Principes de fonctionnement des machines binaires

2022–2023

Matthieu Picantin

Tests et examens

- CC : résultat des tests TD/TP/moodle
- E1 : examen sur table janvier
- E2 : examen sur table mi juin

Notes finales: contrôle continu intégral

- Note session 1 : 50% CC + 50% E1
- Note session 2 : max(E2, 33% CC + 67% E2)

Rappel

pas de note $\Longrightarrow$ pas de moyenne $\Longrightarrow$ pas de semestre

`moodle.u-paris.fr`

Tests et examens

- CC : résultat des tests TD/TP/moodle
- E1 : examen sur table janvier
- E2 : examen sur table mi juin

Notes finales: contrôle continu intégral

- Note session 1 : 50% CC + 50% E1
- Note session 2 : max(E2, 33% CC + 67% E2)

Rappel

pas de note $\Longrightarrow$ pas de moyenne $\Longrightarrow$ pas de semestre

`moodle.u-paris.fr`

Tests et examens

- CC : résultat des tests TD/TP/moodle
- E1 : examen sur table janvier
- E2 : examen sur table mi juin

Notes finales: contrôle continu intégral

- Note session 1 : 50% CC + 50% E1
- Note session 2 : max(E2, 33% CC + 67% E2)

Rappel

pas de note $\Longrightarrow$ pas de moyenne $\Longrightarrow$ pas de semestre

moodle.u-paris.fr

Tests et examens

- CC : résultat des tests TD/TP/moodle
- E1 : examen sur table janvier
- E2 : examen sur table mi juin

Notes finales: contrôle continu intégral

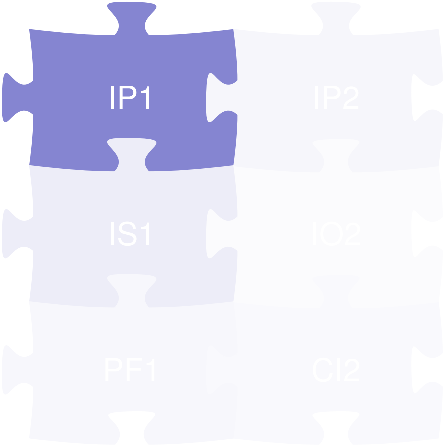
- Note session 1 : 50% CC + 50% E1
- Note session 2 : max(E2, 33% CC + 67% E2)

Rappel

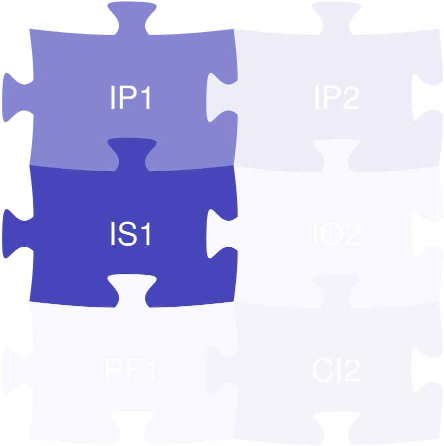
pas de note $\Longrightarrow$ pas de moyenne $\Longrightarrow$ pas de semestre

`moodle.u-paris.fr`

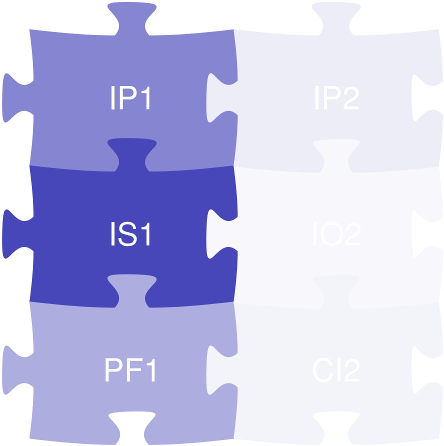
moodle.u-paris.fr



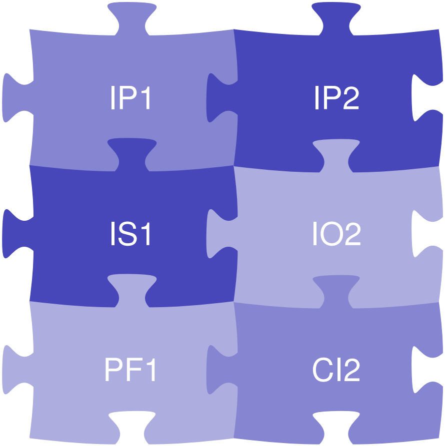
moodle.u-paris.fr



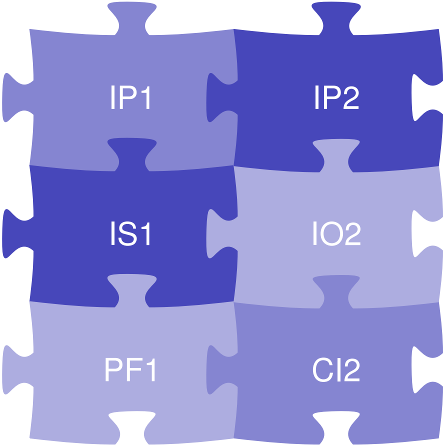
moodle.u-paris.fr



moodle.u-paris.fr



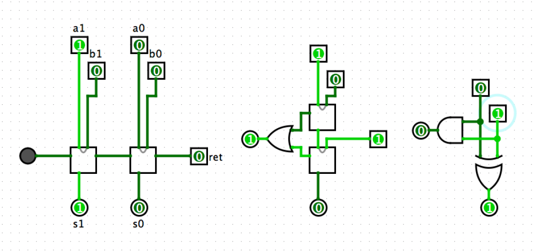
moodle.u-paris.fr



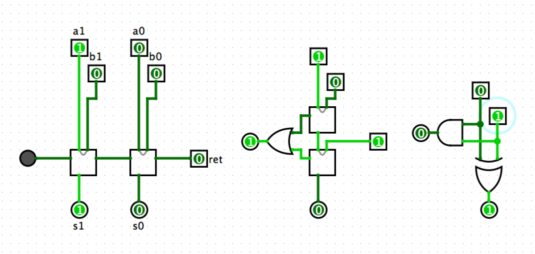
`moodle.u-paris.fr`

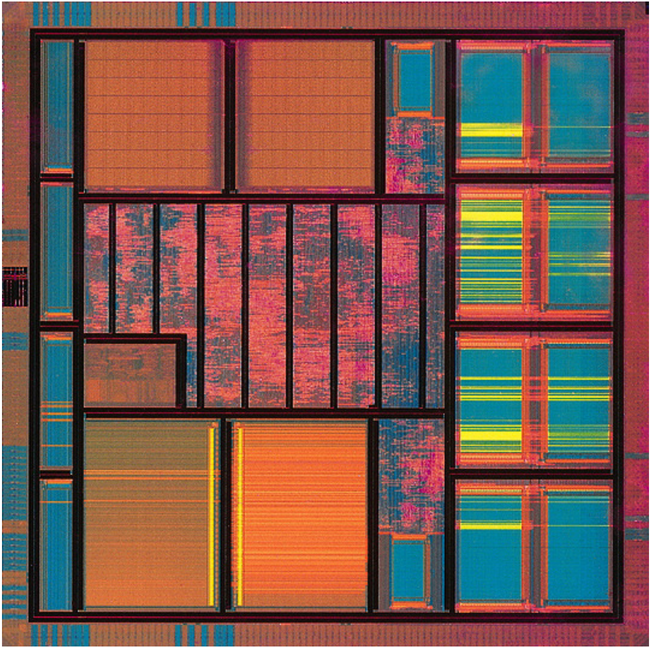
- numération et arithmétique
- numération et arithmétique en machine
- codes, codages, compression, crypto
- contrôle d'erreur (détection, correction)
- logique et calcul propositionnel
- circuits numériques

- numération et arithmétique
- numération et arithmétique en machine
- codes, codages, compression, crypto
- contrôle d'erreur (détection, correction)
- logique et calcul propositionnel
- circuits numériques



- numération et arithmétique
- numération et arithmétique en machine
- codes, codages, compression, crypto
- contrôle d'erreur (détection, correction)
- logique et calcul propositionnel
- circuits numériques





Calcul propositionnel

- système logique dans lequel on considère
  - un ensemble de variables propositionnelles
  - des formules logiques construites à partir de ces variables avec des connecteurs logiques
- système paraissant simpliste, mais très utile et utilisé

Problèmes de décision

- problème de validité
  - telle formule est-elle toujours vraie, peu importent les valeurs des variables?
- problème de satisfiabilité
  - existe-t-il des valeurs des variables pour lesquelles telle formule est vraie?
- deux problèmes décidables, mais de complexité élevée

Calcul propositionnel

- système logique dans lequel on considère
  - un ensemble de variables propositionnelles
  - des formules logiques construites à partir de ces variables avec des connecteurs logiques
- système paraissant simpliste, mais très utile et utilisé

Problèmes de décision

- problème de validité
  - telle formule est-elle toujours vraie, peu importent les valeurs des variables?
- problème de satisfiabilité
  - existe-t-il des valeurs des variables pour lesquelles telle formule est vraie?
- deux problèmes décidables, mais de complexité élevée

Calcul propositionnel

- système logique dans lequel on considère
  - un ensemble de variables propositionnelles
  - des formules logiques construites à partir de ces variables avec des connecteurs logiques
- système paraissant simpliste, mais très utile et utilisé

Problèmes de décision

- problème de validité
  - telle formule est-elle toujours vraie, peu importent les valeurs des variables?
- problème de satisfiabilité
  - existe-t-il des valeurs des variables pour lesquelles telle formule est vraie?
- deux problèmes décidables, mais de complexité élevée

Calcul propositionnel

- système logique dans lequel on considère
  - un ensemble de variables propositionnelles
  - des formules logiques construites à partir de ces variables avec des connecteurs logiques
- système paraissant simpliste, mais très utile et utilisé

Problèmes de décision

- problème de validité
  - telle formule est-elle toujours vraie, peu importent les valeurs des variables?
- problème de satisfiabilité
  - existe-t-il des valeurs des variables pour lesquelles telle formule est vraie?
- deux problèmes décidables, mais de complexité élevée

Calcul propositionnel

- système logique dans lequel on considère
  - un ensemble de variables propositionnelles
  - des formules logiques construites à partir de ces variables avec des connecteurs logiques
- système paraissant simpliste, mais très utile et utilisé

Problèmes de décision

- problème de validité
  - telle formule est-elle toujours vraie, peu importent les valeurs des variables?
- problème de satisfiabilité
  - existe-t-il des valeurs des variables pour lesquelles telle formule est vraie?
- deux problèmes décidables, mais de complexité élevée

Calcul propositionnel

- système logique dans lequel on considère
  - un ensemble de variables propositionnelles
  - des formules logiques construites à partir de ces variables avec des connecteurs logiques
- système paraissant simpliste, mais très utile et utilisé

Problèmes de décision

- problème de validité
  - telle formule est-elle toujours vraie, peu importent les valeurs des variables?
- problème de satisfiabilité
  - existe-t-il des valeurs des variables pour lesquelles telle formule est vraie?
- deux problèmes décidables, mais de complexité élevée

Calcul propositionnel

- système logique dans lequel on considère
  - un ensemble de variables propositionnelles
  - des formules logiques construites à partir
    de ces variables avec des connecteurs logiques
- système paraissant simpliste, mais très utile et utilisé

Problèmes de décision

- problème de validité
  - telle formule est-elle toujours vraie, peu importent les valeurs des variables?
- problème de satisfiabilité
  - existe-t-il des valeurs des variables pour lesquelles telle formule est vraie?
- deux problèmes décidables, mais de complexité élevée

Calcul propositionnel

- système logique dans lequel on considère
  - un ensemble de variables propositionnelles
  - des formules logiques construites à partir de ces variables avec des connecteurs logiques
- système paraissant simpliste, mais très utile et utilisé

Syntaxe

- définition formelle des énoncés:
  - appelés généralement formules
  - exprimés dans un langage symbolique
  - dont la structure peut être analysée par un parseur
- donnée par une définition inductive ou une grammaire

Sémantique

- association d'une valeur de vérité ($\top$ ou $\bot$) à chaque formule
- définie par récurrence sur la syntaxe

Calcul propositionnel

- système logique dans lequel on considère
  - un ensemble de variables propositionnelles
  - des formules logiques construites à partir de ces variables avec des connecteurs logiques
- système paraissant simpliste, mais très utile et utilisé

Syntaxe

- définition formelle des énoncés:
  - appelés généralement formules
  - exprimés dans un langage symbolique
  - dont la structure peut être analysée par un parseur
- donnée par une définition inductive ou une grammaire

Sémantique

- association d'une valeur de vérité ($\top$ ou $\bot$) à chaque formule
- définie par récurrence sur la syntaxe

Algèbre de Boole

- calcul des propositions dont le domaine $\mathbb{B}$ est celui des booléens
  - $\mathbb{B}$ peut être $\{\text{FAUX}, \text{VRAI}\}$, $\{\texttt{false}, \texttt{true}\}$, $\{\bot, \top\}$, ou $\{0, 1\}$
  - ces deux éléments sont appelés valeurs de vérité
- ensemble $\mathbb{B}$ à deux éléments muni d'un opérateur unaire $\neg$ (négation) et de deux opérateurs binaires $\vee$ (disjonction) et $\wedge$ (conjonction)

Négation

x	$\neg x$
0	1
1	0

Disjonction

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Conjonction

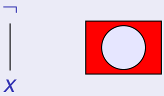
x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Algèbre de Boole

- calcul des propositions dont le domaine $\mathbb{B}$ est celui des **booléens**
  - $\mathbb{B}$ peut être $\{\text{FAUX}, \text{VRAI}\}$, $\{\texttt{false}, \texttt{true}\}$, $\{\bot, \top\}$, ou $\{0, 1\}$
  - ces deux éléments sont appelés **valeurs de vérité**
- ensemble $\mathbb{B}$ à deux éléments muni d'un opérateur unaire $\neg$ (négation) et de deux opérateurs binaires $\vee$ (disjonction) et $\wedge$ (conjonction)

Négation

x	$\neg x$
0	1
1	0



Disjonction

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Conjonction

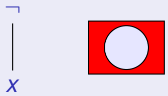
x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Algèbre de Boole

- calcul des propositions dont le domaine $\mathbb{B}$ est celui des booléens
  - $\mathbb{B}$ peut être $\{\text{FAUX}, \text{VRAI}\}$, $\{\texttt{false}, \texttt{true}\}$, $\{\bot, \top\}$, ou $\{0, 1\}$
  - ces deux éléments sont appelés valeurs de vérité
- ensemble $\mathbb{B}$ à deux éléments muni d'un opérateur unaire $\neg$ (négation) et de deux opérateurs binaires $\vee$ (disjonction) et $\wedge$ (conjonction)

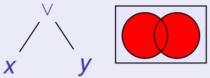
Négation

x	$\neg x$
0	1
1	0



Disjonction

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1



Conjonction

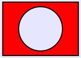
x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Algèbre de Boole

- calcul des propositions dont le domaine $\mathbb{B}$ est celui des booléens
  - $\mathbb{B}$ peut être $\{\text{FAUX}, \text{VRAI}\}$, $\{\texttt{false}, \texttt{true}\}$, $\{\bot, \top\}$, ou $\{0, 1\}$
  - ces deux éléments sont appelés valeurs de vérité
- ensemble $\mathbb{B}$ à deux éléments muni d'un opérateur unaire $\neg$ (négation) et de deux opérateurs binaires $\vee$ (disjonction) et $\wedge$ (conjonction)

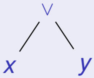
Négation

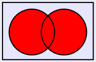
x	$\neg x$
0	1
1	0



Disjonction

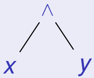
x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

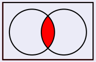


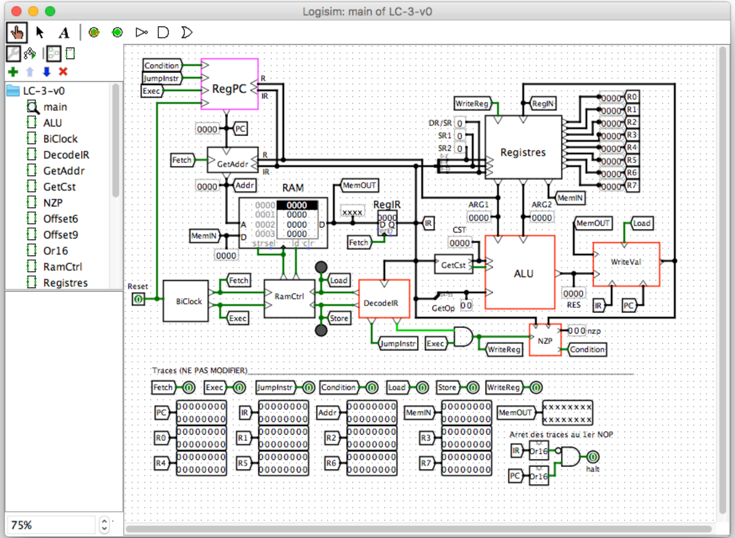


Conjonction

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1







https://logisim.altervista.org/

Algèbre de Boole

- ensemble $\mathbb{B} = \{0, 1\}$ muni d'un opérateur unaire $\neg$ (négation) et de deux opérateurs binaires $\vee$ (disjonction) et $\wedge$ (conjonction)
- vérifiant, pour toutes variables booléennes x, y, z de $\mathbb{B}$, les axiomes suivants :

associativité	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$
commutativité	$x \vee y = y \vee x$	$x \wedge y = y \wedge x$
absorption	$x \vee (x \wedge y) = x$	$x \wedge (x \vee y) = x$
distributivité	$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
complémentation	$x \vee \neg x = 1$	$x \wedge \neg x = 0$

- ayant, pour toutes variables booléennes x, y de $\mathbb{B}$, les propriétés suivantes :

idempotence	$x \vee x = x$	$x \wedge x = x$
neutres	$x \vee 0 = x$	$x \wedge 1 = x$
involution	$\neg\neg x = x$	
De Morgan	$\neg(x \vee y) = \neg x \wedge \neg y$	$\neg(x \wedge y) = \neg x \vee \neg y$

- interprétation des valeurs :

$$0 \leftrightsquigarrow \text{false} \qquad \text{et} \qquad 1 \leftrightsquigarrow \text{true}$$

Algèbre de Boole

- ensemble $\mathbb{B} = \{0, 1\}$ muni d'un opérateur unaire $\neg$ (négation) et de deux opérateurs binaires $\vee$ (disjonction) et $\wedge$ (conjonction)
- vérifiant, pour toutes variables booléennes x, y, z de $\mathbb{B}$, les axiomes suivants :

associativité	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$
commutativité	$x \vee y = y \vee x$	$x \wedge y = y \wedge x$
absorption	$x \vee (x \wedge y) = x$	$x \wedge (x \vee y) = x$
distributivité	$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
complémentation	$x \vee \neg x = 1$	$x \wedge \neg x = 0$

- ayant, pour toutes variables booléennes x, y de $\mathbb{B}$, les propriétés suivantes :

idempotence	$x \vee x = x$	$x \wedge x = x$
neutres	$x \vee 0 = x$	$x \wedge 1 = x$
involution	$\neg\neg x = x$	
De Morgan	$\neg(x \vee y) = \neg x \wedge \neg y$	$\neg(x \wedge y) = \neg x \vee \neg y$

- interprétation des valeurs :

$$0 \leadsto \text{false} \qquad \text{et} \qquad 1 \leadsto \text{true}$$

Algèbre de Boole

- ensemble $\mathbb{B} = \{0, 1\}$ muni d'un opérateur unaire $\neg$ (négation) et de deux opérateurs binaires $\vee$ (disjonction) et $\wedge$ (conjonction)
- vérifiant, pour toutes variables booléennes x, y, z de $\mathbb{B}$, les axiomes suivants :

associativité	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$
commutativité	$x \vee y = y \vee x$	$x \wedge y = y \wedge x$
absorption	$x \vee (x \wedge y) = x$	$x \wedge (x \vee y) = x$
distributivité	$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
complémentation	$x \vee \neg x = 1$	$x \wedge \neg x = 0$

- ayant, pour toutes variables booléennes x, y de $\mathbb{B}$, les propriétés suivantes :

idempotence	$x \vee x = x$	$x \wedge x = x$
neutres	$x \vee 0 = x$	$x \wedge 1 = x$
involution	$\neg\neg x = x$	
De Morgan	$\neg(x \vee y) = \neg x \wedge \neg y$	$\neg(x \wedge y) = \neg x \vee \neg y$

- interprétation des valeurs :

$0 \leftrightsquigarrow$ false   et $1 \leftrightsquigarrow$ true

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les quatre opérateurs booléens unaires ($k = 1$)

$f(0)$	$f(1)$

Il existe exactement 2^{2^k} fonctions booléennes d'arité k

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les quatre opérateurs booléens unaires ($k = 1$)

$f(0)$	$f(1)$

Il existe exactement 2^{2^k} fonctions booléennes d'arité k

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les quatre opérateurs booléens unaires ($k = 1$)

$f(0)$	$f(1)$	
0	0	contradiction

Il existe exactement 2^{2^k} fonctions booléennes d'arité k

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les quatre opérateurs booléens unaires ($k = 1$)

$f(0)$	$f(1)$	
0	0	contradiction
0	1	affirmation

Il existe exactement 2^{2^k} fonctions booléennes d'arité k

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les quatre opérateurs booléens unaires ($k = 1$)

$f(0)$	$f(1)$	
0	0	contradiction
0	1	affirmation
1	0	négation

Il existe exactement 2^{2^k} fonctions booléennes d'arité k

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les quatre opérateurs booléens unaires ($k = 1$)

$f(0)$	$f(1)$	
0	0	contradiction
0	1	affirmation
1	0	négation
1	1	tautologie

Il existe exactement 2^{2^k} fonctions booléennes d'arité k

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les quatre opérateurs booléens unaires ($k = 1$)

$f(0)$	$f(1)$	
0	0	contradiction
0	1	affirmation
1	0	négation **NOT**
1	1	tautologie

Il existe exactement 2^{2^k} fonctions booléennes d'arité k

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les quatre opérateurs booléens unaires ($k = 1$)

	$f(0)$	$f(1)$	
	0	0	contradiction
	0	1	affirmation
	1	0	négation **NOT**
	1	1	tautologie

Il existe exactement 2^{2^k} fonctions booléennes d'arité k

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les quatre opérateurs booléens unaires ($k = 1$)

	$f(0)$	$f(1)$	
	0	0	contradiction
	0	1	affirmation
	1	0	négation **NOT**
	1	1	tautologie

Il existe exactement 2^{2^k} fonctions booléennes d'arité k

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les seize opérateurs booléens binaires ($k = 2$)

$f(0,0)$	$f(0,1)$	$f(1,0)$	$f(1,1)$

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les seize opérateurs booléens binaires ($k = 2$)

$f(0,0)$	$f(0,1)$	$f(1,0)$	$f(1,1)$

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les seize opérateurs booléens binaires ($k = 2$)

$f(0,0)$	$f(0,1)$	$f(1,0)$	$f(1,1)$	
0	0	0	0	contradiction
0	0	0	1	conjonction **AND**
0	0	1	0	négation de l'implication $\not\supset$
0	0	1	1	affirmation de x
0	1	0	0	négation de l'implication inverse $\not\subset$
0	1	0	1	affirmation de y
0	1	1	0	ou eXclusif **XOR**
0	1	1	1	disjonction **OR**
1	0	0	0	négation connexe de Peirce **NOR**
1	0	0	1	équivalence **NXOR**
1	0	1	0	négation de y
1	0	1	1	implication inverse $\subset$
1	1	0	0	négation de x
1	1	0	1	implication $\supset$
1	1	1	0	incompatibilité de Shaffer **NAND**
1	1	1	1	tautologie

Connecteurs, opérateurs, fonctions

Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les seize opérateurs booléens binaires ($k = 2$)

$f(0,0)$	$f(0,1)$	$f(1,0)$	$f(1,1)$	
0	0	0	0	contradiction
0	0	0	1	conjonction **AND**
0	0	1	0	négation de l'implication $\not\supset$
0	0	1	1	affirmation de x
0	1	0	0	négation de l'implication inverse $\not\subset$
0	1	0	1	affirmation de y
0	1	1	0	ou eXclusif **XOR**
0	1	1	1	disjonction **OR**
1	0	0	0	négation connexe de Peirce **NOR**
1	0	0	1	équivalence **NXOR**
1	0	1	0	négation de y
1	0	1	1	implication inverse $\subset$
1	1	0	0	négation de x
1	1	0	1	implication $\supset$
1	1	1	0	incompatibilité de Shaffer **NAND**
1	1	1	1	tautologie

Connecteurs, opérateurs, fonctions

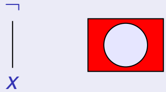
Une fonction booléenne est une fonction $f : \mathbb{B}^k \to \mathbb{B}$ où k est l'arité de f

Les seize opérateurs booléens binaires ($k = 2$)

	$f(0,0)$	$f(0,1)$	$f(1,0)$	$f(1,1)$	
	0	0	0	0	contradiction
	0	0	0	1	conjonction **AND**
	0	0	1	0	négation de l'implication $\not\supset$
	0	0	1	1	affirmation de x
	0	1	0	0	négation de l'implication inverse $\not\subset$
	0	1	0	1	affirmation de y
	0	1	1	0	ou eXclusif **XOR**
	0	1	1	1	disjonction **OR**
	1	0	0	0	négation connexe de Peirce **NOR**
	1	0	0	1	équivalence **NXOR**
	1	0	1	0	négation de y
	1	0	1	1	implication inverse $\subset$
	1	1	0	0	négation de x
	1	1	0	1	implication $\supset$
	1	1	1	0	incompatibilité de Shaffer **NAND**
	1	1	1	1	tautologie

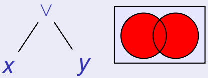
Négation

x	$\neg x$
0	1
1	0



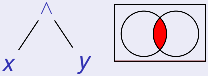
Disjonction

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1



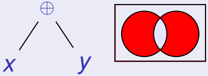
Conjonction

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1



Ou exclusif

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0



Connecteur de Peirce

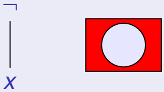
x	y	$x \downarrow y$
0	0	1
0	1	0
1	0	0
1	1	0

Incompatibilté de Sheffer

x	y	$x \uparrow y$
0	0	1
0	1	1
1	0	1
1	1	0

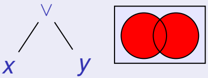
Négation

x	$\neg x$
0	1
1	0



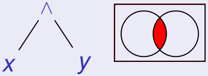
Disjonction

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1



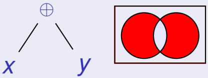
Conjonction

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1



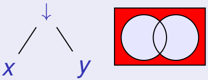
Ou exclusif

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0



Connecteur de Peirce

x	y	$x \downarrow y$
0	0	1
0	1	0
1	0	0
1	1	0

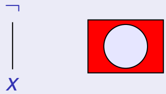


Incompatibilté de Sheffer

x	y	$x \uparrow y$
0	0	1
0	1	1
1	0	1
1	1	0

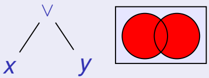
Négation

x	$\neg x$
0	1
1	0



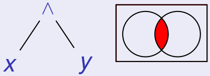
Disjonction

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1



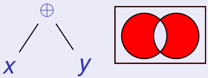
Conjonction

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1



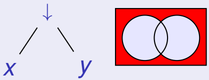
Ou exclusif

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0



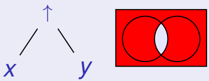
Connecteur de Peirce

x	y	$x \downarrow y$
0	0	1
0	1	0
1	0	0
1	1	0



Incompatibilté de Sheffer

x	y	$x \uparrow y$
0	0	1
0	1	1
1	0	1
1	1	0



Expressivité *vs* non-unicité

- les équivalences entre formules montrent qu'une même propriété peut s'exprimer de plusieurs manières différentes
- cette redondance est intéressante pour l'expressivité
  - expression claire des notions via un vocabulaire riche
- mais elle peut aussi être un frein au traitement informatique
  - comparaisons, cas multiples à considérer, ...
- on recherche des formes canoniques pour les formules propositionnelles

Forme normale de négation

- les lois de de Morgan permettent de propager $\neg$ vers les variables
- un littéral est une formule qui est soit une variable, soit sa négation
- toute formule propositionnelle est équivalente à une formule propositionnelle sans connecteur $\neg$ (sauf dans un littéral) et seulement des conjonctions et disjonctions

Expressivité *vs* non-unicité

- les équivalences entre formules montrent qu'une même propriété peut s'exprimer de plusieurs manières différentes
- cette redondance est intéressante pour l'expressivité
  - expression claire des notions via un vocabulaire riche
- mais elle peut aussi être un frein au traitement informatique
  - comparaisons, cas multiples à considérer, ...
- on recherche des formes canoniques pour les formules propositionnelles

Forme normale de négation

- les lois de de Morgan permettent de propager $\neg$ vers les variables
- un littéral est une formule qui est soit une variable, soit sa négation
- toute formule propositionnelle est équivalente à une formule propositionnelle sans connecteur $\neg$ (sauf dans un littéral) et seulement des conjonctions et disjonctions

Expressivité *vs* non-unicité

- les équivalences entre formules montrent qu'une même propriété peut s'exprimer de plusieurs manières différentes
- cette redondance est intéressante pour l'expressivité
  - expression claire des notions via un vocabulaire riche
- mais elle peut aussi être un frein au traitement informatique
  - comparaisons, cas multiples à considérer, ...
- on recherche des formes canoniques pour les formules propositionnelles

Forme normale de négation

- les lois de de Morgan permettent de propager $\neg$ vers les variables
- un littéral est une formule qui est soit une variable, soit sa négation
- toute formule propositionnelle est équivalente à une formule propositionnelle sans connecteur $\neg$ (sauf dans un littéral) et seulement des conjonctions et disjonctions

Expressivité *vs* non-unicité

- les équivalences entre formules montrent qu'une même propriété peut s'exprimer de plusieurs manières différentes
- cette redondance est intéressante pour l'expressivité
  - expression claire des notions via un vocabulaire riche
- mais elle peut aussi être un frein au traitement informatique
  - comparaisons, cas multiples à considérer, ...
- on recherche des formes canoniques pour les formules propositionnelles

Forme normale de négation

- les lois de de Morgan permettent de propager $\neg$ vers les variables
- un littéral est une formule qui est soit une variable, soit sa négation
- toute formule propositionnelle est équivalente à une formule propositionnelle sans connecteur $\neg$ (sauf dans un littéral) et seulement des conjonctions et disjonctions

Expressivité *vs* non-unicité

- les équivalences entre formules montrent qu'une même propriété peut s'exprimer de plusieurs manières différentes
- cette redondance est intéressante pour l'expressivité
  - expression claire des notions via un vocabulaire riche
- mais elle peut aussi être un frein au traitement informatique
  - comparaisons, cas multiples à considérer, ...
- on recherche des formes canoniques pour les formules propositionnelles

Forme normale de négation

- les lois de de Morgan permettent de propager $\neg$ vers les variables
- un littéral est une formule qui est soit une variable, soit sa négation
- toute formule propositionnelle est équivalente à une formule propositionnelle sans connecteur $\neg$ (sauf dans un littéral) et seulement des conjonctions et disjonctions

Expressivité *vs* non-unicité

- les équivalences entre formules montrent qu'une même propriété peut s'exprimer de plusieurs manières différentes
- cette redondance est intéressante pour l'expressivité
  - expression claire des notions via un vocabulaire riche
- mais elle peut aussi être un frein au traitement informatique
  - comparaisons, cas multiples à considérer, ...
- on recherche des formes canoniques pour les formules propositionnelles

Forme normale disjonctive

- un littéral est une formule qui est soit une variable, soit sa négation
- une clause conjonctive est soit la constante $\top$, soit un littéral, soit une conjonction d'au moins deux littéraux
- toute formule propositionnelle est équivalente à une formule qui est soit la constante $\bot$, soit une clause conjonctive, soit une disjonction d'au moins deux clauses conjonctives

Expressivité *vs* non-unicité

- les équivalences entre formules montrent qu'une même propriété peut s'exprimer de plusieurs manières différentes
- cette redondance est intéressante pour l'expressivité
  - expression claire des notions via un vocabulaire riche
- mais elle peut aussi être un frein au traitement informatique
  - comparaisons, cas multiples à considérer, ...
- on recherche des formes canoniques pour les formules propositionnelles

Forme normale disjonctive

- un littéral est une formule qui est soit une variable, soit sa négation
- une clause conjonctive est soit la constante $\top$, soit un littéral, soit une conjonction d'au moins deux littéraux
- toute formule propositionnelle est équivalente à une formule qui est soit la constante $\bot$, soit une clause conjonctive, soit une disjonction d'au moins deux clauses conjonctives

Expressivité *vs* non-unicité

- les équivalences entre formules montrent qu'une même propriété peut s'exprimer de plusieurs manières différentes
- cette redondance est intéressante pour l'expressivité
  - expression claire des notions via un vocabulaire riche
- mais elle peut aussi être un frein au traitement informatique
  - comparaisons, cas multiples à considérer, ...
- on recherche des formes canoniques pour les formules propositionnelles

Forme normale disjonctive

- un littéral est une formule qui est soit une variable, soit sa négation
- une clause conjonctive est soit la constante $\top$, soit un littéral, soit une conjonction d'au moins deux littéraux
- toute formule propositionnelle est équivalente à une formule qui est soit la constante $\bot$, soit une clause conjonctive, soit une disjonction d'au moins deux clauses conjonctives

Choix des connecteurs

- tous les connecteurs peuvent se déduire d'un nombre limité des autres
- différents choix sont possibles pour ces *connecteurs primitifs*
  et la simplicité des expressions dépend beaucoup de ce choix
  - dans les langages de programmation, on dispose généralement de
    - la *négation* $\neg$ (en Java : `!`),
    - la *conjonction* $\wedge$ (en Java : `&&`),
    - la *disjonction* $\vee$ (en Java : `||`),
    - et parfois le *ou exclusif* $\oplus$ (comme en C ou Java : `^`)
  - chacun des connecteurs NAND et NOR est universel

Circuit combinatoire

- un *circuit combinatoire* est une mise en œuvre matérielle d'une fonction combinatoire
- un système logique est dit *combinatoire* si l'état de sa sortie ne dépend que de l'état de son entrée

Choix des connecteurs

- tous les connecteurs peuvent se déduire d'un nombre limité des autres
- différents choix sont possibles pour ces *connecteurs primitifs*
  et la simplicité des expressions dépend beaucoup de ce choix
  - dans les langages de programmation, on dispose généralement de
    - la *négation* $\neg$ (en Java : `!`),
    - la *conjonction* $\wedge$ (en Java : `&&`),
    - la *disjonction* $\vee$ (en Java : `||`),
    - et parfois le *ou exclusif* $\oplus$ (comme en C ou Java : `^`)
  - chacun des connecteurs NAND et NOR est universel

Circuit combinatoire

- un *circuit combinatoire* est une mise en œuvre matérielle d'une fonction combinatoire
- un système logique est dit *combinatoire* si l'état de sa sortie ne dépend que de l'état de son entrée

Choix des connecteurs

- tous les connecteurs peuvent se déduire d'un nombre limité des autres
- différents choix sont possibles pour ces *connecteurs primitifs*
  et la simplicité des expressions dépend beaucoup de ce choix
  - dans les langages de programmation, on dispose généralement de
    - la *négation* $\neg$ (en Java : `!`),
    - la *conjonction* $\wedge$ (en Java : `&&`),
    - la *disjonction* $\vee$ (en Java : `||`),
    - et parfois le *ou exclusif* $\oplus$ (comme en C ou Java : `^`)
  - chacun des connecteurs NAND et NOR est universel

Circuit combinatoire

- un *circuit combinatoire* est une mise en œuvre matérielle d'une fonction combinatoire
- un système logique est dit *combinatoire* si l'état de sa sortie ne dépend que de l'état de son entrée

Choix des connecteurs

- tous les connecteurs peuvent se déduire d'un nombre limité des autres
- différents choix sont possibles pour ces *connecteurs primitifs* et la simplicité des expressions dépend beaucoup de ce choix
  - dans les langages de programmation, on dispose généralement de
    - la *négation* $\neg$ (en Java : `!`),
    - la *conjonction* $\wedge$ (en Java : `&&`),
    - la *disjonction* $\vee$ (en Java : `||`),
    - et parfois le *ou exclusif* $\oplus$ (comme en C ou Java : `^`)
  - chacun des connecteurs NAND et NOR est universel

Circuit combinatoire

- un *circuit combinatoire* est une mise en œuvre matérielle d'une fonction combinatoire
- un système logique est dit *combinatoire* si l'état de sa sortie ne dépend que de l'état de son entrée

Choix des connecteurs

- tous les connecteurs peuvent se déduire d'un nombre limité des autres
- différents choix sont possibles pour ces *connecteurs primitifs* et la simplicité des expressions dépend beaucoup de ce choix
  - dans les langages de programmation, on dispose généralement de
    - la *négation* $\neg$ (en Java : `!`),
    - la *conjonction* $\wedge$ (en Java : `&&`),
    - la *disjonction* $\vee$ (en Java : `||`),
    - et parfois le *ou exclusif* $\oplus$ (comme en C ou Java : `^`)
  - chacun des connecteurs NAND et NOR est universel

Circuit combinatoire

- un *circuit combinatoire* est une mise en œuvre matérielle d'une fonction combinatoire *(une parmi une infinité de possibilités)*
- un système logique est dit *combinatoire* si l'état de sa sortie ne dépend que de l'état de son entrée

Choix des connecteurs

- tous les connecteurs peuvent se déduire d'un nombre limité des autres
- différents choix sont possibles pour ces *connecteurs primitifs*
  et la simplicité des expressions dépend beaucoup de ce choix
  - dans les langages de programmation, on dispose généralement de
    - la *négation* $\neg$ (en Java : `!`),
    - la *conjonction* $\wedge$ (en Java : `&&`),
    - la *disjonction* $\vee$ (en Java : `||`),
    - et parfois le *ou exclusif* $\oplus$ (comme en C ou Java : `^`)
  - chacun des connecteurs NAND et NOR est universel

Circuit combinatoire

- un *circuit combinatoire* est une mise en œuvre matérielle d'une fonction combinatoire *(une parmi une infinité de possibilités)*
- un système logique est dit *combinatoire* si l'état de sa sortie ne dépend que de l'état de son entrée

Choix des connecteurs

- tous les connecteurs peuvent se déduire d'un nombre limité des autres
- différents choix sont possibles pour ces *connecteurs primitifs* et la simplicité des expressions dépend beaucoup de ce choix
  - dans les langages de programmation, on dispose généralement de
    - la *négation* $\neg$ (en Java : `!`),
    - la *conjonction* $\wedge$ (en Java : `&&`),
    - la *disjonction* $\vee$ (en Java : `||`),
    - et parfois le *ou exclusif* $\oplus$ (comme en C ou Java : `^`)
  - chacun des connecteurs NAND et NOR est universel

Circuit combinatoire

- un *circuit combinatoire* est une mise en œuvre matérielle d'une fonction combinatoire *(une parmi une infinité de possibilités)*
- un système logique est dit *combinatoire* si l'état de sa sortie ne dépend que de l'état de son entrée (pas de l'histoire du système)

Choix des connecteurs

- tous les connecteurs peuvent se déduire d'un nombre limité des autres
- différents choix sont possibles pour ces *connecteurs primitifs*
  et la simplicité des expressions dépend beaucoup de ce choix
  - dans les langages de programmation, on dispose généralement de
    - la *négation* $\neg$ (en Java : `!`),
    - la *conjonction* $\wedge$ (en Java : `&&`),
    - la *disjonction* $\vee$ (en Java : `||`),
    - et parfois le *ou exclusif* $\oplus$ (comme en C ou Java : `^`)
  - chacun des connecteurs NAND et NOR est universel

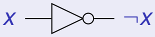
Circuit combinatoire

- un *circuit combinatoire* est une mise en œuvre matérielle d'une fonction combinatoire *(une parmi une infinité de possibilités)*
- un système logique est dit *combinatoire* si l'état de sa sortie ne dépend que de l'état de son entrée (pas de l'histoire du système)

https://logisim.altervista.org/

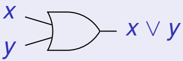
Négation — NOT

x	$\neg x$
0	1
1	0



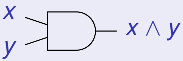
Disjonction — OR

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1



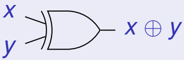
Conjonction — AND

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1



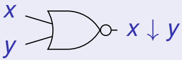
Ou exclusif — XOR

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0



Conn. de Peirce — NOR

x	y	$x \downarrow y$
0	0	1
0	1	0
1	0	0
1	1	0



Conn. de Sheffer — NAND

x	y	$x \uparrow y$
0	0	1
0	1	1
1	0	1
1	1	0