1 Consistency

$$\begin{tikzcd} \text{S} \arrow[r, "\mathcal{B}_t"'] \arrow[d, "gen(S)"'] & \text{R-S} \arrow[d, "gen(R-S)"] \\ \text{I} \arrow[r, "\text{CONCUR}"'] & \text{R-I} \end{tikzcd}$$

Rules have the following shape:

$$r : t \to t' \text{ if } \overline{eq}_n$$

where $\overline{eq}_n$ is a sequence of equational conditions that we use to define which transitions are possible and which are not.

Rules are defined by using variables so that with a finite amount of rules we can capture infinite possible scenarios.

Now we proceed to show how concrete instances can be generated starting from the schemas. First some notation.

Definition 1 (Ground Terms)**.** For a given signature $\Sigma = \langle Sorts, Operations \rangle$ the set of ground terms T_Σ is inductively defined as follows:

a. All constants of sorts S in Operations are ground terms of sort S.

b. For every function symbol $f : S_1, \ldots, S_n \to S$ in Operations, if $t_1, \ldots, t_n$ are ground terms of sorts $S_1, \ldots, S_n$, respectively, then $f(t_1, \ldots, t_n)$ is a ground term of sort S where $S_1, \ldots, S_n \in Sorts$

With T^n_Σ we indicate all the ground terms up to the n-nth iteration of the inductive definition. For example, T^0_Σ is the set of constant symbols, T^1_Σ is the set of ground terms given by all the possible applications of the operators to T^0_Σ, and so on.
With $T^n_{\Sigma,S}$ we indicate the ground terms of sort S up to the n-nth iteration.

Algorithm 1 getInstances(Σ, r, E, n)

$\langle Sorts, Operators\rangle \leftarrow \Sigma$
$V \leftarrow vars(r)$ ▷ Get vars used by the rule
$S = \{\{v\} \times T^n_{\Sigma,type(v)} | v \in V\}$
$ConcInst = S_{v_1} \times \ldots \times S_{v_m}$
$ConcRules \leftarrow \emptyset$
for each $i \in ConcInst$ **do**
    $CR \leftarrow instantiate(i, r)$
    $t' \to t'$ if $\overline{eq}_l \leftarrow CR$
    if $E \vdash \overline{eq}_l$ **then**
        $ConcRules \leftarrow ConcRules \cup \{t \to t'\}$
    end if
end for
return $ConcRules \cup getInstances(\Sigma, r, E, n+1)$