

Language for RegBL Quality Constraints

Defining quality through logic — a declarative approach to data validation

Loïc Herman

6 mars 2025

HEIG-VD — PLM

1. Contexte

- 1.1 Depuis les données des registres fédéraux
- 1.2 Vers une formalisation de contraintes de qualités
- 1.3 La programmation logique sous contraintes

2. Langage

- 2.1 Grammaire formelle

3. Mise en pratique dans l'industrie

- 3.1 Projet SIBAT

Contexte

Depuis les données des registres fédéraux

Dans le cadre du registre fédéral des bâtiments et des logements (RegBL), l'Office fédéral de la statistique définit 7 entités et ~ 800 caractères (attributs) sur ces entités¹. Cela comprend :

- **Attributs obligatoires** : identifiants uniques, coordonnées, adresses
- **Attributs structurels** : nombre d'étages, surface habitable, type de chauffage
- **Attributs temporels** : dates de mise à jour, historique des modifications
- **Relations spatiales** : commune, quartier, parcelle cadastrale

1. <https://www.housing-stat.ch/catalog/fr/4.2/final>

Depuis les données des registres fédéraux

Ces données jouent un rôle essentiel dans la gestion des tâches légales de la confédération à l'ère du numérique.

- **Cadre légal** : *Gestion commune des données de référence de la Confédération*
- **Obligation** : Usage des données de référence à tous les niveaux (cantonal, communal, et fédéral)
- **Avantage clé** : Élimination des contradictions entre autorités
- **Principe once-only** : La collecte de données par l'administration est effectuée qu'une seule fois auprès de la même personne

Vers une formalisation de contraintes de qualités

La définition d'un tel ensemble d'attributs vient avec un besoin d'en valider la qualité, tant en typage qu'en termes relationnels.

L'OFS définit donc un ensemble de ~ 500 règles de qualité sous différentes classes :

- **Automatismes** : hors-scope, règles devant démarrer une action automatique
- **Obligation d'annonce** : information manquante selon un critère spécifique
- **Exigence de qualité** : validations de cohérence et contraintes d'intégrités
- **Spécification technique** : définitions formelles du domaine des attributs

Vers une formalisation de contraintes de qualités

Pour simplifier la compréhension, l'OFS définit pour chaque règle un pseudo-code comparable à du code SQL permettant d'avoir une version déclarative des règles.

Listing 1 – Pseudocode for rule BQ9980

```
GSTAT=1007  
WHERE GABBJ NOT NULL
```

L'idée du langage implémenté s'inspirera du format du pseudo-code pour proposer un langage déclaratif facilement éditable par les gestionnaires d'applications permettant d'offrir une couverture satisfaisante des contraintes avec une retranscription minime.

La programmation logique sous contraintes

Paradigme de programmation déclarative où :

- Les relations entre variables sont exprimées sous forme de contraintes
- Le programme spécifie *ce qui* doit être calculé, pas *comment*
- Les contraintes définissent des propriétés que la solution doit satisfaire

Formellement, on parle d'un problème sous forme d'un triplet $\langle X, D, C \rangle$, où :

$X = \{x_1, x_2, \dots, x_n\}$ est l'ensemble des variables (1)

$D = \{D_1, D_2, \dots, D_n\}$ est l'ensemble des domaines (2)

$C = \{c_1, c_2, \dots, c_m\}$ est l'ensemble des contraintes (3)

Une solution est une affectation $\sigma : X \rightarrow \bigcup_{i=1}^n D_i$ telle que $\forall c_j \in C, \sigma \models c_j$

Application à la validation d'attributs

1. **Modélisation** : Tout attribut i du registre devient un fait, une variable $x_i \in X$
2. **Domaines** : Chaque attribut i a un domaine de valeurs valides D_i
3. **Règles de qualité** : Exprimées comme contraintes C entre attributs
4. **Validation** : Un enregistrement est valide \Leftrightarrow toutes les contraintes sont satisfaites

Par exemple, pour les bâtiments :

$\forall b \in \text{Bâtiments} : \text{dateConstruction}(b) \leq \text{dateDémolition}(b)$ (BQ7447)

$\forall b \in \text{Bâtiments} : \exists \text{dateDémolition}(b) \Rightarrow \text{statut}(b) = \text{démoli}$ (BQ9980)

Langage

Le langage souhaité doit donc remplir les critères suivants :

- Permet de modéliser un ensemble de règles représentant des contraintes, éventuellement dépendantes, sur un ou plusieurs attributs d'un registre (inspiré par la programmation logique sous contraintes),
- Peut facilement être modifié par des personnes aisées en informatique, mais pas nécessairement développeurs (sémantiquement équivalent à un langage connu, comme le SQL),
- Permet d'exprimer facilement un ensemble de conditions sur un attribut, proposant une syntaxe simplifiable,
- Détecte si des règles arithmétiques se chevauchent inutilement.

Une grammaire formelle complète sera définie lors du rendu intermédiaire.

La structure générale du langage est toutefois définie formellement comme suit :

- Le *Program* ne peut être constitué que d'un seul *Statement*
- Un *Statement* est constitué d'une *Expression*, éventuellement complémentée d'une condition sous forme d'*Expression*
- Une *Expression* modélise une ou plusieurs contraintes (*Condition*) ou relations logiques (égalité, vérité) sur un attribut
- Les contraintes sont des *Condition* qui peuvent également être combinées au sein d'une même *Expression*

Le *Program* représente alors la production de plus haut niveau pour la représentation d'une règle à évaluer.

Grammaire formelle

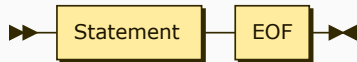


Figure 1 – Définition du *Program*



Figure 3 – Définition du *Statement*

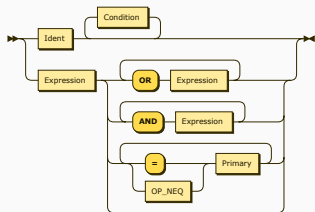


Figure 2 – Définition des *Expression*

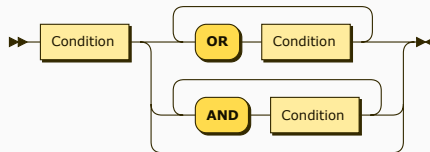


Figure 4 – Définition des *Condition*

Mise en pratique dans l'industrie

Pour assurer la gestion de la collecte des données auprès des communes vaudoises, l'État de Vaud développe le Système d'information des Bâtiments (SIBAT).

Toutes les communes du canton peuvent ensuite entrer, via le portail des prestations, des « mutations » pour un ou plusieurs bâtiments dont elles ont accès.

Les mutations seront ensuite traitées par le **moteur de validation** établi à partir des **contraintes** retranscrites depuis le format fourni par l'OFS.

Si toutes les règles de qualités sont acceptées, les mutations peuvent ensuite être envoyées dans le RegBL.

Questions