Bases de données

Je mets ci-dessous le bagage minimum sur les bases de données qu'il faut avoir en allant à l'agrégation. C'est organisé comme mon plan de la leçon 932.

Voici un exemple de base de données :

Films

Titre	Réalisateur	Acteur
Didier	A. Chabat	A. Chabat
Au Poste	Q. Dupieux	G. Ludig
Réalité	Q. Dupieux	A. Chabat
Astérix	A. Chabat	E. Baer

Séances

Cinéma	Horaires	Titre
UGC	$12^{\text{h}}\,00$	Au Poste
Gaumont	$20^{\text{h}}\,50$	Didier
Pathé	$13^{\text{h}}\,30$	Réalité
Pathé	$13^{\text{h}}\,30$	Didier

On veut se poser les questions suivantes :

- Q_1 : Quels films ont été réalisés par A. Chabat ?
- Q_2 : Quels sont les films réalisés par A. Chabat ou Q. Dupieux ?
- Q_3 : Quels sont les films de Q. Dupieux dans lesquels A. Chabat ne joue pas ?

Donnons maintenant le formalisme que l'on met sur les bases de données.

I - Formalisme

Soient les ensembles dénombrables distincts suivants :

- de variables $\textsc{Var} = \{x, y, z, \ldots\}$
- d'attributs Att
- de domaines Dom
- de noms de schéma de relation RelName

Définition. On se donne une fonction

$$sort : \textsc{RelName} \to \mathcal{P}_f(\textsc{Att})$$

qui a $R \in \textsc{RelName}$ associe un nombre fini d'attributs.

Définition. On dira que, pour $R \in \textsc{RelName}$, la quantité $|sort(R)|$ est l'arité de R.

Définition. Un élément $R \in \textsc{RelName}$ est un schéma de relation. On notera $R[U]$ où $U = sort(R)$.

Définition. Un nombre fini de schémas de relation $R^{(1)}[U^{(1)}], \ldots, R^{(n)}[U^{(n)}]$ est un schéma de base de données. On notera

$$\mathcal{R} = \{R^{(1)}[U^{(1)}], \ldots, R^{(n)}[U^{(n)}]\}.$$

Exemple.

$$\mathcal{R} = \{\textsc{Films}[\text{Titre}, \text{Réalisateur}, \text{Acteur}], \textsc{Séances}[\text{Cinéma}, \text{Horaires}, \text{Titre}]\}$$

Définition. Un tuple sur un schéma de relation $R[U]$ est une fonction

$$u : U \to \textsc{Dom}.$$

On notera le tuple par

$$\langle U_1 : u(U_1), \dots, U_n : u(U_n) \rangle$$

On s'autorisera à noter $\langle u(U_1), \dots, u(U_n) \rangle$ s'il n'y a pas ambiguïté des attributs.

Exemple. Voici un tuple sur le schéma de relation Films[Titre, Réalisateur, Acteur].

$$\langle \text{Titre} : \text{Didier}, \text{Réalisateur} : \text{A. Chabat}, \text{Acteur} : \text{A. Chabat} \rangle$$

Définition. Un tuple libre sur un schéma de relation $R[U]$ est une fonction $u : U \to \textsc{Dom} \cup \textsc{Var}$.

Exemple. $\langle x, \text{A. Chabat}, y \rangle$ est un tuple libre.

Définition. Une relation I sur un schéma de relation $R[U]$ est un nombre fini de tuples sur $R[U]$.

Exemple. La table Films[Titre, Réalisateur, Acteur] de l'exemple ci-dessus est une relation.

Définition. Une base de données $\mathcal{I}$ sur un schéma de base de données $\mathcal{R}$ est un ensemble fini de relations, où une relation I est sur un schéma de relation $R[U] \in \mathcal{R}$.

Exemple. L'ensemble des tables de l'exemple ci-dessus est une base de données.

On veut maintenant pouvoir effectuer des requêtes sur les bases de données afin d'en extraire des informations. Pour cela, on va donner plusieurs approches qui sont de plus en plus expressives.

II - Requêtes conjonctives

1) Règles conjonctives

Définition. Une règle conjonctive sur $\mathcal{R}$ est une expression de la forme

$$ans(u) \leftarrow R^{(1)}(u^{(1)}), \dots, R^{(n)}(u^{(n)})$$

où $u, u^{(1)}, \dots, u^{(n)}$ sont des tuples libres sur des relations $R^{(i)} \in \mathcal{R}$ et $ans \notin \mathcal{R}$.

Exemple. $ans(x) \leftarrow \textsc{Films}(x, \text{A. Chabat}, y)$ est une règle conjonctive.

Définition. Une valuation ν sur Var est une fonction $\nu : \textsc{Var} \to \textsc{Dom}$. On la prolonge sur les tuples libres en imposant que $\nu_{|\textsc{Dom}} = \text{Id}_{|\textsc{Dom}}$.

Définition. La sémantique d'une règle conjonctive q pour $\mathcal{I}$ sur $\mathcal{R}$ est donnée par

$$q(\mathcal{I}) = \left\{ \nu(u) \ : \ \nu \text{ est une valuation sur } \textsc{Var}(q) \text{ et } \nu(u_i) \in \mathcal{I}(R^{(i)}) \, \forall i \in [\![1; n]\!] \right\}$$

Exemple. $ans(x) \leftarrow \textsc{Films}(x, \text{A. Chabat}, y)$ a pour sémantique

$$\langle \text{Didier}, \text{A. Chabat}, \text{A. Chabat} \rangle$$
$$\langle \text{Astérix}, \text{A. Chabat}, \text{E. Baer} \rangle$$

Comme on demande l'information x, il faudra ensuite donner le titre des films, ici Didier et Astérix. On a répondu à la question Q_1.

Définition. Le domaine actif de $\mathcal{I}$ (resp. de q, $q(\mathcal{I})$), noté $adom(\mathcal{I})$ (resp. $adom(q)$, $adom(q(\mathcal{I}))$) est l'ensemble des éléments de Dom présents dans $\mathcal{I}$ (resp. q, $q(\mathcal{I})$).

On remarque que l'on a $adom(q(\mathcal{I})) \subset adom(q) \cup adom(\mathcal{I})$.

2) Calcul conjonctf

Définition. Soit un schéma de base de données $\mathcal{R}$. Une formule du calcul conjonctif est de la forme

- $R[u]$, avec u un tuple libre sur $R[U]$ et $R[U] \in \mathcal{R}$;
- $\phi \wedge \psi$ avec ϕ et ψ des formules du calcul conjonctif ;
- $\exists x\, \phi$, avec $x \in$ Var et ϕ une formule du calcul conjonctif.

Définition. Une requête du calcul conjonctif q sur $\mathcal{R}$ est une expression de la forme

$$q = \{e_1, \dots, e_n \mid \varphi\}$$

où $\langle e_1, \dots, e_n \rangle$ est un tuple libre, φ une formule du calcul conjonctif et l'ensemble des variables libres de φ est $\{e_1, \dots, e_n\}$.

Définition. Soient un schéma de base de données $\mathcal{R}$, une formule du calcul conjonctif φ sur $\mathcal{R}$ et une valuation ν sur les variables libres de φ. On dit que la base de données $\mathcal{I}$ sur $\mathcal{R}$ satisfait φ pour ν, noté $\mathcal{I} \vDash \varphi[\nu]$, si

- pour $\varphi = R[u]$, on a $\nu(u) \in \mathcal{I}$;
- pour $\varphi = \phi \wedge \psi$, on a $\mathcal{I} \vDash \phi[\nu]$ et $\mathcal{I} \vDash \psi[\nu]$;
- pour $\varphi = \exists x\, \phi$, il existe $c \in$ Dom tel que $\mathcal{I} \vDash \phi[\nu \cup [x \mapsto c]]$.

Définition. La sémantique d'une requête q du calcul conjonctif est donnée par

$$q(\mathcal{I}) = \{\nu(\langle e_1, \dots, e_n \rangle) \ : \ \nu \text{ une valuation sur les variables libres de } \varphi \text{ et } \mathcal{I} \vDash \varphi[\nu]\}$$

Exemple. Pour répondre à la question Q_1, on écrit

$$q = \{x \mid \exists y\, \textsc{Films}(x, \text{A. Chabat}, y)\}$$

Théorème 1. Le calcul conjonctif et les règles conjonctives sont équivalentes.

Toutes les requêtes conjonctives sont finies.

III - L'algèbre SPC

Définition. On définit les trois opérations suivantes sur $I, J \in \mathcal{I}$ qui formeront l'algèbre SPC.

- la sélection : $\sigma_{j=a}(I) = \{t \in I, t(j) = a\}$ avec $j \in$ Att, $a \in$ Dom et t un tuple. On peut aussi définir $\sigma_{j=k}(I) = \{t \in I, t(j) = t(k)\}$ avec $j, k \in$ Att et t un tuple.
- la projection : $\prod_{j_1,\dots,j_n}(I) = \{\langle t(j_1), \dots, t(j_n)\rangle, t \in I\}$ avec $j_1, \dots, j_n \in$ Att.

- le produit carthésien : $I \times J = \{\langle t(1), \dots, t(n), s(1), \dots, s(m)\rangle, t \in I, s \in J\}$ avec $n =$ arité(I) et $m =$ arité(J).

Exemple. Pour répondre à la question Q_1, on écrit

$$\prod_{\text{Titre}} (\sigma_{\text{Réalisateur=A. Chabat}}(\text{FILMS}))$$

Proposition : On peut effectuer la jointure naturelle et l'intersection avec les trois opérations de l'algèbre SPC.

Toutes les requêtes de l'algèbre SPC sont finies mais ne sont pas forcément satisfiables.

Exemple. La requête $\prod_A \sigma_{A=0}\sigma_{A=1}(I)$ est insatisfiable.

Théorème 2. Il y a équivalence entre les requêtes par règles conjonctives, le calcul conjonctif et les requêtes satisfiables de l'algèbre SPC.

Pour l'instant, on ne peut pas répondre aux questions Q_2 et Q_3. C'est pourquoi on va introduire de nouvelles opérations.

IV - L'algèbre relationnelle et le calcul relationnel

1) Ajout de l'union

Définition. On ajoute l'union dans l'algèbre SPC (qui devient alors l'algèbre SPCU) en autorisant la sélection

$$\sigma_{j=a \text{ ou } j=b}$$

Définition. On ajoute le connecteur logique « ou », noté $\vee$, aux formules du calcul conjonctif.

Exemple. On peut donc maintenant répondre à la question Q_2.

$$\prod_{\text{Titre}} (\sigma_{\text{Réalisateur=A. Chabat ou Réalisateur=Q. Dupieux}}(\text{FILMS}))$$

On peut alors avoir des requêtes infinies.

Exemple. La requête $\{x, y \mid R(x) \vee R(y)\}$ avec $R(x_0) = 1$ pour $x_0 \in \text{DOM}$. En effet, pour tout $y \in \text{DOM}$, on a $R(x_0) \vee R(y) = 1$. Ainsi, pour un domaine infini, on a une requête infinie.

Définition. On dit qu'une requête est sûre si $|q(\mathcal{I})| < +\infty$ pour toute base de données $\mathcal{I}$ sur $\mathcal{R}$.

Le problème RELSURE est indécidable.

RELSURE
- entrée : une requête q en calcul relationnel ;
- sortie : oui si $|q(\mathcal{I})| < +\infty$ pour toute base de données $\mathcal{I}$, non sinon.

2) Ajout de la négation

Définition. On ajoute la différence ensembliste dans l'algèbre SPCU, notée $\setminus$, qui devient alors l'algèbre relationnelle SPCUD.

Définition. On ajoute le connecteur logique de négation, noté $\neg$, aux formules afin d'obtenir le calcul relationnel.

On peut donc exprimer le quantificateur universel $\forall$.

Exemple. On peut donc maintenant répondre à la question Q_3.

$$\prod_{\text{Titre}} (\sigma_{\text{Réalisateur=Q. Dupieux}}(\textsc{Films}) \setminus \sigma_{\text{Acteur=A. Chabat}}(\textsc{Films}))$$

On peut de nouveau avoir des requêtes infinies.

Exemple. La requête $\{x \mid \neg R(x_0)\}$ pour $x_0 \in \textsc{Dom}$. En effet, pour tout $x \in \textsc{Dom} \setminus \{x_0\}$, on a $\neg R(x_0) = 1$. Ainsi, pour un domaine infini, on a une requête infinie.

Pour une base de données $\mathcal{I}$ sur $\mathcal{R}$, on restreint le domaine au domaine actif ainsi, pour une requête et une relation fixées, la requête sera finie.

Théorème 3. Restreints au domaine actif, l'algèbre relationnelle SPCUD et le calcul relationnel sont équivalents.

Les problèmes suivants sont indécidables :

RelSat
- entrée : une requête q en calcul relationnel ;
- sortie : oui s'il existe $\mathcal{I}$ telle que $q(\mathcal{I}) \neq \emptyset$, non sinon.

RelEqu
- entrée : deux requêtes q et q' du calcul relationnel ;
- sortie : oui si q et q' sont sémantiquement équivalentes, non sinon.

On montre l'indécidabilité de RelSat dans le développement Indécidabilité de la satisfiabilité d'une requête.

Remarques :

On peut chercher à optimiser les opérations d'écriture et de lecture dans la base de données. On peut par exemple utiliser des B-arbres comme il est montré dans le développement sur les B-arbres.