****

**INF8410 – Ontologie et web sémantique**

**Automne 2021**

**TP No. 1**

**Groupe 1**

**1803486 – Sofia Marsou**

**1909765 – Bouchra Dahamni**

**2164296 – Virgil Retault**

**Soumis à : Jean Simon Marrocco**

**03-10-2021**

**2. Partie 1**

**2.1 Analyse de données RDF**

Décrivez, en vos propres mots et le plus précisément possible, ce qui est décrit dans ce graphe, et discutez de la qualité de cette représentation. Proposez, s'il y a lieu, une version améliorée. Pour vous aider dans votre discussion, prenez en compte les considérations suivantes (notez qu'il peut y en avoir d'autres qui ne sont pas listées ici):

Ce graphe montre un livre B1 qui est écrit en langue Anglaise. Ce livre est une traduction du livre “Les joies du web sémantique” écrit en langue Française. Le livre B1 possède deux auteurs qui sont Michelle Tremblay qui est une personne qui a comme occupation écrivain et politicien et Michel Gagnon. Il y a une note attribuée au livre B1 qui a comme valeur 9. Cette représentation est de mauvaise qualité pour plusieurs raison qui sont détaillé ci-dessous.

* **Fait-on une bonne utilisation des vocabulaires disponibles?**

Il n’y a pas une bonne utilisation des vocabulaires disponibles. En effet, l’utilisation des URI présente n’est pas maximisée. Pour maximiser leur utilisation, il est possible de:

* Remplacer le prédicat :langue à inlanguage de schema.org
* Ajouter une URI pour les langues Anlgais et Français
* Il aurait été intéressant d’utiliser une URI au lieu d’un littéral pour "Michel Gagnon" et "Michelle Tremblay" pour éviter l’ambiguité de personne, bien qu’il soit difficile de trouver ce nom commun.
* Il aurait été intéressant d'utiliser une URI pour le titre “Les joies du web sémantique” pour éviter les ambiguité de livre.
* Utiliser translationOfWork ou interpretedAsClain de schema.org au lieu de derivedFrom.
* Utiliser hasOccupation de schema.org au lieu de occupation

• **Y a-t-il une meilleure manière de structurer les informations en RDF?**

Il y a plusieurs manières de structurer les informations en RDF. Dans un premier temps, il est préférable d’éviter l’utilisation de liste s’il n’y a pas d’utilité à le faire. En effet, il n’y a que deux auteurs dans la liste, il est donc préférable de l’enlever pour éviter de complexifier la maintenance et la mise à jour des données. Aussi, on pourrait spécifier un attribut langue pour spécifier la langue des littéraux et leur donné un type lorsque nécessaire. De plus, le fait d’ajouter des URI permet d’éviter l'ambiguïté et de s’assurer de la consistance du vocabulaire. Le fait de changer le triplet Rating-attributedTo-B1 à B1-contentRating-Rating rend le graphe plus lisible et offre une plus faible charge cognitive.

• **Y a-t-il des informations pertinentes qui sont manquantes?**

Il y a plusieurs informations pertinentes qui manquent. En effet, il manque l’occupation de Michelle Gagnon et le fait que c’est une personne. Cependant, cela peut être inféré, car puisqu’il est l’auteur du livre, il doit avoir une occupation d’écrivain qui lui-même est considéré comme une personne. Aussi, il manque l’objet du triplet sujet-prédicat-objet dans B1-schema:translator. Ainsi, on ne sait pas ce que représente T1.

**• Sera-t-il facile de maintenir et mettre à jour ces données?**

Il serait difficile de mettre à jour ses données. Dans un premier temps, du fait que le rating pointe vers le livre B1, il est difficile de trouver rating associé à B1. En effet, il faut parcourir tous les nœuds et s’avoir s’il pointe vers B1 pour confirmer qu’il s’agit d’un rating associé à B1, ce qui est long et coûteux en performance. Il aurait été préférable de faire mettre la flèche de B1 vers le rating pour ainsi trouver tous les rating associé à B1 seulement à partir du nœud B1. Dans un deuxième temps, il est très difficile de mettre à jour des données, du fait qu’il y a une liste pour représenter les auteurs de B1. Pour ajouter ou supprimer un auteur associé à B1, il faut parcourir toute la liste, ce qui est long et coûteux en performance.

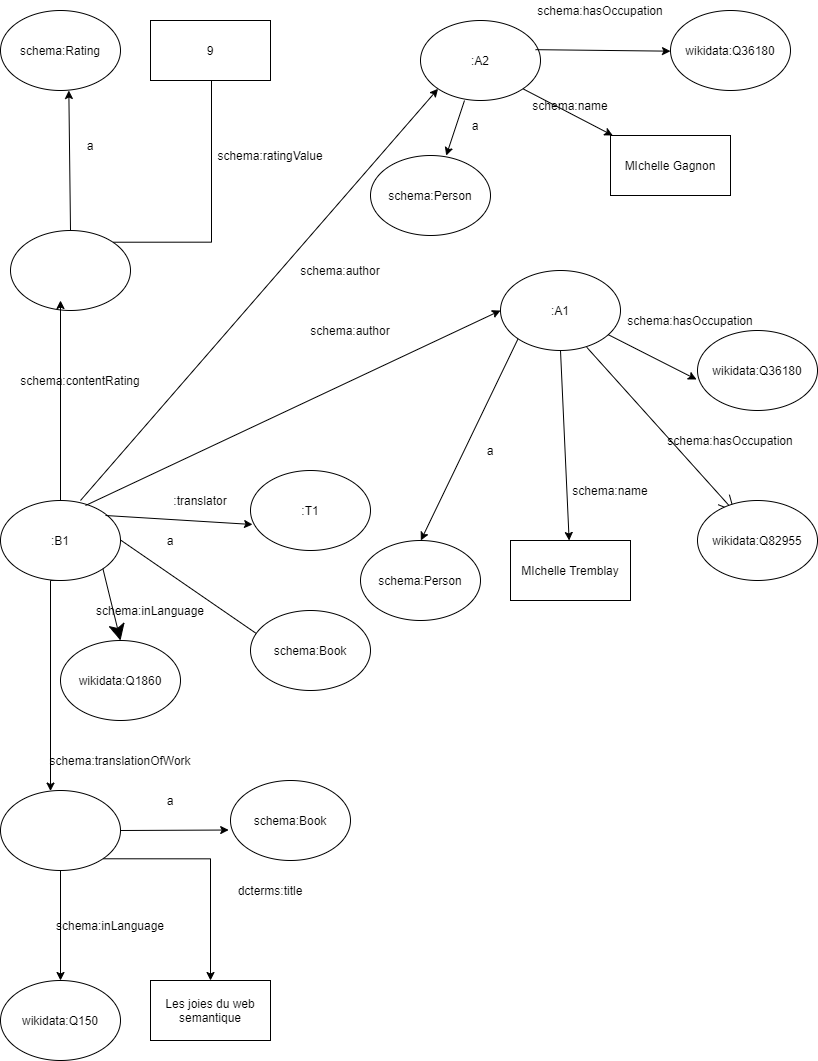
**• Quelles sont les conséquences sur le genre de requête qu'on peut faire?**

Il y a plusieurs conséquences sur le genre de requête qu’on peut faire. Premièrement, il est très long et coûteux en performance de déterminer qu’elles sont les rating associé à B1, car il faut parcourir tous les nœuds et confirmer qu’ils pointent bien vers B1. Deuxièmement, lorsqu’on veut ajouter ou supprimer ou modifier un auteur qui a écrit le livre B1, il faut parcourir toute la liste, ce qui est long et peu performant.

**La version améliorée du graphe : (code dans lab1-1.ttl)**

****

**2.2 RDF vers graphe**



**2.3 Description textuelle vers RDF (code dans lab1-2.ttl lab1-2.json et lab1-2.xml)**

**Turtle :**



**JSON :**



**XML :**



**3. Partie 2**

Le jeux d’échec :

Lien vers l’ontologie en ligne : [https://g33kex.github.io/ChessRDF/Index.xml](https://g33kex.github.io/ChessRDF/Index.xml?fbclid=IwAR0vaZBRoTXTRGddoyN3GMRzkb3Qn2VBUAhcQDjBJ1dRWucESe5gThQV9ek)

L’ontologie que nous avons choisi décrit le jeu d'échecs. Elle permet de décrire le déroulement d’une partie d’échec et de répondre à plusieurs questions pertinentes comme Quel était le troisième coup de telle partie? Comment s'est finie une telle partie ? Quelle est la position du plateau a la fin d'une partie? Est ce que le pion blanc en b4 est protégé? Quelle est l'ouverture d’une certaine partie?

Pour créer cette ontologie, il fallait trouver toutes les classes qui permettent d’expliquer en globalité le fonctionnement d’une partie d’échec. Nous avons retenu comme classe Pièce qui englobe toutes les pièces associées aux jeux. Chaque pièce du jeu est dans une classe spécifique (Knight, Pawn, Queen, King, Rook, Bishop). La classe Pièce et toutes les classes qui représentent des classes spécifiques se trouvent déjà sur wikidata. Nous n’avons pas directement remplacé les classes par celle déjà existante sur wikidata. Nous avons plutôt utilisé un sameAs pour dire que nôtre classe est similaire à celle de wikidata, mais qu’elle possède plus de propriétés. Nous avons mis la classe Square pour représenter une case avec des propriétés pour indiquer si une pièce se trouve sur la case (hasPiece), la case à gauche de la case courante (left), la case à droite (right), la cause au-dessus (top) et la case au-dessous (bottom). De plus, la case contient une coordonnée précise comme celle d’une vraie partie d’échec. Ill y a aussi la classe Player qui permet de décrire un joueur de la partie. Il y a la classe Game qui décrit la partie et qui contient une propriété pour indiquer le joueur blanc (whitePlayer) et le joueur noir (blackPlayer) et une propriété pour indiquer tous les mouvements de la partie (moves). Il y a la classe qui décrit une séquence de mouvements (Séquence) qui contient la propriété startingBoard pour indiquer les positions initiales des pièces. La classe Séquence dérive de rdf:List car une séquence de mouvements est représentée par l’état initial du plateau et une liste de mouvements. Cela permet de définir des séquences de mouvements qui ne sont pas une partie entière, par exemple pour définir ce qu’est l’ouverture de la défense sicilienne. La classe Board permet de décrire l’échiquier. La classe qui permet de décrire un certain mouvement dans la partie est la classe Move. Cette classe contient des propriétés pour indiquer de quel type de mouvement il s’agit, par exemple un castle ou l’avancée d’un pion. Elle contient aussi une propriété pour indiquer si le mouvement amène à un mouvement d’échec ou d’échec et mat ou bien qu’elle sont les pièces prises lors d’un certain mouvement.

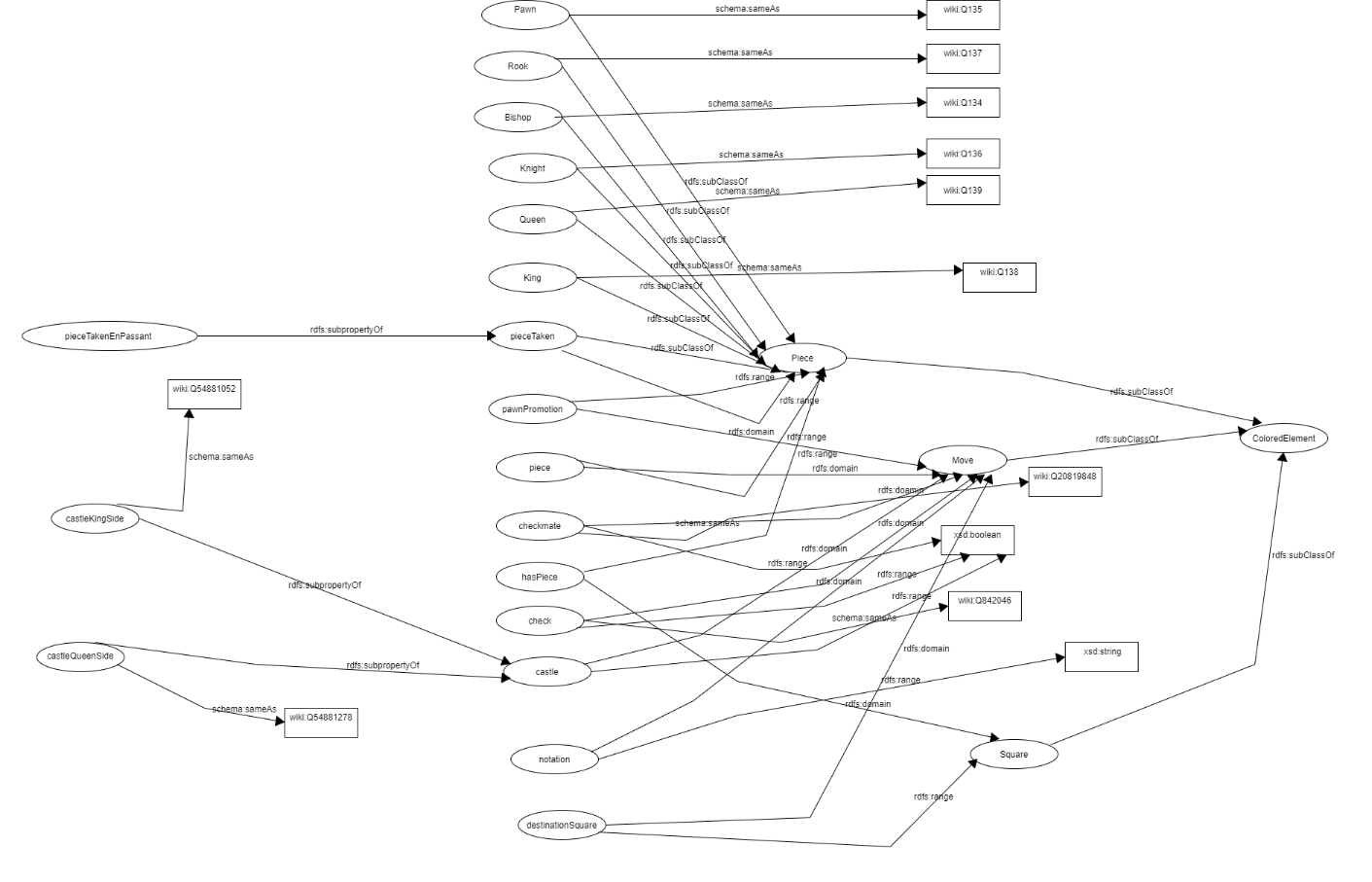
Nous avons aussi utilisé des graphes nommés avec la syntaxe TriG en considérant que deux graphes nommés dans un même fichier partagent leurs nœuds vides. Cela permet par exemple dans le fichier Example.ttl de faire référence à un plateau d'échecs de départ lorsqu’on décrit une séquence de mouvement. Le plateau est alors un graphe qui a comme type Board.

Pour créer les instances, nous avons utilisé un programme de python qui va créer toutes les instances possibles d’un jeu d'échecs selon notre ontologie. Le programme defaultBoard.py génère le graphe StartingBoard.ttl qui représente le plateau de départ du jeu d’échecs avec toutes les pièces aux positions de départ. Pour le prochain TP, nous développerons aussi un programme python qui permet de traduire une séquence de mouvements au format PGN en une séquence de mouvements dans notre ontologie pour pouvoir importer des parties. Des instances d’exemple sont aussi disponibles dans le fichier Example.ttl, notamment un plateau avec seulement 6 cases, et une séquence de fin de jeu avec un mat en deux coups.

Nôtre ontologie est très utile pour décrire avec précision ce qui c’est passé durant une certaine partie. On peut aussi bien répondre à des questions sur la position des pièces durant la partie, qu’à des questions sur le types de mouvements effectués. Cette ontologie nous permettra d’effectuer des requêtes pour répondre à des questions particulières de partie et même créer notre propre jeu d'échecs. On peut imaginer une intelligence artificielle qui joue aux échecs qui se base sur cette ontologie pour prendre des décisions, par exemple en définissant quelles pièces sont attaqués, ou encore quelles pièces de l’adversaires sont mal défendues, pour l'amener à une compréhension du jeu plus profonde que de la simple force brute de test de positions. On pourrait aussi imaginer une exploration par l’algorithme de Monte Carlo sur un arbre de jeu ontologique, ce qui pourrait permettre de prioriser certains chemins.

Notre ontologie décrit les parties d’échecs, et nous nous sommes rendu compte que l’équipe de Clara travaillait sur les compétitions d’échecs. Nous avons donc décidé de nous coordonner pour pouvoir lier nos deux ontologies. Ainsi la classe :Game de leur ontologie dérive de notre classe chess:Game, ce qui leur permet de représenter les coups d’une partie d’un match de compétition à l’aide de notre ontologie. Ce lien permettra par exemple de répondre à des questions telles que: quelle est l’ouverture qui a permis de gagner le plus de parties à l’US Open en 1999? Ou encore quelle est la pièce qui est la plus utilisée pour effectuer un mat par Kasparov?

## La graphe des différentes pièces et les mouvements :



## Le graphe du jeu et de l'échiquier :

