

Memo pour l'année

LAURENT Thomas

Master 2 informatique 2018

Contents

I	Fouille de donnée	1
1	Rappel sur les probabilités	2
2	Pré traitement des données	3
2.1	Nettoyage des données	3
2.1.1	Caractéristiques descriptives	3
2.2	Normalisation	3
3	Classification	5
3.1	Évaluation des classifieurs	5
3.1.1	Matrice de confusion	5
4	Arbre de décision	6
4.1	critères de sélection C4.5	6
4.1.1	Entropie	7
4.2	critères d'arrêt	8
4.2.1	Critères d'arrêt	8
4.2.2	critères d'arrêt: Paramètre utilisateur	9
5	Classificateur bayésiens	10
II	Apprentissage automatique par la pratique	13
6	Rappel	14
6.1	Matrices et calculs sur les Matrices	14
6.1.1	Addition	14
6.1.2	Multiplication	14
6.1.3	Transposer	14

6.1.4	Inverse	14
7	Algorithms Learn a Mapping From Input to Output	16
7.1	linear ML algorithms	16
7.2	Supervised machine learning	16
7.3	Unsupervised machine learning	16
7.4	semi-supervised machine learning	17
7.5	Overview of bias and variance	17
8	Overfitting and Underfitting	18
8.1	Overfitting	18
8.2	Underfitting	18
9	Linear Algorithms	19
9.1	Régression linéaire	19
9.2	Least squares linear regression	21
9.3	Gradient Descent	22
10	Logistic Regression	23
10.1	Logistic function	23
10.2	Linear Discriminant Analysis	23
10.2.1	la règle bayésienne	24
III	Outils formel	25
11	Logique classique des propositions	26
11.1	Vocabulaire	26
11.2	Propriétés de l'opérateur Models	27
11.3	Ensemble de connecteurs fonctionnellement complet	28
11.4	Preuve par induction structurelle sur un ensemble de connecteurs non fonctionnellement complet	28
11.5	Décomposition de Shannon	29
11.6	Arbre de Shannon, ROBDD	29
11.6.1	Remplacement ou vérifonctionnalité	30
11.6.2	Substitution	30
11.7	Notion de impliquant premier	30
11.7.1	Table de Karnaugh	30
11.7.2	Calcul arithmétique	31

11.8	Système de Hilbertien	31
11.9	théorème de finitude	31
12	Logique classique et prédicat du premier ordre	32
12.1	Syntaxe via les arbres	32
12.1.1	Occurrences libre	32
12.1.2	Occurrences liée	33
12.1.3	Occurrences quantifié	33
12.1.4	Vocabulaire	33
12.2	Sémantique	34
IV	Recherche Opérationnel	36
13	Rappel	37
13.1	Pivot de gauss	37
14	Introduction à la PL	38
14.1	Modèle linéaire continu à 2 variables	38
14.1.1	Recherche de solutions	39
14.1.2	recherche de la solution optimal	39
15	Le simplexe	41
15.1	Initialisation du simplexe	41
15.2	Canonicité du modèle	42
15.3	Premier itération	42
15.3.1	Choix de la variable entrante	42
15.3.2	Choix de la variable sortante	42
15.3.3	pivotage	43
15.3.4	Nouveau modèle	44
15.4	Seconde itération	44
15.4.1	Choix de la variable entrante	44
15.4.2	Choix de la variable sortante	44
15.4.3	pivotage	45
15.4.4	Nouveau modèle	46

V	Représentation des connaissances et raisonnement	47
16	Logique propositionnel	48
16.1	Vocabulaire	48
16.2	cohérence d'un ensemble de clauses	48
17	Introduction à la logique de description	50
17.1	Attributive Language with Complement	50
17.1.1	Sémantique	50
17.1.2	Propriétés	51
17.2	Logique de description	51
17.2.1	Sémantique	51
17.2.2	Assertions	51
17.3	TBoxes et ABoxes	52
17.3.1	Subsumption	52
17.3.2	Classification	52
17.3.3	Instance checking	53
17.3.4	Retrieval	53
17.3.5	Equivalence of concept	53
17.3.6	Concept satisfiability	53
17.3.7	ABox consistency	54
17.3.8	Réduction et consistance	54
18	Méthode des Tableau pour les ALC	55
18.1	Pre processing	55
18.1.1	Réécriture	55
18.1.2	Vocabulaire	56
18.1.3	Règles d'expansion	56
18.2	Exemple	57
18.3	Exemple 2	58
19	Logique presque tout	59
19.1	Système P	60
VI	XML	61
20	DTD	62

21 XSD	63
22 XPATH	65

Part I

Fouille de donnée

Chapter 1

Rappel sur les probabilités

Quelques rappels de probabilités : Soient X et Y deux variables aléatoires discrètes prenant leurs valeurs dans $DX=x_1, \dots, x_n$ et $DY=y_1, \dots, y_m$ respectivement.

$$P(x_i) = \frac{|x_i|}{\sum_{j=1}^n |x_j|}$$

$$\sum_{i=1}^n P(x_i) = 1$$

$$P(x_i|y_i) = \frac{P(x_i, y_i)}{p(y_i)}$$

$$P(x_i, y_i) = p(x_i) * p(y_i) \text{ Si X et Y sont indépendantes}$$

$$\textbf{r\`egle de chainage } P(x_1, x_2, x_3, \dots, x_n) = p(x_1) * p(x_2|x_1) * \dots * p(x_n|x_{n-1} \dots x_1)$$

$$\textbf{distribution conditionnel } \forall x \text{ in } X, \forall y \text{ in } Y \Rightarrow P(x|y)$$

Exemple:

$$: \begin{pmatrix} Anne & Sexe & \# & \% \\ M1 & M & 25 & 25/55 \\ M1 & F & 4 & 4/55 \\ M2 & M & 25 & 25/55 \\ M2 & F & 1 & 1/55 \end{pmatrix}$$

$$P(sexe = M) = P(Sexe = MetAnne = M1) + P(Sexe = MetAnne = M2) = 50/55$$

$$P(Arne = M2 | sexe = M) = P(Sexe = MetAnne = M2) / P(Sexe = M) = \frac{25}{55} / \frac{50}{55} = \frac{25}{50} = \frac{1}{2}$$

Chapter 2

Pré traitement des données

2.1 Nettoyage des données

2.1.1 Caractéristiques descriptives

Objectifs: Résumer, décrire certains aspects (tendances, variation, dispersion...) des données en utilisant certaines mesures :

Moyenne (espérance) : $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$

Ecart moyen : $\frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|$

Variance : $v = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$

Ecart type : $\sigma x := \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} = \sqrt{\frac{1}{n} (\sum_{i=1}^n x_i^2) - \bar{x}^2}$

Médiane : Valeur se trouvant au milieu d'une série de données ordonnées

Mode : Valeur la plus fréquente

Amplitude : min, max

2.2 Normalisation

Min-max : $v_n = \frac{v - v_{min}}{v_{max} - v_{min}}$

Min-max dans l'intervalle [A,B] : $v_n = \frac{v - v_{min}}{v_{max} - v_{min}} * (B - A) + A$

Z-Score : $v_n = \frac{v - \text{moyenne}}{\text{ecart}_{type}}$

Decimal scaling : $v_n = \frac{v}{100^j}$

Chapter 3

Classification

3.1 Évaluation des classifieurs

3.1.1 Matrice de confusion

Percent of correct classification :

$$\text{PCC}(\%) : = \frac{N_c}{N_t} * 100$$

N_c : nombre d'instances correctement classées

N_t : nombre d'instances testées ($N_t = |D_{test}|$)

Exemple:

$$: \begin{pmatrix} - & c1 & c2 & c3 & c4 \\ c1 & 0 & 1 & 0 & 0 \\ c2 & 1 & 60 & 0 & 1 \\ c3 & 0 & 1 & 23 & 0 \\ c4 & 1 & 0 & 7 & 5 \end{pmatrix}$$

Taux d'erreurs : 100-PCC

$$\text{PCC}(\%) = \frac{0+60+23+5}{100} * 100 = 88\%$$

Chapter 4

Arbre de décision

4.1 critères de sélection C4.5

Construction d'un arbre de décision C4.5 La construction d'un arbre de décision avec C4.5 passe par deux phases:

Phase d'expansion : La construction se fait selon l'approche descendante et laisse croître l'arbre jusqu'à sa taille maximale.

Phase d'élagage : Pour optimiser la taille l'arbre et son pouvoir de généralisation, C4.5 procède à l'élagage (pour supprimer les sous-arbres qui ne minimisent pas le taux d'erreurs)

Approche de construction d'un AD : Partitionner récursivement les données en sous-ensembles plus homogènes ... jusqu'à obtenir des partitions qui contiennent des objets qui appartiennent majoritairement à la même classe.

=> Théorie de l'information pour caractériser le degré de mélange, homogénéité, impureté, incertitude...

Théorie de l'information : Théorie mathématique ayant pour objet l'étude du contenu informationnel d'un message.
Applications en codage, compression, sécurité...

Entropie : Mesure la quantité d'incertitude dans une distribution de probabilités.

4.1.1 Entropie

Entropie : Mesure la quantité d'incertitude (manque d'information) dans une distribution de probabilités. Soit X une variable aléatoire discrète prenant ses valeurs dans $DX = x_1, \dots, x_n$. Soit P la distribution de probabilités associée à X .

$$H(X) = - \sum_{i=1}^n p(x_i) * \log_2(p(x_i))$$

Par convention, quand $p(x) = 0, 0 * \log(0) = 0$

Exemple:

X	P(X)
x_1	1/3
x_2	1/3
x_3	1/3

$$H(X) = -p(x_1) * \log_2(p(x_1)) - p(x_2) * \log_2(p(x_2)) - p(x_3) * \log_2(p(x_3))$$

$$H(X) = -3(\frac{1}{3} * \log_2(\frac{1}{3})) = \log_2(3) = 1.58$$

Autre exemples:

$$[\frac{1}{2}, \frac{1}{4}, \frac{1}{4}] : H(X) = 1.5$$

$$[1, 0, 0] : H(X) = 0$$

$$[\frac{1}{2}, \frac{1}{2}] : H(X) = 1$$

Propriétés:

$$H(X) \geq 0$$

$H(X)$ est maximale pour une distribution uniforme (toutes les valeurs sont équiprobables).

Entropie conjointe : L'entropie conjointe de deux variables aléatoires X et Y est l'incertitude relative à ces deux variables conjointement.

$$H(X, Y) = - \sum_{i,j=1}^n p(x_i, y_j) * \log_2(p(x_i, y_j))$$

Exemple : $[0.2, 0.1, 0.3, 0.4] : H(X, Y) = 1.85$

Critère de sélection: Gain d'information:

$$GAIN(T, A) = Info(T) - Info(T|A)$$

Avec $Info(T)$: Entropie au niveau de T (avant de partitionner)

$$Info(T) = - \sum_{c_i} freq(c_i, T) * \log_2(freq(c_i, T))$$

$$\textbf{Avec } freq(c_i, T) = p(c_i) = \frac{|c_i|}{|T|}$$

Avec $Info(T|A)$ l'entropie conditionnelle de T une fois partitionné selon les valeurs de l'attribut A.

$$Info(T|A) = \sum_{a_j \in A} freq(a_j, T) * Info(T|a_j)$$

Critère de sélection: Gain Ration:

Le gain d'information favorise les attributs ayant de larges domaines.

Le ratio de gain utilise le gain d'information avec un facteur pénalisant les attributs ayant des domaines trop larges.

$$GainRatio(T, A) = \frac{Gain(T, A)}{SplitInfo(T, A)}$$

Avec $SplitInfo(T, A) = - \sum_{a_j \in A} freq(a_j, T) * \log_2(freq(a_j, T))$ = Entropie de A.

4.2 critères d'arrêt

4.2.1 Critères d'arrêt

Si tout les objets d'une partition appartiennent à une même classes

Si il n'y a plus aucun attributs à tester

si le nœud est vide (càd feuille de l'arbre)

Absence d'apport informationnel (le grain est négatif ou nul)

4.2.2 critères d'arrêt: Paramètre utilisateur

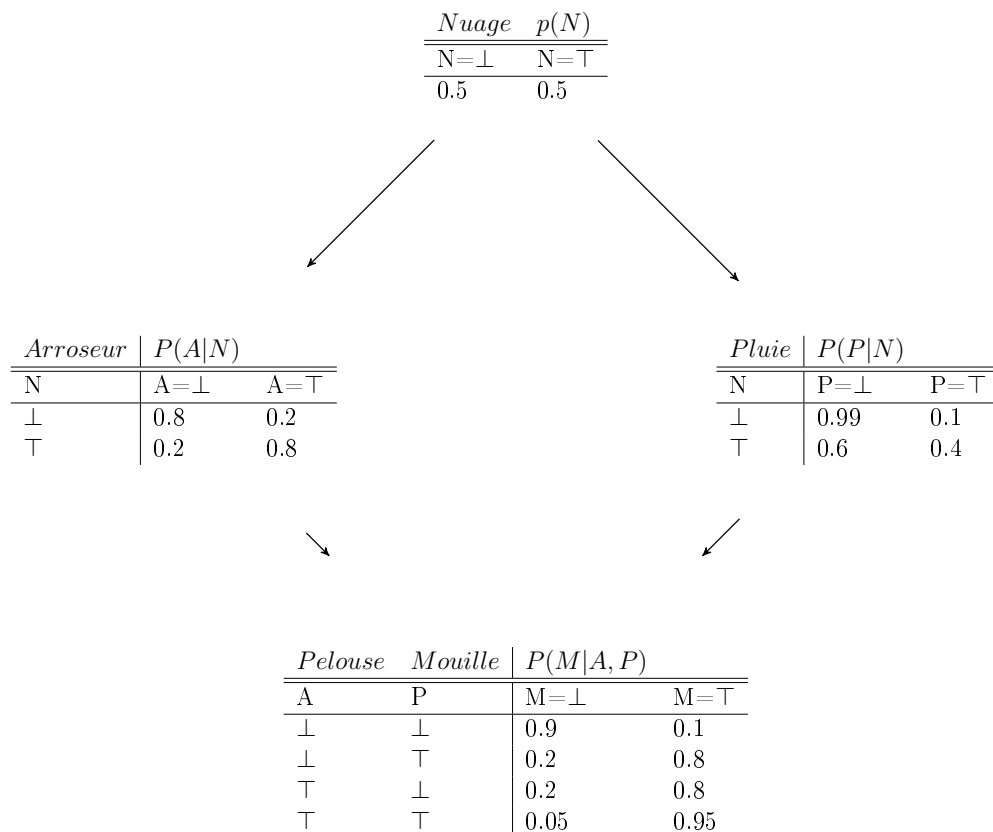
Nombre d'objets minimum par feuille

Taille, profondeur de l'arbre

Temps de construction de l'arbre

Chapter 5

Classificateur bayésiens



Calculer $P(N = \top, P = \top, A = \perp, M = \top)$

$$= P(N = \top) * P(P = \top | N = \top) * P(A = \perp | N = \top, P = \top) * P(M = \top | N = \top, P = \top, A = \perp)$$

$$= .5 * .4 * \frac{P(N=\top, P=\top)P(A=\perp)}{P(N=\top, P=\top)} * \frac{P(N=\top, P=\top, A=\perp)*P(M=\top)}{P(N=\top, P=\top, A=\perp)}$$

$$= .5 * .4 * 1 *$$

Part II

Apprentissage automatique par la pratique

Chapter 6

Rappel

6.1 Matrices et calculs sur les Matrices

6.1.1 Addition

$$\begin{pmatrix} 1 & 3 \\ 1 & 0 \\ 1 & 2 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 7 & 5 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 1+0 & 3+0 \\ 1+7 & 0+5 \\ 1+2 & 2+1 \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 8 & 5 \\ 3 & 3 \end{pmatrix}$$

6.1.2 Multiplication

$$\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$$
$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$$
$$(1 * 5) + (2 * 7) = 19$$

6.1.3 Transposer

$$\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

6.1.4 Inverse

Soit une matrice 2x2 comme : $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Soit Determinant $D = ad - bc$

Si $D \neq 0$ alors il existe une matrice inverse égal à : $\frac{1}{D} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

Chapter 7

Algorithms Learn a Mapping From Input to Output

7.1 linear ML algorithms

Simplifier les processus d'apprentissage et réduire la fonction sur ce qu'on connaît

Soit : $B_0 + B_1X_1 + B_2X_2 + B_3X_3 = 0$

Où B_0, B_1, B_2, B_3 sont les coefficients présent sur l'axe des ordonnées.

Et X_1, X_2, X_3 sont les valeurs en Input.

7.2 Supervised machine learning

L'apprentissage supervisé peut se diviser en 2 partis

Classification : Quand les variables en sortie sont des Classe (*Vert, Carr, Homme*)

Regression : Quand les variables en sortie sont des valeur numérique (*euro, poids, quantités*)

7.3 Unsupervised machine learning

Les problèmes de l'apprentissage non supervisé sont:

Clustering : L'art de faire des paquet d'éléments qui ont des points commun, comme regrouper les clients par paquet de choses qu'ils ont le plus en commun.

Association : Associer des règles d'apprentissage pour décrire une portion du data, comme une personne qui a acheté un item A et qui est aussi tenté par acheter un item B

7.4 semi-supervised machine leaning

L'apprentissage semi supervisé c'est avoir un bonne quantité de données en input X, et un peu de data avec le label Y.

7.5 Overview of bias and variance

La prédiction des erreurs pour les algorithmes sont regroupé en 3 points:

Bias Error : Simplifier l'hypothèse fait par le modèles pour faire une fonction d'apprentissage plus facile.

Variance Error : Et la quantité estimé par la fonction visé qui changera via un différent ensemble de data utilisé.

Irreducible Error : Ne peut pas être réduit

Chapter 8

Overfitting and Underfitting

8.1 Overfitting

L'overfitting intervient lorsque le modèle sur apprend des connaissances, Lorsque l'on sur apprend nous prenons en compte les points plus éloigné de la droite de la fonction.

On peut illustrer l'overfitting en codant un algorithme qui prend en compte les points bleu et rouges de la figure *ap-linear-regression_1* ce dessous.

8.2 Underfitting

C'est l'inverse de l'overfitting, pas assez de données pour pouvoir généraliser le base de connaissance.

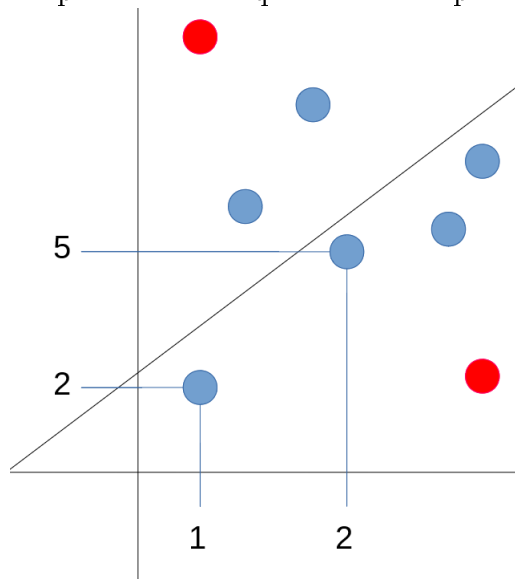
Chapter 9

Linear Algorithms

Soit X l'ensemble des variables indépendantes sur l'axe des l'abscisse et Y l'ensemble des variable dépendantes sur l'axe des ordonnée.

9.1 Régression linéaire

Étant donné un plan à deux dimensions où l'abscisse contient les point d'entrée X et l'ordonnée contient les points de sortie Y , et un nouage de points précédait acquitté de tout point éloigné du nuage.



Figureap – linear – regression₁

Avec : $y = \beta_0 + \beta_1 x$

Pour un hyperPlan (3d) : $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

$P - I_n$: $y = \beta_0 + \beta_1 x_1 + \dots \beta_n x_n$

Exemple:

$$\mathbf{5} = \beta_0 + 2 * \beta_1$$

$$\mathbf{2} = \beta_0 + 1 * \beta_1$$

9.2 Least squares linear regression

Calculer la régression linéaire avec la méthode Least squares:

Soit:

$\mathbf{X} = [1, 2, 3, 4, 5]$ les variables indépendantes d'axe abscisse

$\mathbf{Y} = [2, 4, 5, 4, 5]$ les variables dépendantes d'axe ordonnée

Calculons $y = \beta_0 + \beta_1 x$

Calcule de la moyenne de X et Y:

$$\mathbf{Xm} = \sum x_i \in X = 3$$

$$\mathbf{Ym} = \sum y_i \in Y = 4$$

Toutes ligne de régression doivent passer par le point (Xm,Ym).

Calculer tout les écarts des $x_i \in X$ par rapport à Xm (resp Y):

X	Y	$X - Xm$	$Y - Ym$	$(X - Xm)^2$	$(X - Xm)(Y - Ym)$
1	2	-2	-2	4	4
2	4	-1	0	1	0
3	5	0	1	0	0
4	4	1	0	1	0
5	5	2	1	4	2

Calculer β_1 :

$$\beta_1 = \frac{\sum (X - Xm)(Y - Ym)}{\sum (X - Xm)^2} = \frac{6}{10} = .6$$

$$\beta_0 : Ym = \beta_0 + \beta_1 * Xm : 4 = \beta_0 + .6 * 3 : 4 = \beta_0 + 1.8 : \beta_0 = 2.2$$

9.3 Gradient Descent

Soit:

$$\mathbf{X} = [1, 2, 4, 3, 5]$$

$$\mathbf{Y} = [1, 3, 3, 2, 5]$$

i = une variable qui itère les éléments de X et Y en bouclant à l'infini.

Une initialisation comme:

$$\beta_0 = 0$$

$$\beta_1 = 0$$

α = donnée en énoncé (pour l'exemple égal à 0.01)

Et des fonctions définit tel que:

$$\mathbf{error} = (\beta_0 + \beta_1 * X[i]) - Y[i]$$

$$\beta_{0+1} = \beta_0 - \alpha * error$$

$$\beta_{1+1} = \beta_1 - \alpha * error * X[i]$$

En appliquant l'algorithme des calculs des β_i :

i	$X[i]$	$Y[i]$	$error$	β_0	β_1
0	1	1	-1	0.01	0.01
1	2	3	-2.97	0.06	0.03
2	4	3	-1.77	0.18	0.06
3	3	2	-1.61	0.22	0.08
4	5	5	-4.35	0.44	0.12
0	1	1	-0.42	0.45	0.13
1	2	3	-2.28	0.49	0.49

Chapter 10

Logistic Regression

10.1 Logistic function

Soit:

$$t \in \mathbb{R}[0, 1] \text{ égal à } \beta_0 + \beta_2 * x$$

La fonction de logique de régression, les valeur d'entrée X sont combiné en utilisant les coefficient de valeur pour prédire une sortie Y . Cette sortie sera une valeur binaire.

$$p(x) = \frac{1}{1+e^{-(\beta_0 + \beta_1 x)}}$$

Note : $p(x)$ peut être interprété comme une fonction de probabilité $P(X) = P[Y = 1|X]$.

$$\beta_0 + \beta_1 * x = \ln\left(\frac{P(x)}{1-P(x)}\right) \text{ aussi appelé odds.}$$

10.2 Linear Discriminant Analysis

L'analyse discriminante linéaire fait partie des techniques d'analyse discriminante prédictive, il s'agit de prédire l'appartenance d'un individu à une classe prédéfinie à partir de ses caractéristiques mesurées à l'aide de variables prédictives.

A notre disposition, un échantillon de n observations réparties dans k groupes d'effectifs n_k .

Noté Y les variables prédire $\{y_1, \dots, y_k\}$

J variables prédictives $X = (X_1, \dots, X_j)$

μ_k la moyenne (ou *mean* en anglais) valant $\lambda(list) - > \frac{\sum list[i]}{taille(list)}$

σ^2 la variance de toutes les classes $\frac{\sum_{i=1}^n (x_i - \mu_k)^2}{n - k}$

la fonction discriminante pour la classe k avec x donné $D_k(x) = x * \frac{\mu_k}{\omega^2} - \frac{\mu_k^2}{2x\omega^2} + \ln(P(k))$

Où $P(k)$ vaut la probabilité appliqué aux valeurs de Y

10.2.1 la règles bayésienne

L'objectif est de produire une règle d'affectation $X(\omega) \rightarrow Y(\omega)$ qui permet de prédire, pour une observation ω donné, sa valeur associé de Y à partir des valeurs prises par X . via une probabilité

$$P(Y = y_k) = \frac{P(Y=y_k) * P(X|Y=y_k)}{\sum_{i=1}^k P(Y=y_i) * P(X|Y=y_i)}$$

Où $P(Y = y_k)$ est la probabilité à *priori* d'appartenance à une classe

$P(X|Y = y_k)$ représente la fonction de densité des X conditionnellement à la classe y_k

Part III

Outils formel

Chapter 11

Logique classique des propositions

11.1 Vocabulaire

Déduction $\models \alpha$ ssi $\neg\alpha$ est contradictoire

Absurde ϕ est contradictoire ssi $\neg\phi$ est valide

DAG : Un graphe dirigé acyclique

Taille(Arbre) = $\{\text{tout les symboles} + \text{connecteurs}\}$

Var(Arbre) = $\{\text{Toutes les feuilles}\}$

Sous formules(Arbres) = $\{T + \cup_{i=0}^k \text{SousFormules}(\text{Arbre}_i)\}$

Interprétation : ω de $PROP_{ps}$ est une application de PS dans 0.1

Sémantique : $\|\phi\|(\omega)$ d'une formule ϕ de $PROP_{ps}$ dans l'interprétation ω est un élément de 0.1 défini inductivement par:

si $\phi \in PS$ alors $\|\phi\|(\omega) = \omega(\phi)$

si $\phi = cX_1 \dots X_n$ alors $\|\phi\|(\omega) = C_F(\|x_1\|(\omega) \dots \|x_n\|(\omega))$

ω **satisfait** ϕ noté $\omega \models \phi$ ssi $\|\phi\|(\omega) = 1$

Lorsque $\omega \models \phi$ on dit que ω est un modèle de ϕ

on note $\eta(\phi)$ l'ensemble des modèles de ϕ

$\omega \in PROP_{ps}$ est **valide** noté $\models \phi$, ssi toute interprétation ω de $PROP_{ps}$ satisfait ϕ

$\phi \equiv \psi$ sont logiquement équivalents ssi $\phi \models \psi$ et $\psi \models \phi$

11.2 Propriétés de l'opérateur Models

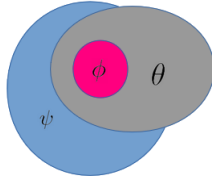
$a \models b \iff M(a) \subseteq M(b)$

Réflexivité : $\phi \models \phi$

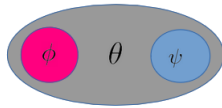
Équivalence à gauche : si $\phi \equiv \theta$ et $\phi \models \psi$ alors $\theta \models \psi$

Affaiblissement à droite (transitivité) : si $\phi \models \psi$ et $\psi \models \theta$ alors $\phi \models \theta$

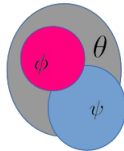
Coupure : si $\phi \wedge \psi \models \theta$ et $\phi \models \psi$ alors $\phi \models \theta$: $\iff (A \cup B) \subseteq C$ ssi $A \subseteq C \cap B \subseteq C$



Ou : $\phi \vee \psi \models \theta$ ssi $\phi \models \theta$ et $\psi \models \theta$



Monotonie : si $\phi \models \theta$ alors $\phi \wedge \psi \models \theta$



11.3 Ensemble de connecteurs fonctionnellement complet

On dit qu'un ensemble est fonctionnellement complet si avec que les connecteurs de cette ensemble on peut exprimer toutes les formules d'un monde.

$\{\neg, \wedge\}$ est fonctionnellement complet pour la logique propositionnel classique

Il en va de même pour $\{\neg, \vee\}, \{vrai, \wedge, \oplus\}, \{\neg, \Rightarrow\}$ ou $\{NAND\}$

Suppression des fils équivalent : Soit un arbre D ayant comme sous arbre plus d'une fois le nœud $\alpha = (\top X \top)$, α peut être remplacé par (\top) tout en concevant les modèles de D.

fusion des nœuds : Soit un arbre D ayant comme sous arbre les nœuds (aBc) et $(a'B'c')$ et $a = a', b = b', c = c'$ alors on peut faire relier les deux branches menant vers ces nœuds vers le même sous arbre.

11.4 Preuve par induction structurelle sur un ensemble de connecteurs non fonctionnellement complet

Soit $\forall P \in \{\wedge, \vee\}_{ps}$, vérifier P:

Cas de base $\varphi \in PS$: $1 \rightarrow (\varphi) = 1$ donc $1 \rightarrow$ constitue un modèle de φ

Étape inductive :

φ s'écrit : $[\alpha \wedge \beta]$ ou $[\alpha \vee \beta]$

Avec $\alpha, \beta \in \{\wedge, \vee\}_{ps}$

Par hypothèse d'induction, α et β vérifient P.

Il ne reste plus qu'à montrer que φ vérifie P.

$$\|\alpha \vee \beta\|(1 \rightarrow) = \vee \models (\|\alpha\|(1 \rightarrow), \|\beta\|(1 \rightarrow)) = \vee \models (1, 1) = 1$$

$$\|\alpha \wedge \beta\|(1 \rightarrow) = \wedge \models (\|\alpha\|(1 \rightarrow), \|\beta\|(1 \rightarrow)) = \wedge \models (1, 1) = 1$$

donc $x \wedge \neg x$ ne vérifie pas P : $\|x \wedge \neg x\|(1 \rightarrow) = 0$

11.5 Décomposition de Shannon

On note $\phi[x \leftarrow 0]$ la formule obtenue en substituant dans ϕ la constante faux à toutes les occurrences du symbole propositionnel x .

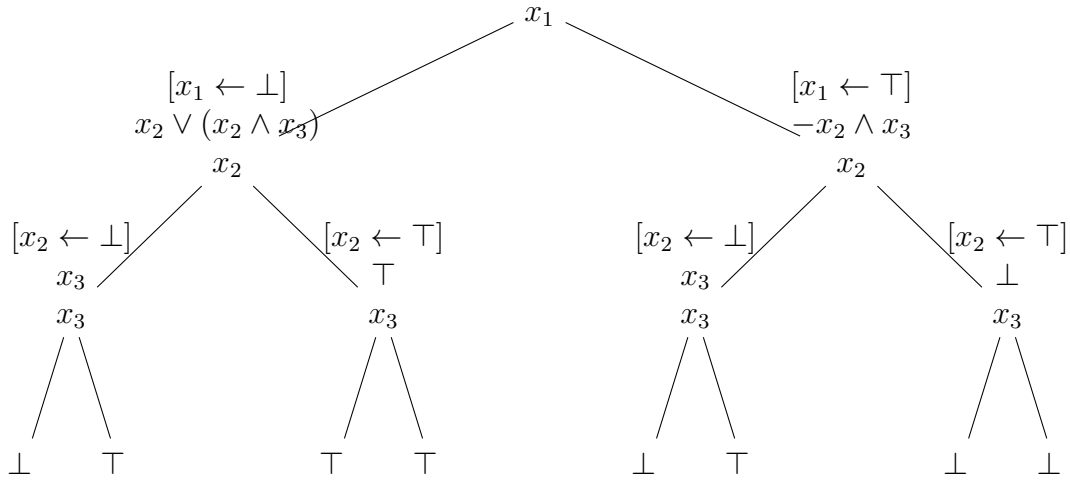
On note $\phi[x \leftarrow 1]$ la formule obtenue en substituant dans ϕ la constante vrai à toutes les occurrences du symbole propositionnel x .

La décomposition de Shannon de ϕ suivant x est la formule:

$$(\neg x \wedge \phi[x \leftarrow 0]) \vee (x \wedge \phi[x \leftarrow 1])$$

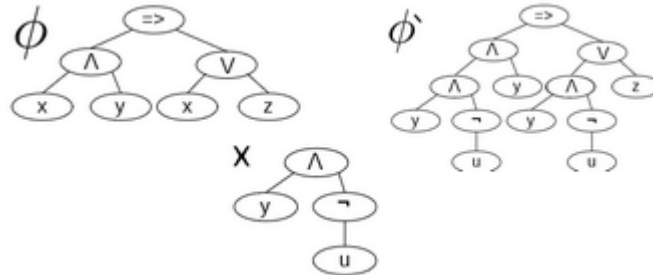
11.6 Arbre de Shannon, ROBDD

Étant donnée un ordre strict total $x_1 < x_2 < x_3$ sur $Var(\phi) = \{x_1, \dots, x_n\}$
Et une formule $\phi = (\neg x_1 \wedge x_2) \vee (\neg x_2 \wedge x_3)$



L'ensemble des modèles de ϕ sont toutes les interprétation où la feuille vaut la valeur T .

11.6.1 Remplacement ou vérifonctionnalité



$\phi \equiv \phi'$ quelque soit la valeur de x (vrai ou faux).

11.6.2 Substitution

Soit un arbre D ayant comme nœud un sous arbre du type infixe $\alpha = (x \Rightarrow y)$ et un sous arbre de substitution $\beta = (\neg x \Rightarrow \neg y)$
 $(D' = D_{\alpha \leftarrow \beta} \equiv D)$

11.7 Notion de impliquant premier

Les impliquant premier sont des sous formules des formules original tel que ces sous formules soit plus petite que la formule d'origine elle conserve les même modèles:

En circuit combinatoire les algo sont appelé Table de Karnaugh ou Quine-McCluskey.

11.7.1 Table de Karnaugh

Appliquer l'algorithme avec la formule $S = \neg a \neg b \neg c d + a \neg b \neg c \neg d + b \neg d$

S	$\neg a \neg b$	$\neg ab$	ab	$a \neg b$
$\neg c \neg d$	X	X	X	X
$\neg cd$		X	X	
cd		X	X	
$c \neg d$	X	X	X	X

les impliquant premier de S sont $b \neg d$

11.7.2 Calcule arithmétique

En logique, les impliquant premier sont calculer que à partir d'une formule en mode CNF transposé en DNF et ensuite détransposé en CNF.

$$\phi = (a \wedge b \wedge c) \vee (\neg b \wedge c)$$

$$\phi = (a \vee \neg b) \wedge (a \vee c) \wedge (b \vee \neg b) \wedge (b \vee c) \wedge (c \vee \neg b) \wedge (c \vee c)$$

$$\phi = (a \vee \neg b) \wedge (a \vee c) \wedge (b \vee c) \wedge (c \vee \neg b) \wedge c$$

$$\phi = (a \vee \neg b) \wedge c$$

$$\phi = (a \wedge c) \vee (\neg b \wedge c) \text{ sont les impliquant premier.}$$

Via une table de Karnaugh:

ϕ	$\neg a \neg b$	$\neg ab$	ab	$a \neg b$
$\neg c$				
c	X		X	X
Égal à $(a \wedge c) \vee (\neg b \wedge c)$.				

11.8 Système de Hilbertien

g

11.9 théorème de finitude

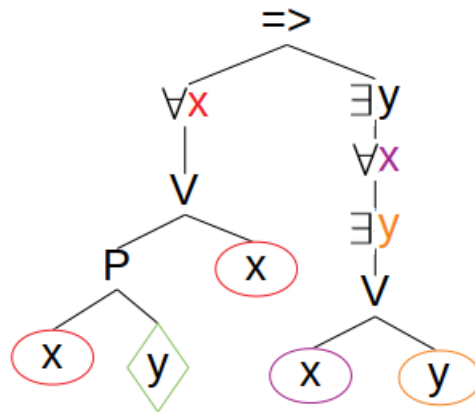
g

Chapter 12

Logique classique et prédicat du premier ordre

12.1 Syntaxe via les arbres

$\phi =$



12.1.1 Occurrences libre

Une occurrence libre est une variable n'ayant aucun quantificateur associé de son noeud à la racine de l'arbre.

par exemple le noeud y ayant un contour un losange vert est une

occurrence libre, elle sera instancié que lors de l'interprétation de ϕ .

12.1.2 Occurrences liée

Une occurrence liée est une variable ayant un quantificateur associé, comme:

la variable x entouré d'un rond rouge est définit via le quantificateur $\forall x$ présent dans ces noeuds parent

la variable x entouré d'un rond violet est définit par le quantificateur de ces parents $\forall x$

la variable y entouré d'un rond orange via le quantificateur $\exists y$

A noté que les x entouré d'un rond de couleurs rouge sont diffèrent des x entouré avec un rond orange, donc on peut tout bien renommer les x de couleur orangé en z sans changer le sens de ϕ .

Les occurrences liée se lient sur leur premier père le définissant, comme le y orange qui se définit que sur le $\exists y$ le plus proche de lui.

12.1.3 Occurrences quantifié

Les occurrences quantifié sont toutes les variable positionné derrière un quantificateur, celle ci montre comme dans la logique classique, le \forall (où quelque soit) ou \exists (où il existe au moins un).

On peut noter que sur la figure ci dessus il y a un $\exists y$ qui n'est pas associé à un y en feuille, on peut s'en débarrasser sans changer le sens de ϕ .

12.1.4 Vocabulaire

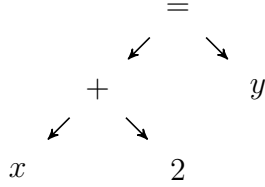
Formule fermée est une formule de $FORM_L$ qui ne contient aucune variable libre.

Formule instanciée est une formule qui ne contient aucune occurrence libre ou liée de symbole de variable

12.2 Sémantique

Soit t un terme de $TERM_L$, la sémantique de t dans l'interprétation de I pour l'assignation X_i noté $\llbracket t \rrbracket(I)(X_i)$ est l'élément de D_i défini inductivement.

$\phi =$



$= \in \Re$ d'arriter 2

$+ \in \Im$ d'arriter 2

$2 \in \Im$ d'arriter 0

$X, Y \in X$

Avec une interprétation tel que:

$D_i = \mathbb{N}$

$+_1 = \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$

$2_i = 3$

Avec une assignation tel que:

$X_i : X \rightarrow \mathbb{N}$

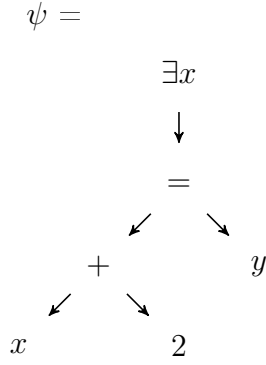
$x \rightarrow 5$

$y \rightarrow 10$

On peut calculer cette sous formule en appliquant chaque terme dans l'interprétation I pour un assignent X_i :

$\llbracket x + 2 \rrbracket(I)(X_i) = +_i(\llbracket x \rrbracket(I)(X_i), \llbracket 2 \rrbracket(I)(X_i)) = +_i(5, 3) = 8$

$\llbracket \phi \rrbracket(I)(X_i) = =_i(8, 10) = 0(faux)$



$$\begin{aligned}
 & \|\psi\|(I)(X_i)[x \leftarrow 7]) = \\
 =_i & (+_i(\|x\|(I)(X_i[x \leftarrow 7]), 3), \|y\|(I)(X_i[x \leftarrow 7])) = \\
 =_i & (+_i(7, 3), 10) = \\
 =_i & (10, 10) = 1(vrai)
 \end{aligned}$$

Le quantificateur \forall ou \exists est plus prioritaire que les variables assigné dans X_i .

Soit ϕ la formule ϕ ci dessus, la formule interprété avec deux assignations différentes:

$$X_i^1 \quad x \rightarrow 5, y \rightarrow 10$$

$$X_i^2 \quad x \rightarrow 6, y \rightarrow 10$$

L'interprétation de ϕ avec X_i^1 est équivalent à ϕ avec X_i^2 car le symbole de quantification \exists est plus prioritaire que les assignations.

Part IV

Recherche Opérationnel

Chapter 13

Rappel

13.1 Pivot de gauss

$$L1 \text{ et } L2 = \begin{cases} L1 : 160 = 8x + 4y \\ L2 : 120 = 4x + 6y \end{cases}$$

$$(L2 * (-2)) = \begin{cases} L1 : 160 = 8x + 4y \\ L2 : -240 = -8x - 12y \end{cases}$$

$$(L2 = L2 + L1) = \begin{cases} L1 : 160 = 8x + 4y \\ L2 : -80 = -8y \end{cases}$$

$$y = 10$$

$$8x + 4 * 10 = 160$$

$$8x + 40 = 160$$

$$8x = 120$$

$$x = 15$$

Chapter 14

Introduction à la PL

Construire une modèle linéaire, c'est donc:

identifier les variables de décision du problème

déterminer : la fonction objectif du modèle

déterminer : les contraintes du modèle

14.1 Modèle linéaire continu à 2 variables

Soit le modèle linéaire suivantes:

Déterminer $(x, y) \in \mathfrak{S}^2$

Minimisant $z = 1000x + 1200y$

sous les contraintes :

$$(1) 8x + 4y \leq 160$$

$$(2) 4x + 6y \leq 120$$

$$(3) x \leq 34$$

$$(4) y \leq 14$$

$$(5) 0 \leq x$$

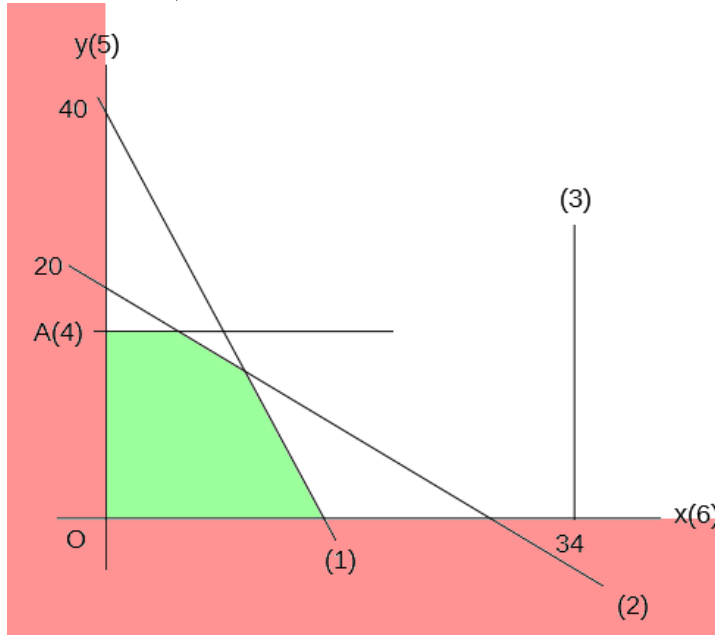
$$(6) 0 \leq y$$

14.1.1 Recherche de solutions

Après avoir tracé graphiquement tout les points:

Pour chaque contrainte, tracer la droite et repérer le demi plan des solution: exemple pour (5) et (6), x et y doivent être supérieurs ou égal à 0, d'où le demi plan des solution sont toutes les valeurs positives.

La partie En vert représente la région admissible, quelque soit le point choisis dans ce vert, aucune contrainte ne sera violé.



14.1.2 recherche de la solution optimal

Changer l'équation z tel que z soit égal à 0

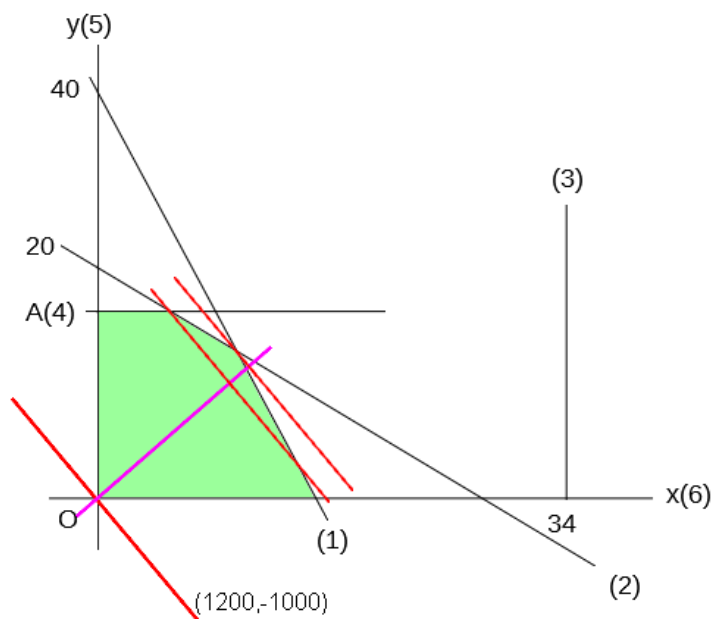
$$z = 1000x + 1200y = 0 = 1000 * (1200) + 1200 * (-1000)$$

Traçons la droite $(0, 0)$, $(1200, -1000)$

Un point extrême : est un point se trouvant sur l'intersection de 2 contraintes et étant dans la zone admissible.

L'altitude : est la droite (rouge) la plus haute touchant un point extrême, ce point sera le vecteur (x, y) le plus optimal pour z .

Les droites rouges doivent être toutes parallèles.



Dans cet exemple le point $(15,10)$ est le point extrême maximal pour l'équation z .

Chapter 15

Le simplexe

Soit le modèle linéaire suivantes:

Déterminer $(x, y) \in \mathfrak{S}^2$

Maximisant $Z = 3x + 7y$

sous les contraintes :

$$(1) -x + y \leq 3$$

$$(2) y \leq 8$$

$$(3) 2x - y \leq 28$$

$$(5) 0 \leq x$$

$$(6) 0 \leq y$$

15.1 Initialisation du simplexe

Pour chaque expression du type (1)(2)(3) intégrer un e_i pour la transformer en équation.

On appelle les e_1 des variables d'accumulation, Ce qui fait

Déterminer $(x, y, e_1, e_2, e_3) \in \mathfrak{S}^5$

Maximisant $Z = 3x + 7y$

sous les contraintes :

$$(1) -x + y + e_1 = 3$$

$$(2) y + e_2 = 8$$

$$(3) 2x - y + e_3 = 28$$

$$(5) 0 \leq x$$

$$(6) 0 \leq y$$

$$(7) e_1, e_2, e_3 \geq 0$$

15.2 Canonicité du modèle

Soit les valeurs (pour la première itération)

Hors Base (x, y)

Base (e_1, e_2, e_3)

Un modèle est canonique que si:

si toutes les variables de Base ne sont pas dans Z .

15.3 Premier itération

15.3.1 Choix de la variable entrante

(x, y) sont deux choix possible, le tout est de choisir une bonne heuristique, comme celle du meilleur gain marginale, ou via la comparaison (en mode graphique):

Y sera choisit, donc Y sera notre variable entrante.

15.3.2 Choix de la variable sortante

Pour chaque résultat d'équation, le diviser par sa valeur de Y (devant être positif (car Y est la variable entrante))

$$-x + y + e_1 = 3 \text{ donne } \frac{3}{1} = 3 \text{ (1 car } y = 1 * y)$$

$$y + e_2 = 8 \text{ donne } \frac{8}{1} = 8$$

$$2x - y + e_3 = 28 \text{ donne } \frac{28}{1} = 28$$

Prendre le minimum des variables, donc se sera 3.
la variable présente dans la Base sera prise comme variable sortante, dans notre cas e_1 .

15.3.3 pivotage

On choisit l'équation associée à la variable e_1 pour définir la variable entrante y .

On n'a:

$$y = x - e_1 + 3$$

Puis on crée les nouvelles équations via le nouveau y :

$$Z = 3x + 7y \text{ devient}$$

$$Z = 3x + 7(x - e_1 + 3)$$

$$Z = 10x - 7e_1 + 27$$

$$x - e_1 = 3 \text{ est déjà normalisé}$$

$$y + e_2 = 8 \text{ devient}$$

$$8 = x - e_1 + 3 + e_2$$

$$5 = x - e_1 + e_2$$

$$2x - y + e_3 = 28 \text{ devient}$$

$$28 = 2x + (x - e_1 + 3) + e_3$$

$$25 = 3x - e_1 + e_3$$

15.3.4 Nouveau modèle

Après cette étape nous voilà avec un nouveau modèle:

Déterminer $(x, y, e_1, e_2, e_3) \in \mathfrak{S}^5$

Maximisant $Z = 10x - 7e_1 + 21$

sous les contraintes :

$$(1) -x + y + e_1 = 3$$

$$(2) x - e_1 + e_2 = 5$$

$$(3) 3x - e_1 + e_3 = 25$$

$$(5) 0 \leq x$$

$$(6) 0 \leq y$$

$$(7) e_1, e_2, e_3 \geq 0$$

Variable hors base $= x, e_1$

Variable Base $= y, e_2, e_3$

Avec comme solution admissible $(0, 3, 0, 5, 25)Z = 21$

A ne pas oublier de vérifier la canonicité du modèle.

15.4 Seconde itération

15.4.1 Choix de la variable entrante

X sera choisit, donc X sera notre variable entrante.

15.4.2 Choix de la variable sortante

$$\frac{5}{1} = 5$$

$$\frac{25}{3} = 8.3$$

Prendre le minimum des variables, donc se sera 5, donc e_2 .

15.4.3 pivotage

$$x = e_1 - e_2 + 5$$

Puis on crée les nouvelles équations via le nouveau y :

$$Z = 10x - 7e_1 + 27 \text{ devient}$$

$$Z = 10(e_1 - e_2 + 5) - 7e_1 + 27$$

$$Z = 3e_1 - 10e_2 + 71$$

$$-x + y + e_1 = 3 \text{ devient}$$

$$3 = -(e_1 - e_2 + 5) + y + e_1$$

$$8 = y + e_2$$

$$3x - e_1 + e_3 = 25 \text{ devient}$$

$$25 = 3(e_1 - e_2 + 5) - e_1 + e_3$$

$$10 = 2e_1 - 3e_2 + e_3$$

15.4.4 Nouveau modèle

Après cette étape nous voilà avec un nouveau modèle:

Déterminer $(x, y, e_1, e_2, e_3) \in \mathfrak{S}^5$

Maximisant $Z = 3e_1 - 10x + 71$

sous les contraintes :

$$(1) \ y + e_2 = 8$$

$$(2) \ x - e_1 + e_2 = 5$$

$$(3) \ 2e_1 - 3e_2 + e_3 = 10$$

$$(5) \ 0 \leq x$$

$$(6) \ 0 \leq y$$

$$(7) \ e_1, e_2, e_3 \geq 0$$

Variable hors base $= e_2, e_1$

Variable Base $= y, x, e_3$

Avec comme solution admissible $(5, 8, 0, 0, 10)Z = 71$

A ne pas oublier de vérifier la canonicité du modèle.

Part V

Représentation des connaissances et raisonnement

Chapter 16

Logique propositionnel

16.1 Vocabulaire

Les *Logiques propositionnel* sont définis via les symboles suivant:

$\top, \perp, C, \neg C, C \wedge C, C \vee C, C \Rightarrow C$

Littéral est un atome ou la négation d'un atome

Clause est une disjonction de littéraux

Cube est une conjonction de littéraux

CNF est une forme normal conjonctive (une conjonction de clauses)

DNF est une forme normal disjonctive (une disjonction de cubes)

16.2 cohérence d'un ensemble de clauses

Soit K un ensemble de clauses pouvant être réduit via les axiomes:

$$x \vee x \vee y_1 \vee \dots y_n \equiv x \vee y_1 \vee \dots y_n$$

$$x \vee \neg x \vee y_1 \vee \dots y_n \equiv \text{'top'}$$

$$x \vee \top \equiv \top$$

$$x \vee \perp \equiv x$$

Si K est vide alors K est cohérente

Si $\perp \in K$ alors K est incohérente

$K_{x \leftarrow \top}$ est le résultat du remplacement des occurrences de x par \top

$K_{x \leftarrow \perp}$ est le résultat du remplacement des occurrences de x par \perp

Chapter 17

Introduction à la logique de description

17.1 Attributive Language with Complement

Les *ALC* sont définis via les symboles suivants :

$\top, \perp, C, \neg C, C \sqcap C, C \sqcup C, \forall r.C, \exists r.C$

17.1.1 Sémantique

Tuple $\iota =_{def} \langle \delta^I, .^I \rangle$ où

δ^I est le domaine (ou un ensemble d'objets)

$.^I$ est une fonction d'interprétation tel que

$$A^I \subseteq \Delta^I$$

$$r^I \subseteq \Delta^I \times \Delta^I$$

$$a^I \in \Delta^I$$

$$\top^I =_{def} \Delta^I$$

$$\perp^I =_{def} \emptyset$$

$$(\neg C)^I =_{def} \Delta^I \setminus C^I$$

$$C \sqcap D)^I =_{def} C^I \cap C^I$$

$$C \sqcup D)^I =_{def} C^I \cup C^I$$

$$\exists r.C)^I =_{def} \{x \in \Delta^I \mid r^I(x) \cap C^I \neq \emptyset\}$$

$$\forall r.C)^I =_{def} \{x \in \Delta^I \mid r^I(x) \subseteq C^I\}$$

17.1.2 Propriétés

Pour toutes les interprétations $\iota = \langle \Delta^I, .^I \rangle$, et pour tout $C, D \in \ell_{ALC}$:

$$(\neg \neg C)^I = C^I$$

$$(\neg \exists r.C)^I = (\forall r. \neg C)^I$$

$$(\neg(C \sqcap D))^I = (\neg C \sqcup \neg D)^I$$

$$(\neg(C \sqcup D))^I = (\neg C \sqcap \neg D)^I$$

$$\exists r. \perp \equiv \perp$$

$$(\neg \forall r.C)^I = (\exists r. \neg C)^I$$

$$\forall r. \top \equiv \top$$

17.2 Logique de description

Définit via les symboles suivant:

$\ell_{ALC}, C \sqsubseteq C, \sqsupseteq C$

17.2.1 Sémantique

$$\iota \models C \sqsubseteq D \text{ (satisfait } C \sqsubseteq D) \text{ si } C^I \subseteq D^I$$

$$\iota \models C \equiv D \text{ si } \iota \models C \sqsubseteq D \text{ et } \iota \models C \sqsupseteq D$$

17.2.2 Assertions

$a : C$ a est une instance de C

$(a, b) : r$ a et b sont attaché avec la relation r

17.3 TBoxes et ABoxes

Soit une base de connaissance $KB = \langle T, A \rangle$ où:

$$T = \begin{cases} EmpStud \equiv Student \sqcap Employee \\ Student \sqcap \neg Employee \sqsubseteq \neg \exists pays.Tax \\ EmpStud \sqcap \neg Parent \sqsubseteq \exists pays.Tax \\ EmpStud \sqcap Parent \sqsubseteq \neg \exists pays.Tax \\ \exists worksFor.Company \sqsubseteq Employee \end{cases}$$

$$A = \begin{cases} ibm : Company \\ mary : Parent \\ john : EmpStud \\ (john, ibm) : workFor \end{cases}$$

17.3.1 Subsumption

D'après la TBoxes et la ABoxes ci dessus, dire que A subsume B c'est dire que A est plus spécifique que B:

Does *EmpStud* subsume *Student* \sqcap *Employee* ? : yes

Does *Student* \sqcap *Parent* subsume *EmpStud* \sqcap *Parent* ? : yes

Does $\exists pays.\perp$ subsume *EmpStud* ? : No

17.3.2 Classification

Les schémas de classification aide pour trouver les subsumptions:



17.3.3 Instance checking

On n'a

ibm est une instance de *Company*

mary est une instance de *Parent*

john est une instance de *EmpStud*, *Student*, *Employee*

john n'est pas une instance de \neg *Parent*

(john, ibm) est une instance de *workFor*

17.3.4 Retrieval

Student ?{*john*}

$\neg \exists$ *pays.Tax* ?{*mary*}

$\neg(\neg$ *Employee* $\sqcap \exists$ *pays.Tax*) ?{*john, mary*}

\forall *worksFor.Company* ?{ }

Employee $\sqcup \forall$ *pays.* \neg *Tax* \sqcup *Company* ?{*ibm, john, mary*}

\neg *Tax* $\sqcup \exists$ *pays.* \perp $\sqcup \forall$ *workdFor.* \forall *pays.* \top ?{*ibm, john, mary*}

17.3.5 Equivalence of concept

Are *Student* \sqcap *Employee* $\sqcap \neg$ *EmpStud* and \exists *worksFor.* \perp équivalent? *Yes*

Are *Student* $\sqcap \forall$ *worksFor.* \neg *Company* and *Student* $\sqcap \neg$ *Employee* équivalent?
No

17.3.6 Concept satisfiability

EmpStud \sqcap *Parent* $\sqcap \exists$ *pays.* \top satisfiable? *Yep*

$\neg \forall$ *worksFor.* \neg *Company* $\sqcap \neg$ *Employee* satisfiable? *No*

Employee \sqcap *Company* satisfiable ? *Yep*

17.3.7 ABox consistency

Is $A_2 = A \cup \{\textit{john} : \exists \textit{worksFor} . \neg \textit{Company}\}$ consistent wrt T ? : *Yes*

Is $A_3 = A \cup \{\textit{mary} : \exists \textit{pays} . \textit{Tax}\}$ consistent wrt T ? : *No*

17.3.8 Réduction et consistance

Soit $KB = \langle T, A \rangle, C, D \in \iota_{ALC}, a \in I$ and a' new in KB

Concept subsumption wrt T : $KB \models C \sqsubseteq D$ ssi $\langle T, A \cup \{a' : C \sqcap \neg D\} \rangle$ est inconsistant

Instance chacking : $KB \models a : C$ ssi $\langle T, A \cup \{a : \neg C\} \rangle$ est inconsistant

Concept satisfiability wrt T : C est satisfiable wrt T ssi $\langle T, A \cup \{a' : C\} \rangle$ est consistent

$KB \models \textit{EmpStud} \sqcap \textit{Parent} \sqsubseteq \neg \exists \textit{pays} . \textit{Tax} \sqcap \textit{Employee}$?

$KB \cup \{a : \textit{EmpStud} \sqcap \textit{Parent} \sqcap (\exists \textit{pays} . \textit{Tax} \sqcup \neg \textit{Employee})\} \models \perp?$, for a new

$KB \models \textit{john} : \textit{Student} \sqcap \exists \textit{empBy} . \top$?

$KB \cup \{\textit{john} : \neg(\textit{Student} \sqcap \exists \textit{empBy} . \top)\} \models \perp?$

Is $\textit{EmpStud} \sqcap \neg \exists \textit{pays} . \textit{Tax}$ satisfiable wrt KB ?

$KB \cup \{a : \textit{EmpStud} \sqcap \neg \exists \textit{pays} . \textit{Tax}\} \not\models \perp?$, for a new

Chapter 18

Méthode des Tableau pour les ALC

18.1 Pre processing

18.1.1 Réécriture

Réécrite chaque:

$$C \sqsubseteq D \text{ dans } T \text{ en } \top \sqsubseteq \neg C \sqcup D$$

$$A \sqsubseteq \exists r.B \text{ en } \top \sqsubseteq \neg A \sqcup \exists r.B$$

Changer la KB en NNF (\neg occurs only in front of concept names)

$$\neg\neg C \rightarrow C$$

$$\neg(C \sqcap D) \rightarrow \neg C \sqcup \neg D$$

$$\neg(C \sqcup D) \rightarrow \neg C \sqcap \neg D$$

$$\neg(\exists r.C) \rightarrow \forall r.\neg C$$

$$\neg(\forall r.C) \rightarrow \exists r.\neg C$$

18.1.2 Vocabulaire

Blocage/Blocking l'apparition d'une boucle infini dans le déroulement de l'algorithme

Clash Quand il existe une contradiction d'un noeud feuille vers l'un de ses ascendant

18.1.3 Règles d'expansion

\sqsubseteq_T – rule

Si $a : C \in A, \top \sqsubseteq D \in T$ **et** $a : D \notin A$ **alors**
 $A := A \cup \{a : D\}$

\sqcap – rule

Si $a : C \sqcap D \in A$ **et** $\{a : C, a : D\} \not\subseteq A$ **alors**
 $A := A \cup \{a : C, a : D\}$

\sqcup – rule

Si $a : C \sqcup D \in A$ **et** $\{a : C, a : D\} \cap A = \emptyset$ **alors**
 $A := A \cup \{a : E\}$, for some $E \in \{C, D\}$

\exists – rule

Si $a : \exists R.C \in A$ **et il n'y a pas de** b **st** $\{(a, b) : R, b : C\} \subseteq A$ **et**
 a **n'est pas en en blocage** **alors**
 $A := A \cup \{(a, c) : R, c : C\}$, for c new in A

\forall – rule

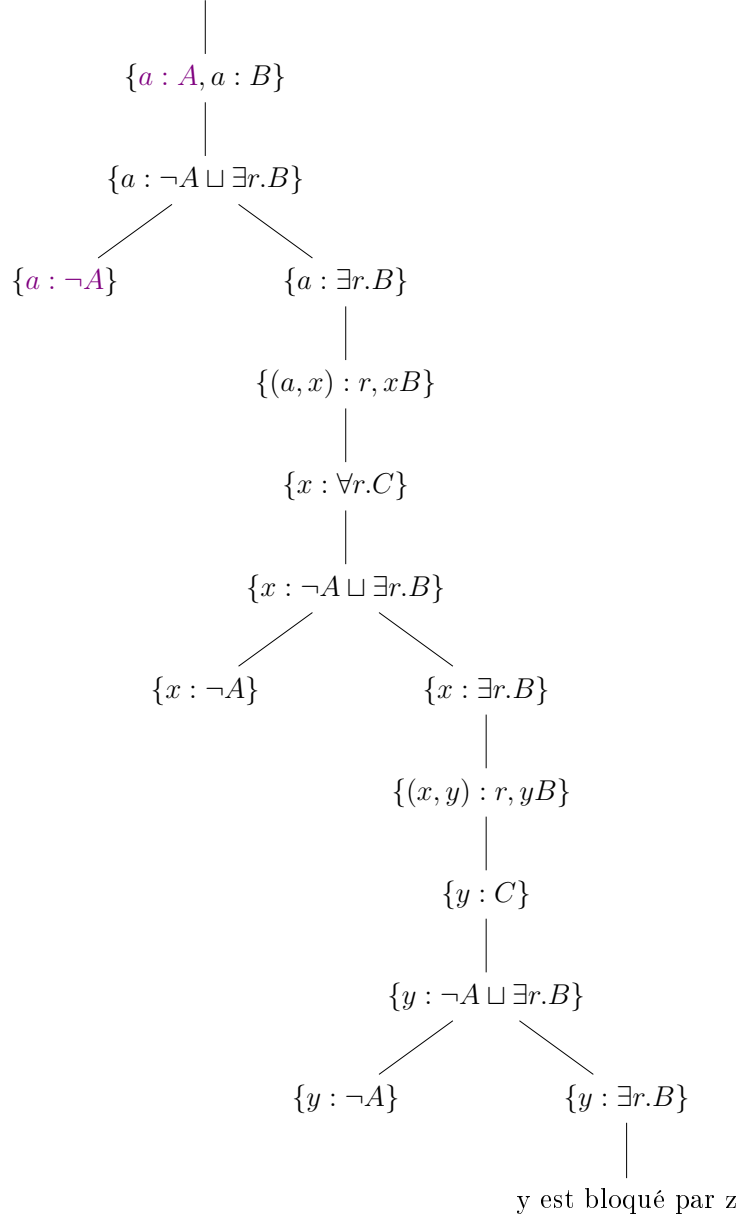
Si $\{a : \forall R.C, (a, b) : R\} \subseteq A$ **et** $b : C \notin A$ **alors**
 $A := A \cup \{b : C\}$

18.2 Exemple

$$T = \{A \sqsubseteq \exists r.B\} \equiv \{\top \sqsubseteq \neg A \sqcup \exists r.B\}$$

$$A = \{a : A \sqcap B, a : \forall r. \forall r.C\}$$

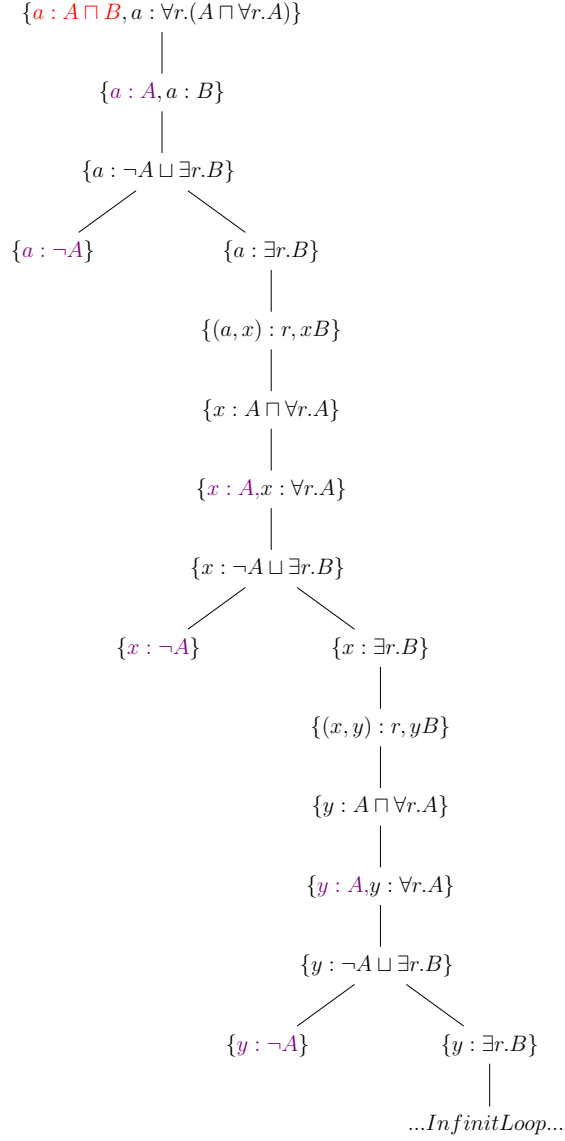
$$\{a : A \sqcap B, a : \forall r. \forall r.C\}$$



18.3 Exemple 2

$$T = \{A \sqsubseteq \exists r.B\} \equiv \{\top \sqsubseteq \neg A \sqcup \exists r.B\}$$

$$A = \{a : A \sqcap B, a : \forall r.(A \sqcap \forall r.A)\}$$

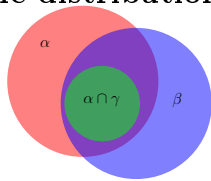


Chapter 19

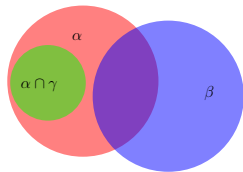
Logique presque tout

Soit le nouvelle opérateur binaire \Subset disent pour *presque tout* A est dans B.

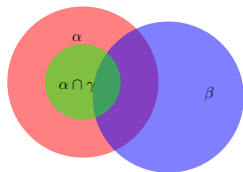
Bonne distribution :



Mauvaise distribution :



Cas général :



19.1 Système P

Réflexivité :

Almost all : $\alpha \vdash \alpha$

ensembliste : $A \in A$

Équilibrage à gauche :

Almost all : Si $\models \alpha \Leftrightarrow \beta$ et $\alpha \vdash \gamma$ alors $\beta \vdash \gamma$

ensembliste : Si $A = B$ et $A \in C$ alors $B \in C$

Équilibrage à droite :

Almost all : Si $\alpha \models \beta$ et $\gamma \vdash \alpha$ alors $\gamma \vdash \beta$

ensembliste : Si $A \subseteq B$ et $C \in A$ alors $C \in B$

Coupure :

Almost all : Si $(\alpha \wedge \beta) \vdash \gamma$ et $\alpha \vdash \beta$ alors $\alpha \vdash \gamma$

ensembliste : Si $(A \cap B) \in C$ et $A \in B$ alors $A \in C$

Monotonie :

Almost all : Si $\alpha \vdash \beta$ et $\alpha \vdash \gamma$ alors $\alpha \wedge \beta \vdash \gamma$

ensembliste : Si $A \in B$ et $A \in C$ alors $(A \cap B) \in C$

Ou :

Almost all : Si $\alpha \vdash \gamma$ et $\beta \vdash \gamma$ alors $\alpha \vee \beta \vdash \gamma$

ensembliste : Si $A \in C$ et $B \in C$ alors $(A \cup B) \in C$

Part VI

XML

Chapter 20

DTD

inclusion dans xml `<!DOCTYPE nom SYSTEM "fichier.dtd" >`

`<!ELEMENT <nom de la balise> (< contenue >) >`

contenue :

`(#PCDATA)` du texte

`(objet+)` au moins un objet

`(objet?)` au plus un objet

`(objet*)` de zero à infini

`(objet, aliment)` un objet et une aliment (dans l'ordre)

`(objet|aliment)` l'un des deux

`<!ATTRIBUT <nom de la balise>< clef >< contenue > [#REQUIRED|#IMPLIED] >`

Chapter 21

XSD

```
<?xml version="1.0" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="age" type="xs:integer" />
    <xs:element name="prenom" type="xs:string" />
    <xs:element name="pseudo" type="xs:string" />

    <xs:element name="Personnage">$
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="age">
                <xs:element ref="name" maxOccurs="2">
            </>
        </>
    </>

    <xs:element name="Joueur">
        <xs:complexType>
            <xs:sequence>
                <xs:choise minOccurs="1" maxOccurs="2">
                    <xs:element ref="pseudo">
                    <xs:element ref="prenom">
                </>
            </>
        </>
    </>

</>
```

```
</>

<xs:complexType name="Liste">
  <xs:sequence>
    <xs:element ref="Joueur" maxOccurs="UNBOUNDED">
      </>
    </>
  </>

  <xs:element name="photo" type="xs:string">
    <xs:attribute name="resolution" type="xs:integer"/>
    <xs:attribute name="type" type="xs:string" default="img/jpg"/>
  </xs:element>
</xs:schema>
```

Chapter 22

XPATH

tous les noeuds éléments de nom attr *"movie/filmography/*/attr"*

tous les noeuds éléments qui ont un attribut order *"movie/filmography/*[@order]"*

les noeuds attributs de nom order *"movie/filmography/*/@order"*

le quatrième fils de filmography
"movie/filmography/[position() = 4]"*

le noeud attribut de nom Crew (attribut de filmography)
"movie/filmography/@Crew"

les quatre premiers fils de filmography
"movie/filmography/[position() <= 4]"*

retournez le sous-arbre parent et aller dans Editor *"../editor"*

le nombre total de comédiens (cast, remainder)
"count(movie/filmography/cast/name | movie/filmography/remainder/name)"

une chaîne de caractères présentant le film : titre (année) - réalisateur
"concat(movie/title/text(), '(', movie/year/text(), ') - ', movie/film/Director/name)"