Spreadsheet Column Type Inference

Philip Blair

10 June 2016

This paper describes the methodology used by Pyret[1] to infer the data types of columns in the language's upcoming `gdrive-sheets` library.

1 Cell Types

Google Sheets gives us the ability to infer the following cell types[2]:

- `String`
- `Number`
- `Bool`
- `null` (AKA `None`)

Furthermore, cells of type `Number` can be specially formatted in one of the following ways:

- `TEXT`
- `NUMBER`
- `PERCENT`
- `CURRENCY`
- `DATE`
- `TIME`
- `DATE_TIME`
- `SCIENTIFIC`
- `NUMBER_FORMAT_TYPE_UNSPECIFIED` (implicit)

Since Pyret currently does not have a `datetime` type, I figure that it is currently best to leave the `TIME`,`DATE_TIME`,`DATE`, and (obviously) `TEXT` formats as string upon opening a spreadsheet.

$$\begin{aligned} c_i &::= \text{The } i\text{th column} \\ v_i &::= \text{Value in column } c_i \\ \tau_i &::= \text{Type (differentiated by index } i) \\ [c_i : \tau_i, \ldots] &::= \text{Schema store; “Column } i \text{ has type } \tau_i\text{”} \\ [] &::= \text{Empty Schema Store (starting point)} \end{aligned}$$

Figure 1: Inference Notation

2 Inferring Column Types

From these types, I propose inferring column types using the rules in Figure 2 (see Figure 1 for notation). Note that each inference produces a new schema store.

While it may not be the *most* robust way of doing this inference, I believe that it will be plenty sufficient for our use case.

References

[1] Brown University PLT Group. Pyret. `http://pyret.org`, 2016. [Online; accessed 10-June-2016].

[2] Google Inc. Collection spreadsheets — Sheets API. `https://developers.google.com/sheets/reference/rest/v4/spreadsheets`, 2016. [Online; accessed 10-June-2016].

$$\frac{v_i : \tau_i}{[] \vdash v_i \Rightarrow [c_i : \tau_i]} \qquad \text{(T-Intros)}$$

$$\frac{v_i : \tau_i \qquad \tau_i \neq \texttt{None}}{[c_i : \texttt{None}] \vdash v_i \Rightarrow [c_i : \texttt{Option<}\tau_i\texttt{>}]} \qquad \text{(T-Option-1)}$$

$$\frac{v_i : \texttt{None} \qquad \tau_i \neq \texttt{None} \neq \texttt{Option<}\tau_j\texttt{>}\ (\forall j)}{[c_i : \tau_i] \vdash v_i \Rightarrow [c_i : \texttt{Option<}\tau_i\texttt{>}]} \qquad \text{(T-Option-2)}$$

$$\frac{(v_i : \texttt{None}) \vee (v_i : \tau_i)}{[c_i : \texttt{Option<}\tau_i\texttt{>}] \vdash v_i \Rightarrow [c_i : \texttt{Option<}\tau_i\texttt{>}]} \qquad \text{(T-Option-3)}$$

$$\frac{v_i : \texttt{None}}{[c_i : \texttt{None}] \vdash v_i \Rightarrow [c_i : \texttt{None}]} \qquad \text{(T-None)}$$

$$\frac{v_i : \tau_i}{[c_i : \tau_i] \vdash v_i \Rightarrow [c_i : \tau_i]} \qquad \text{(T-Check)}$$

$$\frac{v_i : \tau_j \qquad \tau_j \neq \tau_i \neq \texttt{None}}{[c_i : \texttt{Option<}\tau_i\texttt{>}] \vdash v_i \Rightarrow \texttt{ERROR}} \qquad \text{(T-Error-1)}$$

$$\frac{v_i : \tau_j \qquad \tau_j \neq \tau_i \neq \texttt{None}}{[c_i : \tau_i] \vdash v_i \Rightarrow \texttt{ERROR}} \qquad \text{(T-Error-2)}$$

Figure 2: Schema Inference Rules