**Bases de Données I, Mons, 23 janvier 2023**

Cet examen contient 10 questions. Durée : exactement 3 heures.

NOM + PRÉNOM :

Orientation + Année :

**Question 1** La Société Nationale des Cinémas Belges (SNCB) stocke des informations sur les cinémas et leur programmation actuelle dans la table montrée ci-dessous. Dans une même ville, il ne peut pas y avoir deux cinémas différents avec le même nom. Chaque cinéma possède une adresse postale unique. Un cinéma peut avoir plusieurs salles de projection, qui sont numérotées I, II, III, ... La capacité des salles est comprise entre 100 et 500 places.

Les premières deux lignes signifient que le film "Felice" de P. Delpeut est programmé à 19:15 et à 21:15 au cinéma Utopia à Alost, dans la salle I. Ce film, avec une durée de 1 heure et 39 minutes, est aussi programmé au cinéma Rex à Alost, à 18:00, dans la salle II (sixième ligne). Il ne faut pas confondre ce film avec celui de S. Spielberg montré à l'Utopia à Namur (dernière ligne).

Il est impossible d'avoir deux cinémas différents à la même adresse postale. Évidemment, aucun cinéma ne peut programmer deux films différents au même moment dans une même salle. Un film est identifié de manière unique par son titre plus son réalisateur. La durée d'un film est une donnée fixe. Notons que deux films différents peuvent avoir le même titre. Chaque cinéma possède un ou plusieurs numéros de téléphone, mais aucun numéro n'est partagé parmi plusieurs cinémas.

| *Cinéma* | *Salle* | *Capacité* | *Rue* | *Ville* | *Téléphone* | *Titre* | *Réalisateur* | *Durée* | *Heure* |
|---|---|---|---|---|---|---|---|---|---|
| Utopia | I | 300 | 6 Pl. du Parc | Alost | 053 66 33 33 | Felice | P. Delpeut | 1:39 | 19:15 |
| Utopia | I | 300 | 6 Pl. du Parc | Alost | 053 66 33 33 | Felice | P. Delpeut | 1:39 | 21:15 |
| Utopia | I | 300 | 6 Pl. du Parc | Alost | 053 88 44 44 | Felice | P. Delpeut | 1:39 | 19:15 |
| Utopia | I | 300 | 6 Pl. du Parc | Alost | 053 88 44 44 | Felice | P. Delpeut | 1:39 | 21:15 |
| Rex | I | 350 | 8 R. du Marché | Alost | 053 44 22 22 | Felice | S. Spielberg | 2:05 | 18:00 |
| Rex | II | 250 | 8 R. du Marché | Alost | 053 44 22 22 | Felice | P. Delpeut | 1:39 | 18:00 |
| Utopia | I | 100 | 5 Av. Codd | Namur | 081 33 66 99 | Sisters | T. Ravolta | 1:30 | 18:15 |
| Utopia | II | 150 | 5 Av. Codd | Namur | 081 33 66 99 | Brothers | T. Ravolta | 1:30 | 18:15 |
| Utopia | I | 100 | 5 Av. Codd | Namur | 081 33 66 99 | Felice | S. Spielberg | 2:05 | 19:00 |

Quelles sont les dépendances fonctionnelles que l'on peut raisonnablement imposer sur ces données ?

.../10

*Cinéma, Ville* $\rightarrow$ *Rue*
*Ville, Rue* $\rightarrow$ *Cinéma*
*Téléphone* $\rightarrow$ *Cinéma, Ville*
*Salle, Cinéma, Ville, Heure* $\rightarrow$ *Titre, Réalisateur*
*Salle, Cinéma, Ville* $\rightarrow$ *Capacité*
*Titre, Réalisateur* $\rightarrow$ *Durée*

Certains étudiants ont proposé

*Cinéma, Ville, Titre, Réalisateur, Heure* $\rightarrow$ *Salle*,

exprimant qu'aucun cinéma ne projette le même film en parallèle dans deux salles différentes ; cette contrainte n'est pas déraisonnable.

**Question 2** Les tables suivantes stockent les résultats de cyclisme pour des courses d'une journée, qui sont organisées une fois par an. Les coureurs d'une course sont classés 1, 2, 3, ... sans ex æquo (i.e., pour une même course, deux participants ne peuvent pas avoir la même valeur pour l'attribut *Position*). La table *ABRÉVIATIONS* donne les abréviations utilisées pour identifier les courses. L'attribut *Temps* est le temps parcouru dans le format $hh : mm : ss$, c'est-à-dire, en heures, minutes, secondes. La table *COUREURS* liste l'affiliation des coureurs par année. La table *DATES* liste, pour chaque course, le jour où la course a eu lieu.

*CLASSEMENTS*

| *Année* | *Course* | *Position* | *Coureur* | *Temps* |
|---|---|---|---|---|
| 1970 | PR | 1 | Eddy Merckx | 05 : 45 : 21 |
| 1970 | PR | 153 | Jean Baptiste | 06 : 05 : 11 |
| 2016 | RV | 1 | Julian Alaphilippe | 06 : 35 : 04 |
| 2016 | RV | 53 | Oliver Naesen | 06 : 39 : 14 |
| 2016 | NC | 1 | Oliver Naesen | 02 : 19 : 04 |
| 2016 | NC | 122 | Michael Matthews | 02 : 37 : 09 |
| 2016 | NC | 123 | Julian Alaphilippe | 02 : 37 : 09 |
| 2019 | LBL | 1 | Julian Alaphilippe | 05 : 01 : 45 |
| 2019 | LBL | 2 | Oliver Naesen | 05 : 01 : 45 |
| 2019 | LBL | 33 | Michael Matthews | 05 : 10 : 07 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

*ABRÉVIATIONS*

| *Code* | *Course* |
|---|---|
| PR | Paris-Roubaix |
| RV | Ronde Van Vlaanderen |
| LBL | Liège-Bastogne-Liège |
| MS | Milano-Sanremo |
| NC | Ninove Classic |
| ⋮ | ⋮ |

*DATES*

| *Année* | *Course* | *Jour* |
|---|---|---|
| 1970 | PR | 12 avril |
| 2016 | MS | 19 mars |
| 2016 | RV | 3 avril |
| 2016 | NC | 23 avril |
| 2019 | LBL | 24 avril |
| 2019 | NC | 24 avril |
| ⋮ | ⋮ | ⋮ |

*COUREURS*

| *Nom* | *Année* | *Équipe* |
|---|---|---|
| Eddy Merckx | 1970 | Molteni |
| Jean Baptiste | 1970 | Molteni |
| Oliver Naesen | 2016 | IAM Cycling |
| Oliver Naesen | 2019 | AG2R |
| Michael Matthews | 2016 | AG2R |
| Michael Matthews | 2019 | Sunweb |
| Julian Alaphilippe | 2016 | Quick Step |
| Julian Alaphilippe | 2019 | Quick Step |
| ⋮ | ⋮ | ⋮ |

Donnez toutes les clés primaires, les clés étrangères et les contraintes UNIQUE.

.../10

*CLASSEMENTS* PRIMARY KEY (*Année, Course, Position*)
UNIQUE (*Année, Course, Coureur*)
FOREIGN KEY (*Coureur, Année*) REFERENCES *COUREURS*
FOREIGN KEY (*Année, Course*) REFERENCES *DATES*

*ABRÉVIATIONS* PRIMARY KEY (*Code*)
UNIQUE (*Course*)

*DATES* PRIMARY KEY (*Année, Course*)
FOREIGN KEY (*Course*) REFERENCES *ABRÉVIATIONS*

*COUREURS* PRIMARY KEY (*Nom, Année*)

**Question 3** Pour la base de données de la question 2, écrivez une requête en algèbre relationnelle pour répondre à la question :

> Pour chaque course, listez le dernier coureur de la course (c'est-à-dire, le coureur avec la position la plus élevée).

Pour la base de données de la question 2, le résultat est :

| *Année* | *Course* | *Coureur* |
|---|---|---|
| 1970 | PR | Jean Baptiste |
| 2016 | RV | Oliver Naesen |
| 2016 | NC | Julian Alaphilippe |
| 2019 | LBL | Michael Matthews |

Pour cette question, on peut utiliser la sélection $\sigma_{A<B}(E)$ avec $A, B \in sorte(E)$. La sémantique est :

$$[\![\sigma_{A<B}(E)]\!]^{\mathcal{I}} = \{t \in [\![E]\!]^{\mathcal{I}} \mid t(A) < t(B)\}.$$

---

$\boxed{\ldots/10}$

Soit $F := \rho_{\textit{Position}\to\textit{Rang}}\left(\pi_{\textit{Année, Course, Position}}\left(\textit{CLASSEMENTS}\right)\right)$.

Soit

$$G := \pi_{\textit{Année, Course, Coureur}}\left(\sigma_{\textit{Position}<\textit{Rang}}\left(\textit{CLASSEMENTS} \bowtie F\right)\right). \tag{1}$$

Un tuple $\{\textit{Année} : x, \textit{Course} : y, \textit{Coureur} : z\}$ dans $G$ signifie que le coureur $z$ a participé à la course $y$ dans l'année $x$, et au moins un autre coureur était derrière lui dans cette course.

La requête demandée est alors :

$$\pi_{\textit{Année, Course, Coureur}}\left(\textit{CLASSEMENTS}\right) - G.$$

*Notez : pour que la jointure dans* (1) *donne le résultat souhaité, il est important que "Coureur" n'est pas un attribut de $F$.*

**Question 4** Pour la base de données de la question 2, écrivez une requête en calcul relationnel qui renvoie le nom de chaque cycliste qui a gagné au moins une course dans chaque année où il était coureur. Pour la base de données de la question 2, le résultat est comme suit :

| |
|---|
| Eddy Merckx |
| Julian Alaphilippe |

Oliver Naesen n'est pas dans le résultat, car il était coureur en 2019, mais n'apparaît pas en 1ère position d'une course en 2019. Noter qu'Eddy Merckx est dans la réponse, même s'il n'a pas gagné de course en 2016 ; en fait, Eddy Merckx n'était plus coureur en 2016.

---

.../10

$$\begin{aligned}\{x \mid\ & \exists y \exists z\, (\mathit{COUREURS}(x, y, z)) \\ & \wedge \forall y \forall z\, (\mathit{COUREURS}(x, y, z) \rightarrow \exists s \exists t\, (\mathit{CLASSEMENTS}(y, s, \text{“1”}, x, t)))\}\end{aligned}$$

**Question 5** Supposons que l'on ajoute un nouvel opérateur binaire $\ominus$ à l'algèbre SPJRUD. De façon formelle, $\ominus$ est défini comme suit :

**Syntaxe :** Si $E_1$ et $E_2$ sont des expressions algébriques telles que $sorte(E_1) \subseteq sorte(E_2)$, alors $E_1 \ominus E_2$ est une expression algébrique telle que $sorte(E_1 \ominus E_2) = sorte(E_1)$.

**Sémantique :** Si $\mathcal{I}$ est une instance de base de données, alors $[\![E_1 \ominus E_2]\!]^{\mathcal{I}}$ contient chaque tuple de $[\![E_1]\!]^{\mathcal{I}}$ qui est inclus dans au plus 1 tuple de $[\![E_2]\!]^{\mathcal{I}}$. Noter que *"dans au plus* 1 *tuple"* est synonyme de *"dans aucun ou un seul tuple, mais pas dans deux tuples"*.

Par exemple, soit $e_1$ et $e_2$ les relations suivantes :

$$e_1 \begin{array}{|cc} C & D \\ \hline 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{array} \qquad e_2 \begin{array}{|cccc} B & C & D & E \\ \hline a & 0 & 1 & b \\ a & 0 & 1 & c \\ a & 1 & 1 & b \end{array} .$$

Puisque le tuple $\{C : 0, D : 1\}$ de $e_1$ est inclus dans 2 tuples de $e_2$, ce tuple n'est pas dans $e_1 \ominus e_2$. On obtient :

$$e_1 \ominus e_2 \begin{array}{|cc} C & D \\ \hline 0 & 0 \\ 1 & 1 \end{array} .$$

Le tuple $\{C : 0, D : 0\}$ de $e_1$ est dans $e_1 \ominus e_2$, car il est inclus dans aucun tuple de $e_2$. Le tuple $\{C : 1, D : 1\}$ de $e_1$ est dans $e_1 \ominus e_2$, car il est inclus dans seulement un tuple de $e_2$.

Démontrez que l'opérateur $\ominus$ est non-monotone.

---

$\boxed{\ldots/3}$

Soit $r$, $s_1$ et $s_2$ les relations suivantes :

$$r \begin{array}{|c} A \\ \hline a \end{array} \qquad s_1 \begin{array}{|cc} A & B \\ \hline a & 1 \end{array} \qquad s_2 \begin{array}{|cc} A & B \\ \hline a & 1 \\ a & 2 \end{array}$$

Clairement,

$$s_1 \subseteq s_2. \tag{2}$$

On a $r \ominus s_1 = \{\{A : a\}\}$, alors que $r \ominus s_2 = \{\}$. Donc,

$$r \ominus s_1 \nsubseteq r \ominus s_2. \tag{3}$$

Sur base de (2) et (3), il est correct de conclure que l'opérateur $\ominus$ est non-monotone.

---

**Question 6** On dénote par SPJRUD$\ominus$ l'algèbre que l'on obtient si l'on ajoute le nouvel opérateur $\ominus$ de la question 5 à l'algèbre SPJRUD. Cochez la case qui précède une expression correcte :

☒ Pour chaque expression $E$ en SPJRUD$\ominus$, il existe une expression $E'$ en SPJRUD telle que $E' \equiv E$. Le nouvel opérateur est donc redondant.

☐ L'algèbre SPJRUD$\ominus$ est plus expressive que SPJRUD.

Argumentez votre choix. Si vous avez cochez la première case, expliquez la construction de $E'$ à partir de $E$.

---

$\boxed{\ldots/10}$

On distingue deux cas.

**1. Cas où** $sorte(E_1) = sorte(E_2)$**.** Dans ce cas, $E_1 \ominus E_2 \equiv E_1$, car aucun tuple de $E_1$ peut apparaître dans plus d'un tuple de $E_2$ (car une relation est un ensemble sans doublons).

**2. Cas où** $sorte(E_1) \subsetneq sorte(E_2)$**.** Soit $sorte(E_2) \setminus sorte(E_1) = \{A_1, A_2, \ldots, A_n\}$ ($n \geq 1$).

On écrit d'abord une requête $G$ qui renvoie les tuples sur $sorte(E_1)$ qui sont inclus dans deux ou plusieurs tuples de $E_2$. Supposons $B_1, B_2, \ldots, B_n \notin sorte(E_2) \cup \{A_1, A_2, \ldots, A_n\}$.

Soit $F := E_2 \bowtie \rho_{A_1 \to B_1} (\rho_{A_2 \to B_2} (\cdots \rho_{A_n \to B_n} (E_2)))$.

Soit $G := \pi_{sorte(E_1)} (F - \sigma_{A_1 = B_1} (\sigma_{A_2 = B_2} (\cdots \sigma_{A_n = B_n} (F))))$.

On a $E_1 \ominus E_2 \equiv E_1 - G$.

Notez que le premier cas peut être obtenu on mettant $n = 0$. En effet, si $n = 0$, $F = E_2 \bowtie E_2 = E_2$ (car aucun attribut n'est renommé) et $G = F - F$, ce qui donne $E_1 \ominus E_2 \equiv E_1 - (F - F) \equiv E_1$. Il n'y a donc pas vraiment besoin de traiter le premier cas séparément.

**Question 7** Considérez le schéma-DF suivant (dénotons ce schéma-DF par $\delta$) :

$$(ABCD, \{A \to C, B \to C, C \to D, B \to D\})$$

Considérez l'ensemble suivant avec deux schémas-DF (dénotons cet ensemble par $\Delta$) :

$$\Big\{(AC, \{A \to C\}), \quad (BC, \{B \to C\}), \quad (CD, \{C \to D\})\Big\}$$

Cochez la case qui précède une expression correcte :

☒ $\Delta$ n'est pas une décomposition de $\delta$ parce que $\Delta$ ne préserve pas le contenu.

☐ $\Delta$ est une décomposition de $\delta$ qui préserve les dépendances fonctionnelles.

☐ $\Delta$ est une décomposition de $\delta$ qui ne préserve pas les dépendances fonctionnelles.

Argumentez votre choix.

---

…/5

Soit $\Sigma = \{A \to C, B \to C, C \to D, B \to D\}$.

Soit $r$ la relation suivante :

$$\begin{array}{c|cccc} r & A & B & C & D \\ \hline & a & 0 & c & d \\ & 1 & b & c & d \end{array},$$

avec $a \neq 1$ et $b \neq 0$. On a $r \models \Sigma$. Cependant, $r \not\models \bowtie[AC, BC, CD]$, parce que $\pi_{AC}(r) \bowtie \pi_{BC}(r) \bowtie \pi_{CD}(r)$ contient le tuple $\{A : a, B : b, C : c, D : d\}$ qui n'est pas dans $r$. Donc, $\Delta$ n'est pas une décomposition.

Voici un raisonnement pour arriver à une telle relation $r$ en utilisant la matière enseignée au cours. Le Théorème de Heath nous dit qu'à cause de $C \to D$, l'on peut décomposer (sans perte d'information) $ABCD$ en $CD$ et $ABC$. Pour arriver à $\Delta$, if faut ensuite encore décomposer $ABC$ en $AC$ et $BC$. Mais pour qu'une telle décomposition soit sans perte d'information, on doit avoir soit $C \to A$, soit $C \to B$ (soit les deux). On peut maintenant construire $r$ comme la plus petite relation qui ne satisfait ni $C \to A$, ni $C \to B$, mais qui satisfait $\Sigma$.

Voici une autre réflexion, plus indirecte, qui mène à une réponse correcte à cette question. Si l'on rend $\Sigma$ irréductible, l'on obtient $\{A \to C, B \to C, C \to D\}$. L'algorithme, vu au cours, pour arriver à une décomposition en 3NF qui préserve les DFs trouve quatre composants : $AC$, $BC$, $CD$, $AB$. Noter qu'$AB$ est le composant qui constitue une clé de $\delta$ et qui a pour but d'assurer que la décomposition soit sans perte d'information. Ce composant est absent dans $\Delta$, ayant pour conséquence que la décomposition perd l'information sur les combinaisons "correctes" des valeurs pour $A$ et $B$. Notamment, pour la relation $r$ ci-dessus, $\pi_{AC}(r) \bowtie \pi_{BC}(r) \bowtie \pi_{CD}(r)$ contient un tuple qui combine $A : a$ avec $B : b$, une combinaison qui n'existe pas dans $r$.

**Question 8** Soit $\varphi = \exists x\,(R(x) \to \forall y\,(R(y)))$, une formule en logique des prédicats. Soit $\mathcal{I}$ une instance de base de données qui contient la relation suivante :

$$\begin{array}{c|c} R & A \\ \cline{2-2} & 1 \\ & 2 \end{array}.$$

Cochez la case qui précède une expression correcte :

☒ La formule $\varphi$ est vraie dans $\mathcal{I}$.

☐ La formule $\varphi$ est fausse dans $\mathcal{I}$.

☐ Puisque $\varphi$ n'est pas *domain independent*, il n'est pas possible de savoir si $\varphi$ est vraie ou fausse dans $\mathcal{I}$ sans connaître le domaine d'interprétation $\mathbf{dom}$.

Argumentez votre choix.

---

$\boxed{\ldots/7}$

Deux cas sont possibles :
- si $\mathbf{dom} = \{1, 2\}$, alors $\forall y\,(R(y))$ est vraie, et donc $\varphi$ est vraie (car $p \to q$ est vrai si $q$ est vrai) ;
- si $\mathbf{dom}$ contient un élément $x$ qui n'est pas dans $R$ (i.e., $R(x)$ est faux), alors $\varphi$ est vraie (car $p \to q$ est vrai si $p$ est faux).

Donc, $\varphi$ est vraie dans $\mathcal{I}$.

Notez que un même raisonnement s'applique pour n'importe quel contenu de la relation $R$. Donc, $\varphi$ est vraie dans toute instance $\mathcal{I}$.

Une autre façon pour répondre à cette question est d'observer que $\varphi$ est une tautologie :

$$\begin{aligned} \varphi &\equiv \exists x\,(\neg R(x) \lor \forall y\,(R(y))) \\ &\equiv \exists x\,(\neg R(x)) \lor \forall y\,(R(y)) \\ &\equiv \exists x\,(\neg R(x)) \lor \neg\exists y\,(\neg R(y)) \\ &\equiv \exists y\,(\neg R(y)) \lor \neg\exists y\,(\neg R(y)) \\ &\equiv q \lor \neg q \text{ avec } q = \exists y\,(\neg R(y)) \end{aligned}$$

Puisque $q \lor \neg q$ est une tautologie, $\varphi$ est une tautologie.

**Question 9** Soient

$$\begin{aligned}\mathcal{A} &= ABCDEFG\\ \Sigma &= \{A \to B, ABC \to D, ABCE \to FG, BCDF \to AG, DEF \to C, F \to E\}\end{aligned}$$

Cochez la case qui précède une expression correcte :

☐ Le schéma $(\mathcal{A}, \Sigma)$ est en BCNF.

☒ Le schéma $(\mathcal{A}, \Sigma)$ est en 3NF mais n'est pas en BCNF.

☐ Le schéma $(\mathcal{A}, \Sigma)$ n'est pas en 3NF.

Détaillez les arguments qui mènent à cette conclusion.

---

…/15

Le schéma $(\mathcal{A}, \Sigma)$ n'est pas en BCNF, car $\Sigma$ contient $A \to B$, mais $\Sigma \not\models A \to C$.

Pour déterminer si le schéma est, oui ou non, en 3NF, il est utile de "simplifier" $\Sigma$ en utilisant les deux règles suivantes :

1. $\{X \to A_1 A_2 \cdots A_n\} \equiv \{X \to A_1, \quad X \to A_2, \quad \ldots, \quad X \to A_n\}$ ;
2. $\{XYA \to B, \quad Y \to A\} \equiv \{XY \to B, \quad Y \to A\}$.

Cette simplification nous donne l'ensemble $\Sigma'$ suivant, tel que $\Sigma' \equiv \Sigma$ :

$$\Sigma' = \left\{ \begin{array}{l} A \to B, AC \to D, ACE \to F, ACE \to G, \\ BDF \to A, BDF \to G, DF \to C, F \to E \end{array} \right\}.$$

*Rappelons en passant que $(\mathcal{A}, \Sigma)$ est en 3NF si (et seulement si) pour chaque DF $X \to A'$ dans $\Sigma'$, soit $X$ inclut une clé, soit $A'$ est premier (soit les deux). Ce rappel de 3NF est correct puisque toute DF dans $\Sigma'$ est singulière.*

On peut vérifier que $ACE$ est une clé, parce que $(ACE)^{*,\Sigma} = \mathcal{A}$, mais $E \notin (AC)^{*,\Sigma}$, $C \notin (AE)^{*,\Sigma}$, et $A \notin (CE)^{*,\Sigma}$. Une autre clé est $BDF$. En conclusion, $A$, $B$, $C$, $D$, $E$ et $F$ sont des attributs premiers.

Aucune clé ne contient $G$. En effet, si $(XG)^{*,\Sigma} = \mathcal{A}$ pour un ensemble $X$ qui ne contient pas $G$, alors $(X)^{*,\Sigma} = \mathcal{A}$. Ceci est vrai parce que $G$ apparaît dans une DF, mais n'apparaît pas à gauche d'une flèche.

*Notez en passant qu'un attribut qui n'apparaît pas à gauche d'une flèche peut quand-même être premier. Par exemple, le schéma $(IJK, \{I \to J\})$ a $IK$ comme clé, alors que $K$ n'apparaît pas à gauche d'une flèche.*

$\Sigma'$ ne contient que deux DFs avec un attribut non-premier à droite de "$\to$", notamment $ACE \to G$ et $BDF \to G$. Ces DFs n'engendrent pas de violation de 3NF, car à gauche de "$\to$", on trouve les clés $ACE$ et $BDF$. En conclusion, $(\mathcal{A}, \Sigma)$ est en 3NF.

*Notez en passant que deux autres clés sont $ACF$ et $ADF$. Cependant, pour conclure que le schéma est en 3NF, on n'a pas besoin de connaître les clés $ACF$ et $ADF$.*

**Question 10** Considérez l'exécution suivante :

$$R_1(A)R_2(B)R_3(B)R_2(A)R_1(B)W_3(B)W_2(A)W_1(C)$$

Est-ce que cette exécution est possible en 2PL ? Complétez l'exécution avec des demandes de verrous ou argumentez pourquoi cette exécution n'est pas possible en 2PL.

…/10

| | | |
|---|---|---|
| $S_1(A)$ | | |
| $S_1(B)$ | | |
| $X_1(C)$ | | |
| $R_1(A)$ | | |
| $U_1(A)$ | | |
| | $X_2(A)$ | |
| | $S_2(B)$ | |
| | $R_2(B)$ | |
| | $U_2(B)$ | |
| | | $S_3(B)$ |
| | | $R_3(B)$ |
| | $R_2(A)$ | |
| $R_1(B)$ | | |
| $U_1(B)$ | | |
| | | $X_3(B)$ |
| | | $W_3(B)$ |
| | | $U_3(B)$ |
| | $W_2(A)$ | |
| | $U_2(A)$ | |
| $W_1(C)$ | | |
| $U_1(C)$ | | |