Jour 13 – La récursion sélective

Partons de la requête décrite hier, qui trouve les randonnées passant par une rue :

```
area["ref:INSEE"=44005];
way[name="Avenue Arthus-Princé"](area);
<;
rel._[route=hiking];
out geom;
```

Cette requête applique la récursion ascendante avec < puis un filtre de tags sur les relations. Ces deux opérations peuvent être combinées en une seule avec la syntaxe suivante :

```
area["ref:INSEE"=44005];
way[name="Avenue Arthus-Princé"](area);
rel(bw)[route=hiking];
out geom;
```

L'instruction `rel(bw)` produit les **relations référençant les ways** du lot de données en entrée. Il s'agit d'une *récursion ascendante sélective*, où le mot-clef bw signifie “backward from ways”. Il est bien sûr possible d'utiliser des variables :

```
area["ref:INSEE"=44005];
way[name="Avenue Arthus-Princé"](area)->.rues;
rel(bw.rues)[route=hiking];
out geom;
```

Oui “.rues” au pluriel : n'oublions pas qu'on manipule des lots de données (Overpass est un langage de programmation ensembliste), et que les rues sont souvent découpées en de multiples sections dans OpenStreetMap.

Il existe plusieurs **filtres de récursion**, 4 pour la récursion descendante :

- `node(w)` : nodes référencés par les ways
- `node®` : nodes référencés par les relations
- `way®` : ways référencés par les relations
- `rel®` : relations référencées par les relations

et 4 pour la récursion ascendante :

- `way(bn)` : ways référençant les nodes
- `rel(bn)` : relations référençant les nodes
- `rel(bw)` : relations référençant les ways
- `rel(br)` : relations référençant les relations

La récursion sélective est particulièrement utile pour les données de **transport en commun**. Prenons par exemple une ligne de bus à Nantes. Nous pouvons obtenir les arrêts de la ligne C2, c'est-à-dire les nodes référencés par la relation :

```
rel[route=bus][operator=SEMITAN][ref="C2"];
node(r);
out;
```

Notez qu'on obtient les arrêts et trajets dans les deux sens de la ligne, car nous partons en réalité de deux relations, une pour chaque sens. Ajouter un filtre sur les tags from et/ou to permet de sélectionner un trajet de la ligne.

En remplaçant `node(r)` par `way(r)`, on obtient les ways référencés qui forment le trajet de la ligne. Mais observons le résultat avec cette ligne de tramway :

```
rel[route=tram][operator=SEMITAN][ref="2"];
way(r);
out geom;
```

On obtient en fait **les rails et les quais** de la ligne. Ces quais sont référencés par la relation avec le ***rôle*** *platform*. Les filtres de récursion permettent également de filtrer sur le rôle des membres. Cet exemple retourne les quais de la ligne, c'est-à-dire les ways références avec le rôle platform :

```
rel[route=tram][operator=SEMITAN][ref="2"];
way(r:"platform");
out geom;
```

Si on veut obtenir les points d'arrêts (nodes) **et** les quais (ways), nous devons appliquer une union donc utiliser une variable :

```
rel[route=tram][operator=SEMITAN][ref="2"]->.ligne;
(
  node(r.ligne:"stop");
  way(r.ligne:"platform");
);
out geom;
```

Pour obtenir le trajet, il est possible de filtrer sur les membres dont le rôle est vide :

```
rel[route=tram][operator=SEMITAN][ref="2"]->.ligne;
way(r.ligne:"");
out geom;
```

On peut également filtrer sur le rôle dans une récursion ascendante, ce qui permet de trouver les lignes passant par un arrêt de bus :

```
node
  [highway=bus_stop]
  [network=TAN]
  [name="50 Otages"];
rel(bn:"stop")
  [type=route];
out geom;
```

Notez l'application du filtre de tag [type=route] qui permet d'éliminer la relation stop_area.

Exercices

- Trouvez toutes les adresses (numéros) de votre rue.
- Trouvez les lignes de tansport en commun passant par l'arrêt le plus proche de chez vous.
- Trouvez les arrêts desservis par ces lignes.

© CC-by-sa Carto'Cité

From:
http://wiki.cartocite.fr/ - CartoCité

Permanent link:
http://wiki.cartocite.fr/doku.php?id=tutoverpass:jour_13_la_recursion_selective

Last update: **2021/01/25 17:14**