Online Appendix: Towards Automated Semantic-Driven Web Service Composition: Case Study on Question Answering Systems

1st Aleksandr Perevalov
Department of Computer Science and Languages
Anhalt University of Applied Sciences
Köthen, Germany
Faculty of Computer Science and Media
Leipzig University of Applied Sciences
Leipzig, Germany
aleksandr.perevalov@{hs-anhalt,htwk-leipzig}.de

2nd Andreas Both
Faculty of Computer Science and Media
Leipzig University of Applied Sciences
Leipzig, Germany
Technology Innovation Unit
DATEV eG
Nuremberg, Germany
andreas.both@{htwk-leipzig,datev}.de

3rd Mike Scherfner
Department of Computer Science and Languages
Anhalt University of Applied Sciences
Köthen, Germany
mike.scherfner@hs-anhalt.de

I. Generalization of CPNs for component-based systems

While modeling a component-based system, several specific aspects should be considered. In particular, a place has the ability to hold numerous annotation types, and therefore, the variables that make up those annotations. A color may contain non-primitive values, i.e., it can be not just a string or an integer, but also a complex data structure (e.g., a tuple). Hence, $\sigma \in \Sigma$ is a set of colors, where σ is an n-tuple $(b_1, b_2, \dots, b_n)$, where b_i is a value of a primitive data type. Therefore, the definition of the marking function M has to be extended to $M' : P \to V$, $p_i \mapsto v_i$. The set V consists of the vectors v_i that hold the corresponding marking colors $\sigma \in \Sigma$ of a place p_i, we define v_i as follows (see Equation 1):

$$v_i = (\sigma_0, \dots, \sigma_{r-2}, \epsilon, \sigma_r, \sigma_{r+1}, \dots, \sigma_{s-2}, \epsilon, \sigma_s, \sigma_{s+1}, \dots, \sigma_{n-1}) \tag{1}$$

The definition before means that v_{i} is a vector with n elements where ϵ could be at any position and means that we have an "empty coloring" and n corresponds to the number of elements in Σ. For example, the marking of a place p_i with $\Sigma = \{\sigma_0, \sigma_1\}$ may look as follows, $M'(p_i) = v_i = (\sigma_0, \epsilon)$ which indicates that the place p_i at a particular point of time stores an annotation of class σ_0 and the rest elements are with empty coloring.

The original color function C has to be also adapted respectively. Therefore, $C' : P \to W$, $p_i \mapsto w_i$, where the set W consists of the vectors w_i that hold the corresponding colors $\sigma \in \Sigma$ and is denoted in the same way as v_i. For example, the color of a place p_i with $\Sigma = \{\sigma_0, \sigma_1, \sigma_2\}$ may look as follows $C'(p_i) = w_i = (\sigma_0, \sigma_1, \epsilon)$, which indicates that the place p_i can store an annotation of types σ_0 and σ_1.

The same adjustment applies also to the initialization function I. Hence, $I' : P \to Y$, $p_i \mapsto y_i$, where the set Y consists of the vectors y_i that hold the corresponding initialization marking colors $\sigma \in \Sigma$ and is denoted in the same way as v_i. For example, the initial marking of a place p_i with $\Sigma = \{\sigma_0, \sigma_1\}$ may look as follows $I'(p_i) = y_i = (\epsilon, \sigma_1)$, which indicates that the place p_i was initialized with the marking that holds a token colored with σ_1.

As the default behavior of a CPN implies that a token is consumed after firing a transition, we may define a transition in a way that they pass all the previously created tokens through. It enables other transitions to access all the tokens, produced up to a current moment in time.