Chapitre 4
Formes Normales

Table de Matières

La notion d'une forme normale

- Pour toute formule il y a une infinité d'autres formules équivalentes : p, $p \wedge p$, $p \wedge p \wedge p$, $\ldots$, et plein d'autres.

La notion d'une forme normale

- Pour toute formule il y a une infinité d'autres formules équivalentes : p, $p \wedge p$, $p \wedge p \wedge p$, $\ldots$, et plein d'autres.
- Est-ce qu'on peut définir une forme « standardisée » des formules ?

La notion d'une forme normale

- Pour toute formule il y a une infinité d'autres formules équivalentes : p, $p \wedge p$, $p \wedge p \wedge p$, $\ldots$, et plein d'autres.
- Est-ce qu'on peut définir une forme « standardisée » des formules ?
- Forme normale : spécifiée par des restrictions de la syntaxe.
- Utile pour l'algorithmique.

La notion d'une forme normale

- Pour toute formule il y a une infinité d'autres formules équivalentes : p, $p \wedge p$, $p \wedge p \wedge p$, $\ldots$, et plein d'autres.
- Est-ce qu'on peut définir une forme « standardisée » des formules ?
- Forme normale : spécifiée par des restrictions de la syntaxe.
- Utile pour l'algorithmique.
- Normalement : toute formule doit pouvoir être transfomée en une formule en forme normale, et qui lui est équivalente, alors appelée sa forme normale.

La notion d'une forme normale

- Pour toute formule il y a une infinité d'autres formules équivalentes : p, $p \wedge p$, $p \wedge p \wedge p$, $\ldots$, et plein d'autres.
- Est-ce qu'on peut définir une forme « standardisée » des formules ?
- Forme normale : spécifiée par des restrictions de la syntaxe.
- Utile pour l'algorithmique.
- Normalement : toute formule doit pouvoir être transfomée en une formule en forme normale, et qui lui est équivalente, alors appelée sa forme normale.
- Il y a des notions différentes de forme normale.

La forme normale de négation (NNF)

Definition

Une formule est en forme normale de négation si 1) elle ne contient que les opérateurs $\{\wedge, \vee, \neg\}$ et 2) l'opérateur $\neg$ ne s'applique qu'à des variables propositionnelles.

La forme normale de négation (NNF)

Definition

Une formule est en forme normale de négation si 1) elle ne contient que les opérateurs $\{\wedge, \vee, \neg\}$ et 2) l'opérateur $\neg$ ne s'applique qu'à des variables propositionnelles.

- Exemple : $((\neg x \wedge \neg y) \wedge z)$

La forme normale de négation (NNF)

Definition

Une formule est en forme normale de négation si 1) elle ne contient que les opérateurs $\{\wedge, \vee, \neg\}$ et 2) l'opérateur $\neg$ ne s'applique qu'à des variables propositionnelles.

- Exemple : $((\neg x \wedge \neg y) \wedge z)$
- Exemple : $((\neg x \vee y) \wedge (\neg z \vee x))$

La forme normale de négation (NNF)

Definition

Une formule est en forme normale de négation si 1) elle ne contient que les opérateurs $\{\wedge, \vee, \neg\}$ et 2) l'opérateur $\neg$ ne s'applique qu'à des variables propositionnelles.

- Exemple : $((\neg x \wedge \neg y) \wedge z)$
- Exemple : $((\neg x \vee y) \wedge (\neg z \vee x))$
- Contre-exemple : $\neg(x \wedge y)$

La forme normale de négation (NNF)

Definition

Une formule est en forme normale de négation si 1) elle ne contient que les opérateurs $\{\wedge, \vee, \neg\}$ et 2) l'opérateur $\neg$ ne s'applique qu'à des variables propositionnelles.

- Exemple : $((\neg x \wedge \neg y) \wedge z)$
- Exemple : $((\neg x \vee y) \wedge (\neg z \vee x))$
- Contre-exemple : $\neg(x \wedge y)$
- Contre-exemple : $\neg\neg y$

La forme normale de négation (NNF)

Definition

Une formule est en forme normale de négation si 1) elle ne contient que les opérateurs $\{\wedge, \vee, \neg\}$ et 2) l'opérateur $\neg$ ne s'applique qu'à des variables propositionnelles.

- Exemple : $((\neg x \wedge \neg y) \wedge z)$
- Exemple : $((\neg x \vee y) \wedge (\neg z \vee x))$
- Contre-exemple : $\neg(x \wedge y)$
- Contre-exemple : $\neg\neg y$
- Contre-exemple : $x \rightarrow y$

Algorithme de transformation en NNF

Règles de réécriture :

$$\neg\neg X \leadsto X \tag{1}$$
$$\neg(X \wedge Y) \leadsto (\neg X \vee \neg Y) \tag{2}$$
$$\neg(X \vee Y) \leadsto (\neg X \wedge \neg Y) \tag{3}$$

On peut appliquer une règle de transformation à n'importe quel endroit de la formule à transformer : soit à la formule entière, soit à l'intérieur de la formule.
Si la formule contient des opperateurs autres que $\{\neg, \vee, \wedge)\}$, on applique d'abord les règles vues en chapitre 3 pour les remplacer et après on appllique les rules 1, 2, 3 ci-dessus.

Exemple

La première règle permet les transformations suivantes :

- Transformer la formule $\neg\neg(x_1 \land x_2)$ en $(x_1 \land x_2)$, en appliquant la règle à la formule entière ;

Exemple

La première règle permet les transformations suivantes :

- Transformer la formule $\neg\neg(x_1 \wedge x_2)$ en $(x_1 \wedge x_2)$, en appliquant la règle à la formule entière ;
- Transformer la formule $(x_1 \vee \neg\neg x_2)$ en $(x_1 \vee x_2)$, en appliquant la règle de transformation seulement à la sous-formule rouge.

Remarque

- Les règles de transformation sont écrites avec des variables en majuscules X, Y et pas avec des variables en minuscules x, y comme les variables propositionnelles.

Remarque

- Les règles de transformation sont écrites avec des variables en majuscules X, Y et pas avec des variables en minuscules x, y comme les variables propositionnelles.
- Les variables propositionnelles dénotent des valeurs de vérité, tandis que les variables utilisées dans les règles de transformation dénotent des formules propositionnelles.

Exemple complet

Appliquer les règles tant que possible :

$$\neg(x \wedge (y \vee \neg z))$$

Exemple complet

Appliquer les règles tant que possible :

$$\neg(x \wedge (y \vee \neg z))$$

se transforme en $(\neg x \vee \neg(y \vee \neg z))$ par règle (2)

Exemple complet

Appliquer les règles tant que possible :

$$
\begin{array}{lll}
 & \neg(x \wedge (y \vee \neg z)) & \\
\text{se transforme en} & (\neg x \vee \neg(y \vee \neg z)) & \text{par règle (2)} \\
\text{se transforme en} & (\neg x \vee (\neg y \wedge \neg\neg z)) & \text{par règle (3)}
\end{array}
$$

Exemple complet

Appliquer les règles tant que possible :

$$
\begin{array}{lll}
 & \neg(x \wedge (y \vee \neg z)) & \\
\text{se transforme en} & (\neg x \vee \neg(y \vee \neg z)) & \text{par règle (2)} \\
\text{se transforme en} & (\neg x \vee (\neg y \wedge \neg\neg z)) & \text{par règle (3)} \\
\text{se transforme en} & (\neg x \vee (\neg y \wedge z)) & \text{par règle (1)}
\end{array}
$$

Exemple complet

Appliquer les règles tant que possible :

$$
\begin{array}{lll}
 & \neg(x \wedge (y \vee \neg z)) & \\
\text{se transforme en} & (\neg x \vee \neg(y \vee \neg z)) & \text{par règle (2)} \\
\text{se transforme en} & (\neg x \vee (\neg y \wedge \neg\neg z)) & \text{par règle (3)} \\
\text{se transforme en} & (\neg x \vee (\neg y \wedge z)) & \text{par règle (1)}
\end{array}
$$

Redex

La sous-formule sur laquelle on applique une transformation, ici en rouge.

Questions

Terminaison ?

- Est-ce que ce processus se termine toujours ?

Questions

Terminaison ?

- Est-ce que ce processus se termine toujours ?

  Ce n'est pas évident, un processus défini par une application itérée d'une transformation peut a priori boucler, ou faire grossir le terme infiniment.

Questions

Terminaison ?

- Est-ce que ce processus se termine toujours ?

  Ce n'est pas évident, un processus défini par une application itérée d'une transformation peut a priori boucler, ou faire grossir le terme infiniment.

Confluence ?

- Est-ce qu'on obtient toujours à la fin la même formule quand il y a plusieurs possibilités d'appliquer les règles ?

Questions

Terminaison ?

- Est-ce que ce processus se termine toujours ?

  Ce n'est pas évident, un processus défini par une application itérée d'une transformation peut a priori boucler, ou faire grossir le terme infiniment.

Confluence ?

- Est-ce qu'on obtient toujours à la fin la même formule quand il y a plusieurs possibilités d'appliquer les règles ?

  C'est le cas, mais on ne va pas le montrer dans ce cours.

Terminaison : Regardons d'abord un système simplifié

- Si on avait une seule règle :

$$\neg\neg X \rightsquigarrow X$$

Terminaison : Regardons d'abord un système simplifié

- Si on avait une seule règle :

$$\neg\neg X \rightsquigarrow X$$

- Chaque application de cette règle enlève deux occurrences du symbole $\neg$.

Terminaison : Regardons d'abord un système simplifié

- Si on avait une seule règle :

$$\neg\neg X \rightsquigarrow X$$

- Chaque application de cette règle enlève deux occurrences du symbole $\neg$.
- Si la formule de départ a n occurrences de $\neg$, alors l'algorithme de transformation termine après au plus $\frac{n}{2}$ étapes.

La difficulté pour le système complet

- La deuxième règle

$$\neg(X \wedge Y) \rightsquigarrow (\neg X \vee \neg Y)$$

fait croître le nombre de $\neg$ (pareil pour la troisième règle)

La difficulté pour le système complet

- La deuxième règle

$$\neg(X \wedge Y) \rightsquigarrow (\neg X \vee \neg Y)$$

  fait croître le nombre de $\neg$ (pareil pour la troisième règle)
- Trouver une mesure qui décroit quand on l'applique n'importe laquelle des trois règles ?

La clef de la preuve de terminaison

Proposition

Soit ϕ : *Form* $\rightarrow \mathbb{N}$ *définie*

Alors pour toute formule $p \in$ *Form : Si* p *se transforme en* q *par une application d'une des règles 1, 2 or 3 alors* $\phi(p) > \phi(q)$.

La clef de la preuve de terminaison

Proposition

Soit ϕ : *Form* $\rightarrow \mathbb{N}$ *définie par récurrence :*

- $\phi(x) = 1$, *où* $x \in V$
- $\phi((p_1 \wedge p_2)) = \phi(p_1) + \phi(p_2)$
- $\phi((p_1 \vee p_2)) = \phi(p_1) + \phi(p_2)$
- $\phi(\neg p) = (\phi(p))^2 + 1$

Alors pour toute formule $p \in$ *Form : Si* p *se transforme en* q *par une application d'une des règles 1, 2 or 3 alors* $\phi(p) > \phi(q)$.

D'abord une petite proposition

Proposition

Pour la fonction ϕ définie en haut : $\phi(p) \geq 1$ pour toute formule p.

Démonstration : exercice (induction).

Démonstration de la proposition

- Par induction

Démonstration de la proposition

- Par induction
- À montrer pour une formule p donnée : si p se transforme en q par une application d'une règle alors $\phi(p) > \phi(q)$.

Démonstration de la proposition

- Par induction
- À montrer pour une formule p donnée : si p se transforme en q par une application d'une règle alors $\phi(p) > \phi(q)$.
- Si aucune règle de transformation ne s'applique à p alors il n'y a rien à montrer.

Démonstration de la proposition

- Par induction
- À montrer pour une formule p donnée : si p se transforme en q par une application d'une règle alors $\phi(p) > \phi(q)$.
- Si aucune règle de transformation ne s'applique à p alors il n'y a rien à montrer.
- Attention : Dans le cas général il y a plusieurs possibilités pour appliquer une règle.

Démonstration : le cas d'une variable

Si $p \in V$ alors aucune transformation n'est possible et il n'y a rien à montrer.

Démonstration : le cas d'une conjonction (I)

- Si $p = (p_1 \wedge p_2)$ alors plusieurs possibilités pour appliquer une règle peuvent exister, cela dépend de p_1 et de p_2.

Démonstration : le cas d'une conjonction (I)

- Si $p = (p_1 \wedge p_2)$ alors plusieurs possibilités pour appliquer une règle peuvent exister, cela dépend de p_1 et de p_2.
- Chaque transformation se fait soit sur p_1 soit sur p_2, il n'est pas possible d'appliquer une règle à la racine de p car toute règle commence sur le symbole $\neg$.

Démonstration : le cas d'une conjonction (I)

- Si $p = (p_1 \wedge p_2)$ alors plusieurs possibilités pour appliquer une règle peuvent exister, cela dépend de p_1 et de p_2.
- Chaque transformation se fait soit sur p_1 soit sur p_2, il n'est pas possible d'appliquer une règle à la racine de p car toute règle commence sur le symbole $\neg$.
- Nous considérons le cas où p_1 est transformé en q_1.

Démonstration : le cas d'une conjonction (I)

- Si $p = (p_1 \wedge p_2)$ alors plusieurs possibilités pour appliquer une règle peuvent exister, cela dépend de p_1 et de p_2.
- Chaque transformation se fait soit sur p_1 soit sur p_2, il n'est pas possible d'appliquer une règle à la racine de p car toute règle commence sur le symbole $\neg$.
- Nous considérons le cas où p_1 est transformé en q_1.
- Hypothèse d'induction : $\phi(p_1) > \phi(q_1)$

Démonstration : le cas d'une conjonction (II)

Hypothèse d'induction : $\phi(p_1) > \phi(q_1)$

Démonstration : le cas d'une conjonction (II)

Hypothèse d'induction : $\phi(p_1) > \phi(q_1)$

$\phi((p_1 \wedge p_2))$

$\phi((q_1 \wedge p_2))$

Démonstration : le cas d'une conjonction (II)

Hypothèse d'induction : $\phi(p_1) > \phi(q_1)$

$$\begin{array}{ll} & \phi((p_1 \wedge p_2)) \\ = & \phi(p_1) + \phi(p_2) \quad \text{par définition de } \phi \end{array}$$

$$\phi((q_1 \wedge p_2))$$

Démonstration : le cas d'une conjonction (II)

Hypothèse d'induction : $\phi(p_1) > \phi(q_1)$

$$
\begin{array}{lll}
 & \phi((p_1 \wedge p_2)) & \\
= & \phi(p_1) + \phi(p_2) & \text{par définition de } \phi \\
> & \phi(q_1) + \phi(p_2) & \text{car } \phi(p_1) > \phi(q_1) \text{ par hypothèse d'induction} \\
 & \phi((q_1 \wedge p_2)) &
\end{array}
$$

Démonstration : le cas d'une conjonction (II)

Hypothèse d'induction : $\phi(p_1) > \phi(q_1)$

$$\begin{array}{cll} & \phi((p_1 \wedge p_2)) & \\ = & \phi(p_1) + \phi(p_2) & \text{par définition de } \phi \\ > & \phi(q_1) + \phi(p_2) & \text{car } \phi(p_1) > \phi(q_1) \text{ par hypothèse d'induction} \\ = & \phi((q_1 \wedge p_2)) & \text{par définition de } \phi \end{array}$$

Démonstration : le cas d'une disjonction

- Analogue au cas de la conjonction.

Démonstration : le cas d'une disjonction

- Analogue au cas de la conjonction.
- Justifié par le fait que la fonction ϕ est définie pareil dans le cas de la conjonction et de la disjonction !

Démonstration : le cas d'une négation (I)

$p = \neg p_1$. Sous-cas 1 : transformation de p_1 en q_1.

Démonstration : le cas d'une négation (I)

$p = \neg p_1$. Sous-cas 1 : transformation de p_1 en q_1.
Hypothèse d'induction : $\phi(p_1) > \phi(q_1)$

Démonstration : le cas d'une négation (I)

$p = \neg p_1$. Sous-cas 1 : transformation de p_1 en q_1.
Hypothèse d'induction : $\phi(p_1) > \phi(q_1)$

$$\phi(\neg p_1)$$

$$\phi(\neg q_1)$$

Démonstration : le cas d'une négation (I)

$p = \neg p_1$. Sous-cas 1 : transformation de p_1 en q_1.
Hypothèse d'induction : $\phi(p_1) > \phi(q_1)$

$$\begin{array}{rll} & \phi(\neg p_1) & \\ = & (\phi(p_1))^2 + 1 & \text{par définition de } \phi \end{array}$$

$$\phi(\neg q_1)$$

Démonstration : le cas d'une négation (I)

$p = \neg p_1$. Sous-cas 1 : transformation de p_1 en q_1.
Hypothèse d'induction : $\phi(p_1) > \phi(q_1)$

$$
\begin{array}{lll}
 & \phi(\neg p_1) & \\
= & (\phi(p_1))^2 + 1 & \text{par définition de } \phi \\
> & (\phi(q_1))^2 + 1 & \text{car } \phi(p_1) > \phi(q_1) \text{ par hypothèse d'induction} \\
 & & \text{et } \phi(q_1) \geq 1 \\
 & \phi(\neg q_1) &
\end{array}
$$

Démonstration : le cas d'une négation (I)

$p = \neg p_1$. Sous-cas 1 : transformation de p_1 en q_1.
Hypothèse d'induction : $\phi(p_1) > \phi(q_1)$

$$
\begin{array}{lll}
 & \phi(\neg p_1) & \\
= & (\phi(p_1))^2 + 1 & \text{par définition de } \phi \\
> & (\phi(q_1))^2 + 1 & \text{car } \phi(p_1) > \phi(q_1) \text{ par hypothèse d'induction} \\
 & & \text{et } \phi(q_1) \geq 1 \\
= & \phi(\neg q_1) & \text{par définition de } \phi
\end{array}
$$

Démonstration : le cas d'une négation (II)

Première règle appliquée à la racine : $p = \neg\neg p_2$ est transformé en p_2.

Démonstration : le cas d'une négation (II)

Première règle appliquée à la racine : $p = \neg\neg p_2$ est transformé en p_2.

$$= \quad \begin{array}{c} \phi(\neg\neg p_2) \\ \\ \\ \phi(p_2) \end{array}$$

Démonstration : le cas d'une négation (II)

Première règle appliquée à la racine : $p = \neg\neg p_2$ est transformé en p_2.

$$\begin{array}{lll} & \phi(\neg\neg p_2) & \\ = & (\phi(\neg p_2))^2 + 1 & \text{par définition de } \phi \end{array}$$

$$\phi(p_2)$$

Démonstration : le cas d'une négation (II)

Première règle appliquée à la racine : $p = \neg\neg p_2$ est transformé en p_2.

$$\begin{array}{lll} & \phi(\neg\neg p_2) & \\ = & (\phi(\neg p_2))^2 + 1 & \text{par définition de } \phi \\ = & ((\phi(p_2))^2 + 1)^2 + 1 & \text{par définition de } \phi \end{array}$$

$$\phi(p_2)$$

Démonstration : le cas d'une négation (II)

Première règle appliquée à la racine : $p = \neg\neg p_2$ est transformé en p_2.

$$\begin{array}{cll} & \phi(\neg\neg p_2) & \\ = & (\phi(\neg p_2))^2 + 1 & \text{par définition de } \phi \\ = & ((\phi(p_2))^2 + 1)^2 + 1 & \text{par définition de } \phi \\ = & \phi(p_2)^4 + 2\phi(p_2)^2 + 1 + 1 & \\ & \phi(p_2) & \end{array}$$

Démonstration : le cas d'une négation (II)

Première règle appliquée à la racine : $p = \neg\neg p_2$ est transformé en p_2.

$$
\begin{array}{lll}
 & \phi(\neg\neg p_2) & \\
= & (\phi(\neg p_2))^2 + 1 & \text{par définition de } \phi \\
= & ((\phi(p_2))^2 + 1)^2 + 1 & \text{par définition de } \phi \\
= & \phi(p_2)^4 + 2\phi(p_2)^2 + 1 + 1 & \\
> & \phi(p_2) &
\end{array}
$$

Démonstration : le cas d'une négation (III)

Application de la deuxième règle à la racine : $p = \neg(p_2 \wedge p_3)$ est transformé en $(\neg p_2 \vee \neg p_3)$:

Démonstration : le cas d'une négation (III)

Application de la deuxième règle à la racine : $p = \neg(p_2 \wedge p_3)$ est transformé en $(\neg p_2 \vee \neg p_3)$:

$$\phi(\neg(p_2 \wedge p_3))$$

$$\phi((\neg p_2 \vee \neg p_3))$$

Démonstration : le cas d'une négation (III)

Application de la deuxième règle à la racine : $p = \neg(p_2 \wedge p_3)$ est transformé en $(\neg p_2 \vee \neg p_3)$:

$$\begin{array}{lll} & \phi(\neg(p_2 \wedge p_3)) & \\ = & \phi((p_2 \wedge p_3))^2 + 1 & \text{par définition de } \phi \end{array}$$

$$\phi((\neg p_2 \vee \neg p_3))$$

Démonstration : le cas d'une négation (III)

Application de la deuxième règle à la racine : $p = \neg(p_2 \wedge p_3)$ est transformé en $(\neg p_2 \vee \neg p_3)$:

$$\begin{array}{lll} & \phi(\neg(p_2 \wedge p_3)) & \\ = & \phi((p_2 \wedge p_3))^2 + 1 & \text{par définition de } \phi \\ = & (\phi(p_2) + \phi(p_3))^2 + 1 & \text{par définition de } \phi \end{array}$$

$$\phi((\neg p_2 \vee \neg p_3))$$

Démonstration : le cas d'une négation (III)

Application de la deuxième règle à la racine : $p = \neg(p_2 \wedge p_3)$ est transformé en $(\neg p_2 \vee \neg p_3)$:

$$
\begin{array}{lll}
 & \phi(\neg(p_2 \wedge p_3)) & \\
= & \phi((p_2 \wedge p_3))^2 + 1 & \text{par définition de } \phi \\
= & (\phi(p_2) + \phi(p_3))^2 + 1 & \text{par définition de } \phi \\
= & (\phi(p_2))^2 + 2\phi(p_2)\phi(p_3) + (\phi(p_3))^2 + 1 &
\end{array}
$$

$$\phi((\neg p_2 \vee \neg p_3))$$

Démonstration : le cas d'une négation (III)

Application de la deuxième règle à la racine : $p = \neg(p_2 \wedge p_3)$ est transformé en $(\neg p_2 \vee \neg p_3)$:

$$
\begin{array}{cll}
 & \phi(\neg(p_2 \wedge p_3)) & \\
= & \phi((p_2 \wedge p_3))^2 + 1 & \text{par définition de } \phi \\
= & (\phi(p_2) + \phi(p_3))^2 + 1 & \text{par définition de } \phi \\
= & (\phi(p_2))^2 + 2\phi(p_2)\phi(p_3) + (\phi(p_3))^2 + 1 & \\
> & (\phi(p_2))^2 + 1 + (\phi(p_3))^2 + 1 & \text{car } \phi(p_2) \geq 1 \\
 & & \text{et } \phi(p_3) \geq 1 \\
 & & \\
 & \phi((\neg p_2 \vee \neg p_3)) &
\end{array}
$$

Démonstration : le cas d'une négation (III)

Application de la deuxième règle à la racine : $p = \neg(p_2 \wedge p_3)$ est transformé en $(\neg p_2 \vee \neg p_3)$:

$$
\begin{array}{lll}
 & \phi(\neg(p_2 \wedge p_3)) & \\
= & \phi((p_2 \wedge p_3))^2 + 1 & \text{par définition de } \phi \\
= & (\phi(p_2) + \phi(p_3))^2 + 1 & \text{par définition de } \phi \\
= & (\phi(p_2))^2 + 2\phi(p_2)\phi(p_3) + (\phi(p_3))^2 + 1 & \\
> & (\phi(p_2))^2 + 1 + (\phi(p_3))^2 + 1 & \text{car } \phi(p_2) \geq 1 \\
 & & \text{et } \phi(p_3) \geq 1 \\
= & \phi(\neg p_2) + \phi(\neg p_3) & \text{par définition de } \phi \\
 & \phi((\neg p_2 \vee \neg p_3)) &
\end{array}
$$

Démonstration : le cas d'une négation (III)

Application de la deuxième règle à la racine : $p = \neg(p_2 \wedge p_3)$ est transformé en $(\neg p_2 \vee \neg p_3)$:

$$
\begin{array}{lll}
 & \phi(\neg(p_2 \wedge p_3)) & \\
= & \phi((p_2 \wedge p_3))^2 + 1 & \text{par définition de } \phi \\
= & (\phi(p_2) + \phi(p_3))^2 + 1 & \text{par définition de } \phi \\
= & (\phi(p_2))^2 + 2\phi(p_2)\phi(p_3) + (\phi(p_3))^2 + 1 & \\
> & (\phi(p_2))^2 + 1 + (\phi(p_3))^2 + 1 & \text{car } \phi(p_2) \geq 1 \\
 & & \text{et } \phi(p_3) \geq 1 \\
= & \phi(\neg p_2) + \phi(\neg p_3) & \text{par définition de } \phi \\
= & \phi((\neg p_2 \vee \neg p_3)) & \text{par définition de } \phi
\end{array}
$$

Démonstration : le cas d'une négation (IV)

Il s'agit d'une application de la troisième règle à la racine : analogue au cas précédent.

Mise en forme normale de négation

Theorem

Pour toute formule il existe une formule équivalente en forme normale de négation.

Mise en forme normale de négation

Theorem

Pour toute formule il existe une formule équivalente en forme normale de négation.

Démonstration : Donné p, on lui applique des règles de transformation tant que possible. À montrer :

Mise en forme normale de négation

Theorem

Pour toute formule il existe une formule équivalente en forme normale de négation.

Démonstration : Donné p, on lui applique des règles de transformation tant que possible. À montrer :

1. Le processus termine.

Mise en forme normale de négation

Theorem

Pour toute formule il existe une formule équivalente en forme normale de négation.

Démonstration : Donné p, on lui applique des règles de transformation tant que possible. À montrer :

1. Le processus termine.
2. Chaque étape transforme une formule dans une formule équivalente.

Mise en forme normale de négation

Theorem

Pour toute formule il existe une formule équivalente en forme normale de négation.

Démonstration : Donné p, on lui applique des règles de transformation tant que possible. À montrer :

1. Le processus termine.
2. Chaque étape transforme une formule dans une formule équivalente.
3. Si aucune règle de transformation ne s'applique alors la formule est en forme normale de négation.

Démonstration de (1): Le processus termine

- À chaque transformation la mesure ϕ décroît strictement (montré au-dessus).

Démonstration de (1): Le processus termine

- À chaque transformation la mesure ϕ décroît strictement (montré au-dessus).
- Cette mesure ne peut jamais descendre au-dessous de 1 (petite proposition au-dessus)

Démonstration de (1): Le processus termine

- À chaque transformation la mesure ϕ décroît strictement (montré au-dessus).
- Cette mesure ne peut jamais descendre au-dessous de 1 (petite proposition au-dessus)
- Conséquence : si la formule de départ p a la mesure $\phi(p) = n$ alors on peut au maximum appliquer $n - 1$ transformations.

Démonstration de (2): Chaque étape transforme une formule dans une formule équivalente

Conséquence des lois de la logique propositionnelle.

Démonstration de (3): Si aucune règle de transformation s'applique alors la formule est en forme normale de négation

Supposons qu'aucune règle ne s'applique à p, et que p contient une sous-formule $\neg q$.

Démonstration de (3): Si aucune règle de transformation s'applique alors la formule est en forme normale de négation

Supposons qu'aucune règle ne s'applique à p, et que p contient une sous-formule $\neg q$.

- q ne peut pas être une négation (sinon : règle (1) s'applique)

Démonstration de (3): Si aucune règle de transformation s'applique alors la formule est en forme normale de négation

Supposons qu'aucune règle ne s'applique à p, et que p contient une sous-formule $\neg q$.

- q ne peut pas être une négation (sinon : règle (1) s'applique)
- q ne peut pas être une conjonction (sinon : règle (2) s'applique)

Démonstration de (3): Si aucune règle de transformation s'applique alors la formule est en forme normale de négation

Supposons qu'aucune règle ne s'applique à p, et que p contient une sous-formule $\neg q$.

- q ne peut pas être une négation (sinon : règle (1) s'applique)
- q ne peut pas être une conjonction (sinon : règle (2) s'applique)
- q ne peut pas être une disjonction (sinon : règle (3) s'applique)

Démonstration de (3): Si aucune règle de transformation s'applique alors la formule est en forme normale de négation

Supposons qu'aucune règle ne s'applique à p, et que p contient une sous-formule $\neg q$.

- q ne peut pas être une négation (sinon : règle (1) s'applique)
- q ne peut pas être une conjonction (sinon : règle (2) s'applique)
- q ne peut pas être une disjonction (sinon : règle (3) s'applique)

Donc, q doit être une variable.

Littéral, Clause, DNF

Definition

1. Un littéral est soit une variable propositionnelle, soit la négation d'une variable propositionnelle.

Littéral, Clause, DNF

Definition

1. Un littéral est soit une variable propositionnelle, soit la négation d'une variable propositionnelle.
2. Une clause conjonctive (également appelée un terme) est soit la constante $\top$, soit un littéral, soit une conjonction d'aux moins deux littéraux.

Littéral, Clause, DNF

Definition

1. Un **littéral** est soit une variable propositionnelle, soit la négation d'une variable propositionnelle.
2. Une **clause conjonctive** (également appelée un **terme**) est soit la constante $\top$, soit un littéral, soit une conjonction d'aux moins deux littéraux.
3. Une formule en **forme disjonctive normale** (DNF) est soit la constante $\perp$, soit une clause conjonctive, soit une disjonction d'aux moins deux clauses conjonctives.

- « conjonction d'aux moins deux littéraux » : formule de la forme
$$(l_1 \wedge (l_2 \wedge (l_3 \wedge \ldots) \ldots))$$
où chacun des l_i est un littéral

- « conjonction d'aux moins deux littéraux » : formule de la forme
$$(l_1 \wedge (l_2 \wedge (l_3 \wedge \ldots) \ldots))$$
où chacun des l_i est un littéral
- « disjonction d'aux moins deux clauses conjonctives » : formule de la forme
$$(c_1 \vee (c_2 \vee (c_3 \vee \ldots) \ldots))$$
où chacun des c_i est une clause conjonctive.

Exemples

Clauses conjonctives (syntaxe raccourcie)

- $\top$
- x
- $(x \wedge y \wedge z)$
- $(x \wedge \neg x \wedge y)$

Exemples

Clauses conjonctives (syntaxe raccourcie)

- $\top$
- x
- $(x \wedge y \wedge z)$
- $(x \wedge \neg x \wedge y)$

Forme disjonctive normale (syntaxe raccourcie)

$$(x \wedge y \wedge z) \vee (x \wedge \neg x \wedge y) \vee x$$

Convention

- La conjonction d'une seule formule w est w

Convention

- La conjonction d'une seule formule w est w
- La disjonction d'une seule formule w est w

Éléments neutres

- La conjonction de zéro formules donne $\top$.

Éléments neutres

- La conjonction de zéro formules donne $\top$.
- Justification : Quand on ajoute zéro formules à une conjonction on devrait obtenir une formule équivalente.

Éléments neutres

- La conjonction de zéro formules donne $\top$.
- Justification : Quand on ajoute zéro formules à une conjonction on devrait obtenir une formule équivalente.
- C.-à-d. : la conjonction de zéro formules est l'élément neutre de la conjonction : $\top$

Éléments neutres

- La conjonction de zéro formules donne $\top$.
- Justification : Quand on ajoute zéro formules à une conjonction on devrait obtenir une formule équivalente.
- C.-à-d. : la conjonction de zéro formules est l'élément neutre de la conjonction : $\top$
- Anologue à : la somme de zéro nombres est zéro $\Sigma_{i=1}^{i=0} i = 0$

Éléments neutres

- La conjonction de zéro formules donne $\top$.
- Justification : Quand on ajoute zéro formules à une conjonction on devrait obtenir une formule équivalente.
- C.-à-d. : la conjonction de zéro formules est l'élément neutre de la conjonction : $\top$
- Anologue à : la somme de zéro nombres est zéro $\Sigma_{i=1}^{i=0} i = 0$
- La disjonction de zéro formules donne $\perp$ (pareil)

Simplification de la définition

1. Une clause conjonctive est une conjonction de zéro ou plus littéraux.

Simplification de la définition

1. Une clause conjonctive est une conjonction de zéro ou plus littéraux.
2. Une formule est en forme disjonctive normale si elle est la disjonction de zéro ou plus clauses conjonctives.

NNF et DNF

- Toute formule en DNF est aussi en NNF

NNF et DNF

- Toute formule en DNF est aussi en NNF
- Il y a des formules en NNF qui ne sont pas DNF.

NNF et DNF

- Toute formule en DNF est aussi en NNF
- Il y a des formules en NNF qui ne sont pas DNF.
- Exemple :

$$(x \wedge (\neg y \vee \neg z))$$

Transformation en DNF

- D'abord mettre la formule en NNF

Remarque : Si on permet la syntaxe raccourcie on n'a besoin que des règles 4 et 5.

Transformation en DNF

- D'abord mettre la formule en NNF
- Puis appliquer tant que possible les règles :

$$X \wedge (Y \vee Z) \rightsquigarrow (X \wedge Y) \vee (X \wedge Z) \tag{4}$$

$$(X \vee Y) \wedge Z \rightsquigarrow (X \wedge Z) \vee (Y \wedge Z) \tag{5}$$

$$(X \wedge Y) \wedge Z \rightsquigarrow X \wedge (Y \wedge Z) \tag{6}$$

$$(X \vee Y) \vee Z \rightsquigarrow X \vee (Y \vee Z) \tag{7}$$

Remarque : Si on permet la syntaxe raccourcie on n'a besoin que des règles 4 et 5.

Exemple

$$x_1 \wedge (\neg(y_1 \vee y_2) \wedge \neg\neg(z_1 \vee z_2))$$

Exemple

$$x_1 \wedge (\neg(y_1 \vee y_2) \wedge \neg\neg(z_1 \vee z_2))$$

$$x_1 \wedge (\neg(y_1 \vee y_2) \wedge (z_1 \vee z_2)) \qquad \text{par règle 1}$$

Exemple

$$x_1 \wedge (\neg(y_1 \vee y_2) \wedge \neg\neg(z_1 \vee z_2))$$

$$x_1 \wedge (\neg(y_1 \vee y_2) \wedge (z_1 \vee z_2)) \qquad \text{par règle 1}$$

$$x_1 \wedge ((\neg y_1 \wedge \neg y_2) \wedge (z_1 \vee z_2)) \text{ [NNF]} \qquad \text{par règle 3}$$

Exemple

$$x_1 \wedge (\neg(y_1 \vee y_2) \wedge \neg\neg(z_1 \vee z_2))$$

$$x_1 \wedge (\neg(y_1 \vee y_2) \wedge (z_1 \vee z_2)) \qquad \text{par règle 1}$$

$$x_1 \wedge ((\neg y_1 \wedge \neg y_2) \wedge (z_1 \vee z_2)) \text{ [NNF]} \qquad \text{par règle 3}$$

$$x_1 \wedge (((\neg y_1 \wedge \neg y_2) \wedge z_1) \vee ((\neg y_1 \wedge \neg y_2) \wedge z_2)) \qquad \text{par règle 4}$$

Exemple

$x_1 \wedge (\neg(y_1 \vee y_2) \wedge \neg\neg(z_1 \vee z_2))$	
$x_1 \wedge (\neg(y_1 \vee y_2) \wedge (z_1 \vee z_2))$	par règle 1
$x_1 \wedge ((\neg y_1 \wedge \neg y_2) \wedge (z_1 \vee z_2))$ **[NNF]**	par règle 3
$x_1 \wedge (((\neg y_1 \wedge \neg y_2) \wedge z_1) \vee ((\neg y_1 \wedge \neg y_2) \wedge z_2))$	par règle 4
$x_1 \wedge ((\neg y_1 \wedge \neg y_2 \wedge z_1) \vee (\neg y_1 \wedge \neg y_2 \wedge z_2))$	par règle 6 (2 fois)

Exemple

$x_1 \wedge (\neg(y_1 \vee y_2) \wedge \neg\neg(z_1 \vee z_2))$	
$x_1 \wedge (\neg(y_1 \vee y_2) \wedge (z_1 \vee z_2))$	par règle 1
$x_1 \wedge ((\neg y_1 \wedge \neg y_2) \wedge (z_1 \vee z_2))$ **[NNF]**	par règle 3
$x_1 \wedge (((\neg y_1 \wedge \neg y_2) \wedge z_1) \vee ((\neg y_1 \wedge \neg y_2) \wedge z_2))$	par règle 4
$x_1 \wedge ((\neg y_1 \wedge \neg y_2 \wedge z_1) \vee (\neg y_1 \wedge \neg y_2 \wedge z_2))$	par règle 6 (2 fois)
$(x_1 \wedge \neg y_1 \wedge \neg y_2 \wedge z_1) \vee (x_1 \wedge \neg y_1 \wedge \neg y_2 \wedge z_2)$	par règle 5

Le théorème de terminaison

Proposition

Soit ψ : *Form* $\rightarrow \mathbb{N}$ *définie par récurrence :*

- $\psi(x) = 2$ *si* $x \in V$
- $\psi((p_1 \wedge p_2)) = (\psi(p_1))^2 * \psi(p_2)$
- $\psi((p_1 \vee p_2)) = 2\psi(p_1) + \psi(p_2) + 1$
- $\psi(\neg p) = \psi(p)$

Alors pour toute formule $p \in$ *Form : Si* p *se transforme en* q *par une application d'une des règles 4 à 7 alors* $\psi(p) > \psi(q)$.

Le théorème de terminaison

Proposition

Soit ψ : *Form* $\rightarrow \mathbb{N}$ *définie par récurrence :*

- $\psi(x) = 2$ *si* $x \in V$
- $\psi((p_1 \wedge p_2)) = (\psi(p_1))^2 * \psi(p_2)$
- $\psi((p_1 \vee p_2)) = 2\psi(p_1) + \psi(p_2) + 1$
- $\psi(\neg p) = \psi(p)$

Alors pour toute formule $p \in$ *Form : Si* p *se transforme en* q *par une application d'une des règles 4 à 7 alors* $\psi(p) > \psi(q)$.

Démonstration : exercice

Mise en forme DNF

Theorem

Pour toute formule il existe une formule équivalente en forme disjonctive normale.

Mise en forme DNF

Theorem

Pour toute formule il existe une formule équivalente en forme disjonctive normale.

Démonstration : Mettre en forme NNF, puis appliquer les règles de transformaton en DNF tant que possible. À montrer :

Mise en forme DNF

Theorem

Pour toute formule il existe une formule équivalente en forme disjonctive normale.

Démonstration : Mettre en forme NNF, puis appliquer les règles de transformaton en DNF tant que possible. À montrer :

1. Le processus termine.

Mise en forme DNF

Theorem

Pour toute formule il existe une formule équivalente en forme disjonctive normale.

Démonstration : Mettre en forme NNF, puis appliquer les règles de transformaton en DNF tant que possible. À montrer :

1. Le processus termine.
2. Le résultat est équivalent à la formule de départ.

Mise en forme DNF

Theorem

Pour toute formule il existe une formule équivalente en forme disjonctive normale.

Démonstration : Mettre en forme NNF, puis appliquer les règles de transformaton en DNF tant que possible. À montrer :

1. Le processus termine.
2. Le résultat est équivalent à la formule de départ.
3. À la fin on obtient une DNF

Démonstration du Théorème de mise en DNF (I)

1. Le processus termine : conséquence de la proposition (décroissance de la fonction ψ).

Démonstration du Théorème de mise en DNF (I)

1. Le processus termine : conséquence de la proposition (décroissance de la fonction ψ).
2. Le résultat est équivalent à la formule de départ : conséquence du fait que chaque étape de transformation est une équivalence (voir les lois de la logique propositionnelle)

Démonstration du Théorème de mise en DNF (II)

On veut montrer qu'à la fin on obtient une DNF. Supposons par l'absurde qu'aucune transformation ne s'applique à p, mais p n'est pas en forme disjonctive normale.

Démonstration du Théorème de mise en DNF (II)

On veut montrer qu'à la fin on obtient une DNF. Supposons par l'absurde qu'aucune transformation ne s'applique à p, mais p n'est pas en forme disjonctive normale.

- Si p n'est pas en forme normale de négation : ce n'est pas possible car aucune des règles 4 à 7 ne peut transformer une formule en forme normale de négation en une formule qui n'est pas en forme normale de négation.

Démonstration du Théorème de mise en DNF (II)

On veut montrer qu'à la fin on obtient une DNF. Supposons par l'absurde qu'aucune transformation ne s'applique à p, mais p n'est pas en forme disjonctive normale.

- Si p n'est pas en forme normale de négation : ce n'est pas possible car aucune des règles 4 à 7 ne peut transformer une formule en forme normale de négation en une formule qui n'est pas en forme normale de négation.
- Si p contient une sous-formule $(p_1 \wedge p_2) \wedge p_3$ ou $(p_1 \vee p_2) \vee p_3$: Une des règles (6) ou (7) s'applique alors, contradiction.

Démonstration du Théorème de mise en DNF (II)

On veut montrer qu'à la fin on obtient une DNF. Supposons par l'absurde qu'aucune transformation ne s'applique à p, mais p n'est pas en forme disjonctive normale.

- Si p n'est pas en forme normale de négation : ce n'est pas possible car aucune des règles 4 à 7 ne peut transformer une formule en forme normale de négation en une formule qui n'est pas en forme normale de négation.
- Si p contient une sous-formule $(p_1 \wedge p_2) \wedge p_3$ ou $(p_1 \vee p_2) \vee p_3$: Une des règles (6) ou (7) s'applique alors, contradiction.
- S'il y a en p une disjonction au-dessous d'une conjonction, c'est-à-dire p contient une sous-formule de la forme $(p_1 \wedge (p_2 \vee p_3))$ ou de la forme $((p_1 \vee p_2) \wedge p_3)$: Une des règles (4) ou (5) s'applique alors, contradiction.

Explosion de la taille

Attention, la transformation d'une formule en forme disjonctive normale risque de faire croître la taille de la formule de façon exponentielle. Par exemple : la DNF de la formule

$$(x_1 \vee y_1) \wedge (x_2 \vee y_2)$$

est

Explosion de la taille

Attention, la transformation d'une formule en forme disjonctive normale risque de faire croître la taille de la formule de façon exponentielle. Par exemple : la DNF de la formule

$$(x_1 \vee y_1) \wedge (x_2 \vee y_2)$$

est

$$\begin{array}{cc} & (x_1 \wedge x_2) \\ \vee & (x_1 \wedge y_2) \\ \vee & (y_1 \wedge x_2) \\ \vee & (y_1 \wedge y_2) \end{array}$$

Explosion de la taille (2)

La mise en forme disjonctive normale de

$$(x_1 \vee y_1) \wedge (x_2 \vee y_2) \wedge (x_3 \vee y_3)$$

donne

$$\begin{array}{ll} & (x_1 \wedge x_2 \wedge x_3) \\ \vee & (x_1 \wedge x_2 \wedge y_3) \\ \vee & (x_1 \wedge y_2 \wedge x_3) \\ \vee & (x_1 \wedge y_2 \wedge y_3) \\ \vee & (y_1 \wedge x_2 \wedge x_3) \\ \vee & (y_1 \wedge x_2 \wedge y_3) \\ \vee & (y_1 \wedge y_2 \wedge x_3) \\ \vee & (y_1 \wedge y_2 \wedge y_3) \end{array}$$

Explosion de la taille (3)

En général la mise en forme disjonctive normale de la formule

$$(x_1 \vee y_1) \wedge (x_2 \vee y_2) \wedge \ldots \wedge (x_n \vee y_n)$$

donne une formule avec 2^n clauses conjonctives, chaque clause consistant en n littéraux.

DNF et satisfaisabilité

Theorem

Une formule en forme disjonctive normale

$$(c_1 \vee c_2 \vee \ldots \vee c_n)$$

est satisfaisable si et seulement s'il y a un i *tel que la clause* c_i *ne contient pas à la fois une variable* x *et aussi sa négation.*

Démonstration

Une formule en DNF est satisfaisable si et seulement si elle contient au moins une clause conjonctive qui est satisfaisable.

Démonstration

Une formule en DNF est satisfaisable si et seulement si elle contient au moins une clause conjonctive qui est satisfaisable.
Il reste à montrer qu'une clause conjonctive

$$c = (l_1 \wedge l_2 \wedge \ldots \wedge l_m)$$

est satisfaisable si et seulement si elle ne contient pas à la fois un littéral x et aussi sa négation $\neg x$.

Démonstration

Une formule en DNF est satisfaisable si et seulement si elle contient au moins une clause conjonctive qui est satisfaisable.
Il reste à montrer qu'une clause conjonctive

$$c = (l_1 \wedge l_2 \wedge \ldots \wedge l_m)$$

est satisfaisable si et seulement si elle ne contient pas à la fois un littéral x et aussi sa négation $\neg x$.

- Si c contient un littéral x et aussi $\neg x$ alors évidemment c n'est pas satisfaisable.

Démonstration

Une formule en DNF est satisfaisable si et seulement si elle contient au moins une clause conjonctive qui est satisfaisable.
Il reste à montrer qu'une clause conjonctive

$$c = (l_1 \wedge l_2 \wedge \ldots \wedge l_m)$$

est satisfaisable si et seulement si elle ne contient pas à la fois un littéral x et aussi sa négation $\neg x$.

- Si c contient un littéral x et aussi $\neg x$ alors évidemment c n'est pas satisfaisable.
- Si c ne contient pas à la fois un littéral x et aussi $\neg x$ alors c est satisfaite par l'affectation v définie par

$$v(x) = \begin{cases} 1 & \text{si } x \text{ est un littéral en } c \\ 0 & \text{si } \neg x \text{ est un littéral en } c \text{, ou si } x \notin \mathcal{V}(c) \end{cases}$$

Méthode pratique pour décider la satisfaisabilité?

- Idée naïve : transformer la formule en DNF, puis appliquer le théorème pour savoir si elle est satisfaisable.

Méthode pratique pour décider la satisfaisabilité?

- Idée naïve : transformer la formule en DNF, puis appliquer le théorème pour savoir si elle est satisfaisable.
- Problème : la formule mise en DNF peut être « exponentiellement » plus grande. Cet algorithme n'est pas plus efficace que la méthode par table de vérité.

Question (version 1)

Question 1 : Est-il possible que deux formules différentes en forme disjonctive normale soient équivalentes ?

Question (version 1)

Question 1 : Est-il possible que deux formules différentes en forme disjonctive normale soient équivalentes ?

Réponse : oui ! La différence peut être dans l'ordre des littéraux ou des clauses :

$$(x \wedge y) \vee (\neg z \wedge \neg y)$$

Question (version 1)

Question 1 : Est-il possible que deux formules différentes en forme disjonctive normale soient équivalentes ?

Réponse : oui ! La différence peut être dans l'ordre des littéraux ou des clauses :

$$
\begin{array}{ll}
 & (x \wedge y) \vee (\neg z \wedge \neg y) \\
\models\!\mid & (y \wedge x) \vee (\neg z \wedge \neg y)
\end{array}
$$

Question (version 1)

Question 1 : Est-il possible que deux formules différentes en forme disjonctive normale soient équivalentes ?

Réponse : oui ! La différence peut être dans l'ordre des littéraux ou des clauses :

$$\begin{array}{rl} & (x \wedge y) \vee (\neg z \wedge \neg y) \\ \models\!\mid & (y \wedge x) \vee (\neg z \wedge \neg y) \\ \models\!\mid & (\neg z \wedge \neg y) \vee (y \wedge x) \end{array}$$

Question (version 2) ?

Question 2 : Est-il possible que deux formules en forme disjonctive normale avec plus de différence que simplement l'ordre des littéraux ou l'ordre des clauses, soient équivalentes ?

Question (version 2) ?

Question 2 : Est-il possible que deux formules en forme disjonctive normale avec plus de différence que simplement l'ordre des littéraux ou l'ordre des clauses, soient équivalentes ?

Réponse : oui !

$$(x \wedge y)$$
$$(x \wedge y) \vee (x \wedge y \wedge z)$$

Subsomption

Definition

Une clause conjonctive $l_1 \wedge \ldots \wedge l_n$ subsume une clause conjonctive $k_1 \wedge \ldots \wedge k_m$ si pour tout i avec $1 \leq i \leq n$ il existe un j avec $1 \leq j \leq m$ tel que $k_j = l_i$.

Subsomption

Definition

Une clause conjonctive $l_1 \wedge \ldots \wedge l_n$ subsume une clause conjonctive $k_1 \wedge \ldots \wedge k_m$ si pour tout i avec $1 \leq i \leq n$ il existe un j avec $1 \leq j \leq m$ tel que $k_j = l_i$.

C-à-d : Une clause c subsume une clause d si c est « incluse » en d à l'ordre des littéraux et à des éventuelles répétitions près.

Subsomption

Definition

Une clause conjonctive $l_1 \wedge \ldots \wedge l_n$ subsume une clause conjonctive $k_1 \wedge \ldots \wedge k_m$ si pour tout i avec $1 \leq i \leq n$ il existe un j avec $1 \leq j \leq m$ tel que $k_j = l_i$.

C-à-d : Une clause c subsume une clause d si c est « incluse » en d à l'ordre des littéraux et à des éventuelles répétitions près.

Exemples :

- $x \wedge \neg y$ subsume $z \wedge x \wedge \neg y$

Subsomption

Definition

Une clause conjonctive $l_1 \wedge \ldots \wedge l_n$ subsume une clause conjonctive $k_1 \wedge \ldots \wedge k_m$ si pour tout i avec $1 \leq i \leq n$ il existe un j avec $1 \leq j \leq m$ tel que $k_j = l_i$.

C-à-d : Une clause c subsume une clause d si c est « incluse » en d à l'ordre des littéraux et à des éventuelles répétitions près.

Exemples :

- $x \wedge \neg y$ subsume $z \wedge x \wedge \neg y$
- $x \wedge \neg y$ subsume $z \wedge \neg y \wedge z \wedge x$.

Propriété de la subsomption

Proposition

Si la clause conjonctive c subsume la clause conjonctive d alors $d \models c$.

Par exemple $x \wedge \neg y \wedge z \models x \wedge \neg y$.

Clauses subsumées sont redondantes

Theorem

Soient $c_1, \ldots, c_n$ *des clauses conjonctives. Si* c_j *subsume* c_i *pour* $i \neq j$ *alors*

$$c_1 \vee \ldots \vee c_{i-1} \vee c_i \vee c_{i+1} \vee \ldots \vee c_n \models\mid c_1 \vee \ldots \vee c_{i-1} \vee c_{i+1} \vee \ldots \vee c_n$$

Clauses subsumées sont redondantes

Theorem

Soient $c_1, \ldots, c_n$ des clauses conjonctives. Si c_j subsume c_i pour $i \neq j$ alors

$$c_1 \vee \ldots \vee c_{i-1} \vee c_i \vee c_{i+1} \vee \ldots \vee c_n \models\!\!\!\!\dashv c_1 \vee \ldots \vee c_{i-1} \vee c_{i+1} \vee \ldots \vee c_n$$

Exemple :

$$(x \wedge \neg y) \vee (\neg x \wedge y) \vee (x \wedge \neg y \wedge z_1 \wedge \neg z_2) \models\!\!\!\!\dashv (x \wedge \neg y) \vee (\neg x \wedge y)$$

Question (version 3)

Question 3 : Est-il possible que deux formules en forme disjonctive normale avec plus de différence que simplement l'ordre des littéraux ou l'ordre des clauses, et où aucune clause subsume une autre dans la même formule, soient équivalentes ?

Réponse à la question, version 3

Oui ! Exemple :

Réponse à la question, version 3

Oui ! Exemple :

$$\neg x \vee (x \wedge \neg y)$$

Réponse à la question, version 3

Oui ! Exemple :

$$
\begin{array}{rl}
 & \neg x \vee (x \wedge \neg y) \\
\models\!\mid & \neg y \vee (y \wedge \neg x)
\end{array}
$$

Réponse à la question, version 3

Oui ! Exemple :

$$\begin{array}{cc} & \neg x \vee (x \wedge \neg y) \\ \models\!\mid & \neg y \vee (y \wedge \neg x) \end{array}$$

Ces deux formules sont équivalentes à

$$(\neg x \wedge y) \vee (\neg x \wedge \neg y) \vee (x \wedge \neg y)$$

Preuve de la première équivalence

$$\neg x \vee (x \wedge \neg y)$$

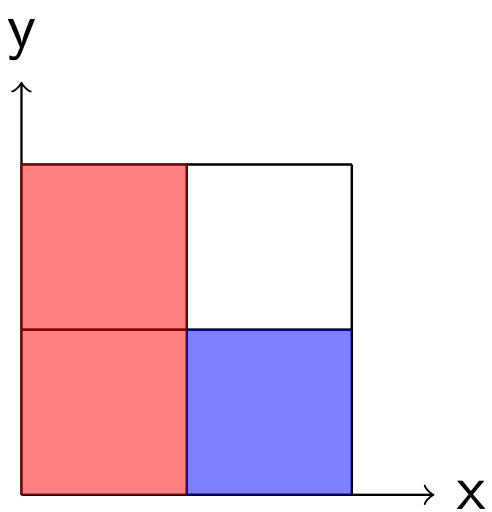
Preuve de la première équivalence

$$\begin{array}{lll} & \neg x \vee (x \wedge \neg y) & \\ \models\!\mid & (\neg x \wedge (y \vee \neg y)) \vee (x \wedge \neg y) & \top \text{ est l'élément neutre de } \wedge \end{array}$$

Preuve de la première équivalence

$$
\begin{array}{lll}
 & \neg x \vee (x \wedge \neg y) & \\
\models\!\!\!\mid & (\neg x \wedge (y \vee \neg y)) \vee (x \wedge \neg y) & \top \text{ est l'élément neutre de } \wedge \\
\models\!\!\!\mid & (\neg x \wedge y) \vee (\neg x \wedge \neg y) \vee (x \wedge \neg y) & \text{Distributivité}
\end{array}
$$

Illustration

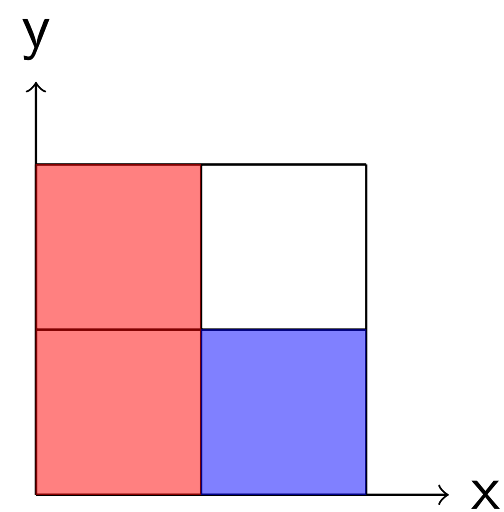


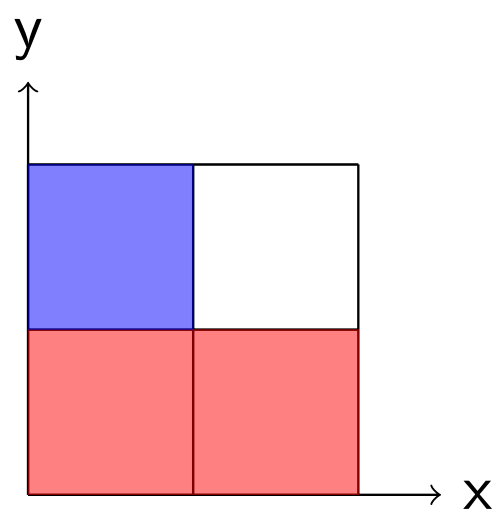
$\neg x$

$\vee\ (x \wedge \neg y)$ $\qquad \vee \qquad \vee$

$\vee$

Illustration



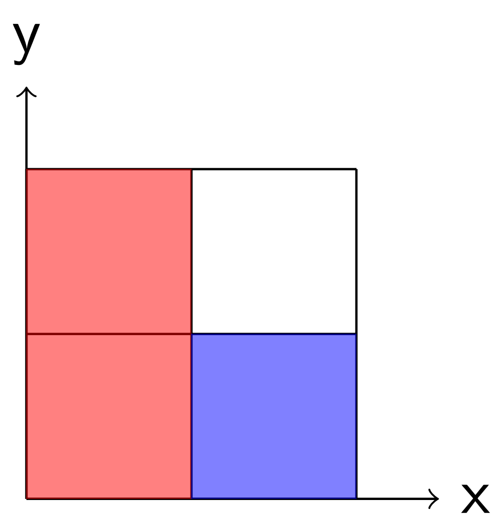


$\neg x$ $\qquad$ $\neg y$

$\vee\ (x \wedge \neg y)$ $\qquad$ $\vee (\neg x \wedge y)$ $\qquad$ $\vee$

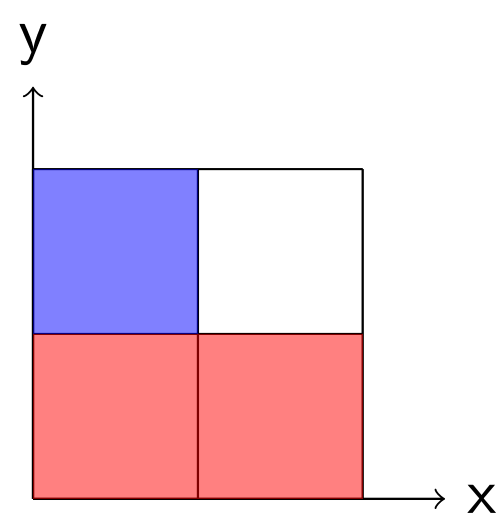
$\vee$

Illustration



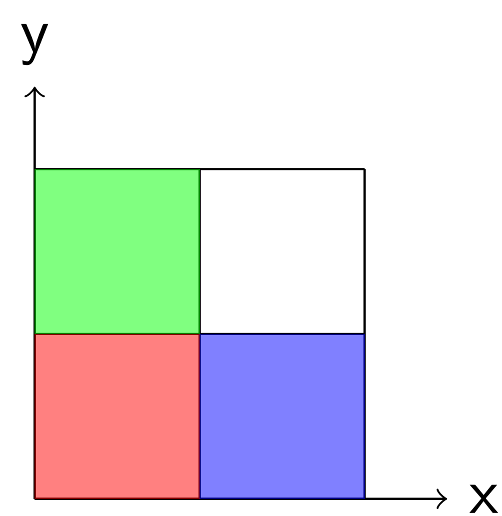
$\neg x$

$\vee\ (x \wedge \neg y)$



$\neg y$

$\vee (\neg x \wedge y)$



$(\neg x \wedge \neg y)$

$\vee (x \wedge \neg y)$

$\vee (\neg x \wedge y)$

Il y a encore une autre . . .

. . . formule en DNF qui est équivalente à $\neg x \vee (x \wedge \neg y)$, et qui est même plus courte :

Il y a encore une autre . . .

. . . formule en DNF qui est équivalente à $\neg x \vee (x \wedge \neg y)$, et qui est même plus courte :

$$\neg x \vee \neg y \tag{8}$$

Preuve de l'équivalence

$$\neg x \vee \neg y$$

Preuve de l'équivalence

$$\begin{array}{lll} & \neg x \vee \neg y & \\ \models\!\!\mid & \neg x \vee (\neg y \wedge (x \vee \neg x)) & \top \text{ est l'élément neutre de } \wedge \end{array}$$

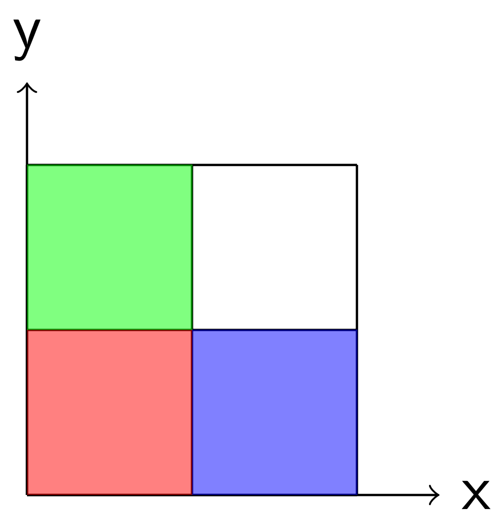
Preuve de l'équivalence

$$\begin{array}{lll} & \neg x \vee \neg y & \\ \models\!\!| & \neg x \vee (\neg y \wedge (x \vee \neg x)) & \top \text{ est l'élément neutre de } \wedge \\ \models\!\!| & \neg x \vee (\neg y \wedge x) \vee (\neg y \wedge \neg x) & \text{Distributivité} \end{array}$$

Preuve de l'équivalence

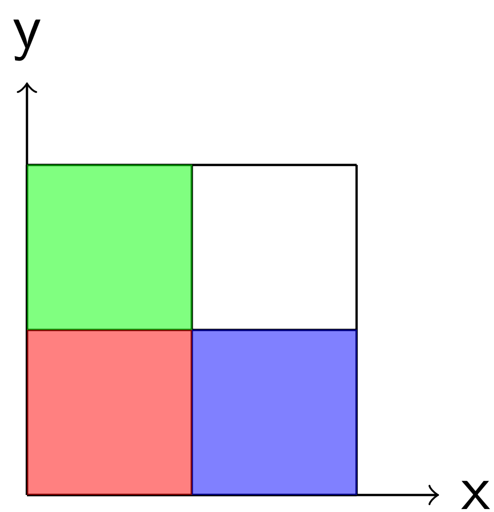
$$
\begin{array}{lll}
 & \neg x \vee \neg y & \\
\models\!\!\!\!\!\dashv & \neg x \vee (\neg y \wedge (x \vee \neg x)) & \top \text{ est l'élément neutre de } \wedge \\
\models\!\!\!\!\!\dashv & \neg x \vee (\neg y \wedge x) \vee (\neg y \wedge \neg x) & \text{Distributivité} \\
\models\!\!\!\!\!\dashv & \neg x \vee (x \wedge \neg y) & \text{car } \neg x \text{ subsume } \neg y \wedge \neg x
\end{array}
$$

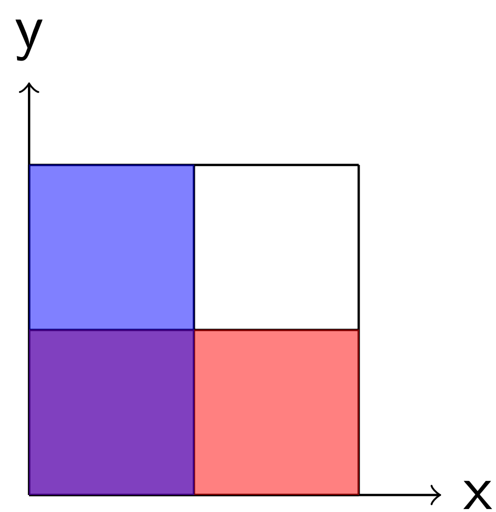
Illustration



$(\neg x \wedge \neg y)$
$\vee(x \wedge \neg y)$ $\vee$
$\vee(\neg x \wedge y)$

Illustration





$(\neg x \land \neg y)$
$\lor(x \land \neg y)$
$\lor(\neg x \land y)$

$\neg y$
$\lor \neg x$

DNF minimale

- Taille d'une formule en DNF : nombre d'occurrences de littéraux

DNF minimale

- Taille d'une formule en DNF : nombre d'occurrences de littéraux
- Exemple : $(x \wedge \neg y) \vee (x \wedge z)$ a taille 4.

DNF minimale

- Taille d'une formule en DNF : nombre d'occurrences de littéraux
- Exemple : $(x \wedge \neg y) \vee (x \wedge z)$ a taille 4.
- DNF minimale : DNF, qui est de taille minimale parmi toutes les DNF équivalentes

DNF minimale

- Taille d'une formule en DNF : nombre d'occurrences de littéraux
- Exemple : $(x \wedge \neg y) \vee (x \wedge z)$ a taille 4.
- DNF minimale : DNF, qui est de taille minimale parmi toutes les DNF équivalentes
- $\neg x \vee \neg y$ est une DNF minimale.

Parenthèse : minimaliser des DNF

- Problème très étudié dans le contexte de l'architecture des circuits.

Parenthèse : minimaliser des DNF

- Problème très étudié dans le contexte de l'architecture des circuits.
- Algorithme le plus célèbre dû à Quine et McCluskey

Parenthèse : minimaliser des DNF

- Problème très étudié dans le contexte de l'architecture des circuits.
- Algorithme le plus célèbre dû à Quine et McCluskey
- Voir le cours d'architecture d'ordinateurs.

Question (version 4) ?

Est-ce que les DNF minimales sont uniques, à l'ordre des littéraux et des clauses près ?

Il n'y a toujours pas d'unicité

$$(\neg x \wedge y) \vee (x \wedge \neg y) \vee (x \wedge y \wedge z)$$

Cette formule est équivalente aux deux formules plus petites en forme disjonctive normale

$$(\neg x \wedge y) \vee (x \wedge \neg y) \vee (x \wedge z)$$
$$(\neg x \wedge y) \vee (x \wedge \neg y) \vee (y \wedge z)$$

Forme conjonctive normale

Analogue à la forme disjonctive normale :

Definition

1. Une clause disjonctive est une disjonction de zéro ou plus littéraux.
2. Une formule est en forme conjonctive normale (abrégé CNF) si elle est la conjonction de zéro ou plus clauses disjonctives.

Exemples CNF

- $(x \vee y) \wedge (\neg z \vee y \vee \neg z)$

Exemples CNF

- $(x \vee y) \wedge (\neg z \vee y \vee \neg z)$
- $(x \vee \neg y \vee z)$ (qui est aussi en DNF !)

Exemples CNF

- $(x \vee y) \wedge (\neg z \vee y \vee \neg z)$
- $(x \vee \neg y \vee z)$ (qui est aussi en DNF !)
- $(x \wedge \neg y \wedge z)$ (qui est aussi en DNF !)

Mise en forme CNF

Theorem

Pour toute formule il existe une formule équivalente en forme conjonctive normale.

Mise en forme CNF

Theorem

Pour toute formule il existe une formule équivalente en forme conjonctive normale.

Procédure est analogue à celle de la mise en forme disjonctive normale, sauf que les deux premières règles de transformation (4) et (5) sont à remplacer par

$$X \vee (Y \wedge Z) \quad \rightsquigarrow \quad (X \vee Y) \wedge (X \vee Z) \tag{9}$$

$$(X \wedge Y) \vee Z \quad \rightsquigarrow \quad (X \vee Z) \wedge (Y \vee Z) \tag{10}$$

CNF et validité

Theorem

Une formule en forme conjonctive normale

$$(d_1 \wedge d_2 \wedge \ldots \wedge d_n)$$

est valide si et seulement si pour tout i *il existe une variable* x *telle que la clause* d_i *contient à la fois le littéral* x *et aussi sa négation* $\neg x$.

Subsomption de clauses disjonctives

Definition

Une clause disjonctive $l_1 \vee \ldots \vee l_n$ subsume une clause disjonctive $k_1 \vee \ldots \vee k_m$ si pour tout i avec $1 \leq i \leq n$ il existe un j avec $1 \leq j \leq m$ tel que $k_j = l_i$.

Par exemple, $x \vee \neg y$ subsume $x \vee \neg y \vee z$.

Subsomption de clauses disjonctives

Proposition

Si la clause disjonctive c subsume la clause disjonctive d alors $c \models d$.

Remarquez que le sens de la conséquence est inverse à celui de la proposition sur la subsomption des clauses conjonctives.

Par exemple $x \vee \neg y \models x \vee \neg y \vee z$.

Les clauses subsumées sont redondantes

Theorem

Soient $d_1, \ldots, d_n$ *des clauses disjonctives. Si* d_j *subsume* d_i *pour* $i \neq j$ *alors*

$$d_1 \wedge \ldots \wedge d_{i-1} \wedge d_i \wedge d_{i+1} \wedge \ldots \wedge d_n \models\!\!\!\dashv d_1 \wedge \ldots \wedge d_{i-1} \wedge d_{i+1} \wedge \ldots \wedge d_n$$

Satisfaisabilité et Validité

Formule en	DNF	CNF
Satisfaisabilité :	trivial	NP-complet
Validité :	coNP-complet	trivial

Voir le chapitre 5 pour un algorithme efficace en pratique.