Algorithm for finding *non-ndg* locations

INPUT: TRS $\mathfrak{R}$
OUTPUT: Finite set of locations $F(\mathfrak{R})$

$$LEFT_0(\mathfrak{R}) := \left\{ (\alpha, L, \tau) \mid \alpha = l \rightarrow r \in \mathfrak{R},\ \tau \in pos(l) \setminus \{\varepsilon\},\ l|_\tau \in D \right\},$$

$$RIGHT_0(\mathfrak{R}) := \left\{ (\alpha, R, \tau) \mid \alpha = l \rightarrow r \in \mathfrak{R},\ \tau, \pi \in pos(r),\ \tau \neq \pi,\ r|_\tau = r|_\pi \in V \right\},$$

$\forall\, i \in N_{>0}$:

$$\begin{aligned} LEFT_i(\mathfrak{R}) := \{ & (\alpha, L, \tau) \mid \alpha = l \rightarrow r \in \mathfrak{R},\ \tau \in pos(l),\ \pi \in pos(r), \\ & l|_\tau = r|_\pi \in V,\ (\alpha, R, \pi) \in RIGHT_{i-1}(\mathfrak{R}) \} \\ \cup\ & LEFT_{i-1}(\mathfrak{R}) \end{aligned}$$

,

$$\begin{aligned} RIGHT_i(\mathfrak{R}) := \{ & (\alpha, R, \tau) \mid \alpha = l \rightarrow r \in \mathfrak{R},\ \tau \in pos(r),\ \tau = \pi.\upsilon, \\ & \alpha \neq \beta = s \rightarrow t \in \mathfrak{R},\ \exists\, MGU\ for\ \mathfrak{R}|_{(\alpha,R,\pi)}\ and\ \mathfrak{R}|_{(\beta,L,\varepsilon)}, \\ & \upsilon \nparallel \omega \in pos(s),\ \exists\, (\beta, L, \omega) \in LEFT_{i-1}(\mathfrak{R}) \} \\ \cup\ & \{ (l \rightarrow r, R, \tau) \mid \exists (\beta, P, \pi) \in M_{i-1}(\mathfrak{R}),\ root(l) = root(\mathfrak{R}|_{(\beta,P,\pi)}) \in D,\ \tau \in pos(r) \} \\ \cup\ & RIGHT_{i-1}(\mathfrak{R})\ . \end{aligned}$$

$$M_i(\mathfrak{R}) := LEFT_i(\mathfrak{R}) \cup RIGHT_i(\mathfrak{R})$$

$$M(\mathfrak{R}) := \bigcup_{i \in N} M_i(\mathfrak{R}) = LEFT_n(\mathfrak{R}) \cup RIGHT_m(\mathfrak{R}),\ for\ some\ n, m \in N$$

$$F(\mathfrak{R}) := M(\mathfrak{R}) \setminus \left(\left\{ (l \rightarrow r, L, \tau) \mid root(l|_\tau) \in V \right\} \cup \left\{ (\alpha, P, \tau) \mid root(\mathfrak{R}|_{(\alpha,P,\tau)}) \in C \right\} \right)$$

Algorithm for tagging locations

INPUT: TRS $\mathfrak{R}$, $M(\mathfrak{R})$
OUTPUT: Finite set of locations $F(\mathfrak{R})$

The aim of these tags is to check which variables locations receive defined symbols as input. We want to know this, because variables that are proven to only receive constructor terms need not be encoded and thus can be removed from the set F($\mathfrak{R}$).

The approach of the tagging is as follows:

- First all defined symbols on the right-hand sides are checked if they flow into a variable on the lhs of a rule. These variable locations are tagged and form the basis on which we propagate this tag further on.
- The two steps we repeat to obtain all locations that need to be tagged are defined in the sets $L2R_i(\mathfrak{R})$ (Left-to-Right) and $R2L_i(\mathfrak{R})$ (Right-to-Left)
- $L2R_i(\mathfrak{R})$ simply spreads the tagging on the variables of the same symbol that are on the rhs.
- $R2L_i(\mathfrak{R})$ takes some inspiration from the initial set $TAG_0(\mathfrak{R})$, which is trying to do a somewhat similar thing. Except now we strictly consider tagged variable locations on the rhs and the relation between positions that are matched is a bit more relaxed, since we are only considering variables.

When spreading the tag *R2L* we also only tag the locations that are already in $M(\mathfrak{R})$, so as to prevent tagging unnecessary locations.

$$TAG_0(\mathfrak{R}) \;:=\; \{\ (\beta, L, \omega) \mid \alpha = l \rightarrow r \in \mathfrak{R},\ \tau \in pos(r),\ \tau = \pi.\upsilon\,,\ r|_\tau \in D$$
$$\alpha \neq \beta = s \rightarrow t \in \mathfrak{R},\ \exists\, MGU\ for\ \mathfrak{R}|_{(\alpha,R,\pi)}\ and\ \mathfrak{R}|_{(\beta,L,\varepsilon)}\,,$$
$$\upsilon \geq \omega \in pos(s),\ s|_\omega \in V,\ \exists\,(\beta, L, \omega) \in M(\mathfrak{R})\ \}$$

$\forall\, i \in N_{>0}$:

$$TAG_i(\mathfrak{R}) \;:=\; L2R_i(\mathfrak{R}) \cup R2L_i(\mathfrak{R})$$

$$L2R_i(\mathfrak{R}) \;:=\; \{\,(\alpha, R, \tau) \mid \alpha = l \rightarrow r \in \mathfrak{R},\ \pi \in pos(l),\ (\alpha, L, \pi) \in TAG_{i-1}(\mathfrak{R})\,,$$

$$\tau \in pos(r),\ \ l|_{\pi} = r|_{\tau} \in V \} \quad \cup \ \ L2R_{i-1}(\Re)$$

$$R2L_i(\Re) \ := \ \{ \ (\beta, L, \omega) \mid \alpha = l \to r \in \Re,\ \tau \in pos(r),\ \tau = \pi.\upsilon\ ,\ r|_{\tau} \in V,\ \ \exists\, (\alpha, R, \tau) \in TAG_{i-1}(\Re),$$
$$\alpha \neq \beta = s \to t \in \Re,\ \exists\, MGU\ for\ \Re|_{(\alpha,R,\pi)}\ and\ \Re|_{(\beta,L,\varepsilon)}\ ,$$
$$\upsilon \nparallel \omega \in pos(s),\ s|_{\omega} \in V,\ \ \exists\, (\beta, L, \omega) \in M(\Re) \ \}$$
$$\cup \ \ R2L_{i-1}(\Re)$$

In a similar fashion to $M(\Re)$, it is also possible to conclude that $for\ some\ n, m \in N$:

$$T(\Re) \qquad := \bigcup_{i \in N} TAG_i(\Re) \ = \ L2R_n(\Re) \ \cup \ R2L_m(\Re)\ ,\ for\ some\ n, m \in N$$

Combining both algorithms

With the results of both algorithms we can now construct a new set with ideally as few marked locations as possible. Basically removing the locations of variables which are not tagged, i.e. receive no defined symbols and therefore encoding them is pointless.

$$L(\Re) \ := \ F(\Re) \setminus \left\{ (l \to r, R, \tau) \mid r|_{\tau} \in V,\ (l \to r, R, \tau) \notin T(\Re) \right\}$$