d'applications c'est plus naturel :
- On a plusieurs contraintes à respecter (donc une conjonction)
- Chaque contrainte représente un choix entre plusieurs alternatives (donc une disjonction)

Colorier la carte de l'Australie



- Colorier cette carte avec 3 couleurs (r,b,v) seulement !

Colorier la carte de l'Australie



- Colorier cette carte avec 3 couleurs (r,b,v) seulement !

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*.

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*.

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*.

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 :

1. Modélisation des fonctions

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

2. Modélisation de la contrainte :

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

Vers une modélisation

On coupe la construction de la formule en deux parties :

1. Modélisation des fonctions

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

2. Modélisation de la contrainte :

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

Vers une modélisation

On coupe la construction de la formule en deux parties :

1. Modélisation des fonctions

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

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) :
- 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

- 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

- 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 ($n \geq 3$) ?
- 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

- 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 :

`[WA,r]`, `[WA,b]`, `[WA,v]`, `[NT,r]`, ..., `[TAS, v]`

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 !

- 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

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

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 :

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

Modélisation des colorations

Exprimer que toute région a au plus une couleur :

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

Modéliser la contrainte spécifique

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

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

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

Modéliser la contrainte spécifique

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

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

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 première étape (modélisation des coloration) 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.

- La formule complète : $P_1 \wedge P_2 \wedge P_3$, elle est déjà en forme CNF.
- La première étape (modélisation des coloration) 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.

- La formule complète : $P_1 \wedge P_2 \wedge P_3$, elle est déjà en forme CNF.
- La première étape (modélisation des coloration) 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 $A\colon \{1, \ldots, n\} \times \{1, \ldots, n\} \to \{\mathit{true}, \mathit{false}\}$ qui indique si deux pays sont adjacents.

Modélisation dans le cas général

- L'ensemble des variables propositionnelles est

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

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

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

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

Modélisation dans le cas général

- L'ensemble des variables propositionnelles est

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

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

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

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

Modélisation dans le cas général

- L'ensemble des variables propositionnelles est

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

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

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

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

Modélisation dans le cas général

- L'ensemble des variables propositionnelles est

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

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

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

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

Le programme complet

On peut maintenant facilement écrire un programme qui

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

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 (par exemple MiniSat) sur cette formule
4. analyse l'affectation trouvée par le SAT-solveur (quand une solution existe) pour afficher une coloration de la carte.

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 (par exemple MiniSat) sur cette formule
4. analyse l'affectation trouvée par le SAT-solveur (quand une solution existe) pour afficher une coloration de la carte.

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 (par exemple MiniSat) 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.
- 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 unaire, et donc déclencher un grand nombre de résolutions unaires.
- En particulier, le choix de la couleur rouge va être enlevé pour tous les pays qui sont adjacents avec WA.

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 unaire, et donc déclencher un grand nombre de résolutions unaires.
- En particulier, le choix de la couleur rouge va être enlevé pour tous les pays qui sont adjacents avec WA.

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 unaire, et donc déclencher un grand nombre de résolutions unaires.
- En particulier, le choix de la couleur rouge va être enlevé pour tous les pays qui sont adjacents avec WA.

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 unaire, et donc déclencher un grand nombre de résolutions unaires.
- 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.
- Qui aime bien qui: Adam aime bien Xanthia et Zoé, Bruno aime bien Ylenia, Chen aime bien Ylenia et Xanthia, tandis que Xanthia aime bien Bruno et Chen, Ylenia aime bien Adam et Bruno, Zoé aime bien Adam et Chen.
- Le tango se danse en couple un homme et une femme.
- Est-il possible de composer trois couples pour un concours de tango, de telle manière que chacun(e) danse avec une personne qu'il (qu'elle) aime bien?

Le problème

- Trois étudiants : Adam, Bruno, Chen.
- Trois étudiantes : Xanthia, Ylenia, Zoé.
- Qui aime bien qui: Adam aime bien Xanthia et Zoé, Bruno aime bien Ylenia, Chen aime bien Ylenia et Xanthia, tandis que Xanthia aime bien Bruno et Chen, Ylenia aime bien Adam et Bruno, Zoé aime bien Adam et Chen.
- Le tango se danse en couple un homme et une femme.
- Est-il possible de composer trois couples pour un concours de tango, de telle manière que chacun(e) danse avec une personne qu'il (qu'elle) aime bien?

Le problème

- Trois étudiants : Adam, Bruno, Chen.
- Trois étudiantes : Xanthia, Ylenia, Zoé.
- Qui aime bien qui: Adam aime bien Xanthia et Zoé, Bruno aime bien Ylenia, Chen aime bien Ylenia et Xanthia, tandis que Xanthia aime bien Bruno et Chen, Ylenia aime bien Adam et Bruno, Zoé aime bien Adam et Chen.
- Le tango se danse en couple un homme et une femme.
- Est-il possible de composer trois couples pour un concours de tango, de telle manière que chacun(e) danse avec une personne qu'il (qu'elle) aime bien?

Le problème

- Trois étudiants : Adam, Bruno, Chen.
- Trois étudiantes : Xanthia, Ylenia, Zoé.
- Qui aime bien qui: Adam aime bien Xanthia et Zoé, Bruno aime bien Ylenia, Chen aime bien Ylenia et Xanthia, tandis que Xanthia aime bien Bruno et Chen, Ylenia aime bien Adam et Bruno, Zoé aime bien Adam et Chen.
- Le tango se danse en couple un homme et une femme.
- Est-il possible de composer trois couples pour un concours de tango, de telle manière que chacun(e) danse avec une personne qu'il (qu'elle) aime bien?

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 bien 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?

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

- Tentative : variable $\langle X, Y \rangle$, où X et Y sont des personnes, qui dénotent le fait que X aime bien 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 danse avec Y ($6 * 6 = 36$ variables).
- Cela nous obligera à exprimer par une formule le fait que chaque couple comporte un garçon et une fille, et en plus que le couple X, Y est la même chose que le couple Y, X.
- Choisir mieux les variables : on choisit des variables $[G, F]$ où G est un garçons, 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]`

Comment définir les variables propositionnelles?

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

Comment définir les variables propositionnelles?

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

Comment définir les variables propositionnelles?

- Variables $[X, Y]$ pour toutes personnes X, Y, exprimant le fait que X danse avec Y ($6 * 6 = 36$ variables).
- Cela nous obligera à exprimer par une formule le fait que chaque couple comporte un garçon et une fille, et en plus que le couple X, Y est la même chose que le couple Y, X.
- Choisir mieux les variables : on choisit des variables $[G, F]$ où G est un garçons, 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 contraintes :
  - tout le monde danse.
  - personne ne danse avec deux partenaires.
- Modéliser la condition que dans chaque couple, on s'aime bien.

Qu'est-ce qu'il reste à faire ?

- Modéliser les contraintes :
  - tout le monde danse.
  - personne ne danse avec deux partenaires.
- Modéliser la condition que dans chaque couple, on s'aime bien.

Qu'est-ce qu'il reste à faire ?

- Modéliser les contraintes :
  - tout le monde danse.
  - personne ne danse avec deux partenaires.
- Modéliser la condition que dans chaque couple, on s'aime bien.

Qu'est-ce qu'il reste à faire ?

- Modéliser les contraintes :
  - tout le monde danse.
  - personne ne danse avec deux partenaires.
- Modéliser la condition que dans chaque couple, on s'aime bien.

Tous dansent

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

Pas de partenaires multiples pour les garçons

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

Pas de partenaires multiples pour les filles

$$
\begin{array}{lccccc}
P'_2 := & & & & & \\
 & (\neg\texttt{[A,X]} \vee \neg\texttt{[B,X]}) & \wedge & (\neg\texttt{[A,X]} \vee \neg\texttt{[C,X]}) & \wedge & (\neg\texttt{[B,X]} \vee \neg\texttt{[C,X]}) \\
\wedge & (\neg\texttt{[A,Y]} \vee \neg\texttt{[B,Y]}) & \wedge & (\neg\texttt{[A,Y]} \vee \neg\texttt{[C,Y]}) & \wedge & (\neg\texttt{[B,Y]} \vee \neg\texttt{[C,Y]}) \\
\wedge & (\neg\texttt{[A,Z]} \vee \neg\texttt{[B,Z]}) & \wedge & (\neg\texttt{[A,Z]} \vee \neg\texttt{[C,Z]}) & \wedge & (\neg\texttt{[B,Z]} \vee \neg\texttt{[C,Z]})
\end{array}
$$

Modéliser les couples

- 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 permettent des raccourcis dans le calcul.

Modéliser les couples

- 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 permettent des raccourcis dans le calcul.

Modéliser les couples

- 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 permettent des raccourcis dans le calcul.

Modéliser les couples

- 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 permettent des raccourcis dans le calcul.

Deuxième étape : exprimer la contrainte spécifique

Il reste maintenant à exprimer le fait que chaque garçon danse avec une fille qu'il aime bien, et vice-versa.

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

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.
- P_1 consiste en des clauses comme

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

(Adam danse avec une fille)
- P_3 consiste en des clauses comme

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

(Adam danse avec une fille qu'il aime bien)

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 danse avec une fille)

- P_3 consiste en des clauses comme

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

(Adam danse avec une fille qu'il aime bien)

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 danse avec une fille)
- P_3 consiste en des clauses comme

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

  (Adam danse avec une fille qu'il aime bien)

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 ferra mieux supprimer les clauses redondantes :

$$P_2 \wedge P'_2 \wedge P_3$$

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 ferra mieux supprimer les clauses redondantes :

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

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 ferra mieux supprimer les clauses redondantes :

$$P_2 \wedge P'_2 \wedge P_3$$