

Anexo A | Anotaciones semánticas del *workflow* EBI InterProScan

El presente anexo extiende la información aportada en el Capítulo 5. Aquí se presentan datos adicionales sobre las anotaciones semánticas del modelo.

A.1 Anotaciones

El modelo del *workflow* que analizamos ha sido anotado con datos semánticos y paramétricos. Por razones de espacio, no es posible incluir toda esta información en este anexo. Sin embargo, se presenta un ejemplo de cada tipo de anotación que podemos encontrar en la red. El siguiente bloque muestra un patrón RDF (RDFG) llamado *job_params_pattern*, asociado al arco de salida de la transición *Job_params*:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <tns:RDFGraphPattern
3   xmlns:tns="http://gidhe.es/edugonza/XMLSchema/RDFGPSchema"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:schemaLocation="http://gidhe.es/edugonza/XMLSchema/RDFGPSchema">
6   <tns:triple>
7     <tns:subject type="variable">package</tns:subject>
8     <tns:predicate type="uri">http://example.org/isA</tns:predicate>
9     <tns:object type="uri">http://example.org/Package</tns:object>
10  </tns:triple>
11  <tns:triple>
12    <tns:subject type="uri">http://example.org/GOTERMS</tns:subject>
13    <tns:predicate type="uri">http://example.org/hasValue</tns:predicate>
14    <tns:object type="variable">goterms</tns:object>
15  </tns:triple>
16  <tns:triple>
17    <tns:subject type="uri">http://example.org/ASYNC</tns:subject>
18    <tns:predicate type="uri">http://example.org/hasValue</tns:predicate>
19    <tns:object type="variable">async</tns:object>
20  </tns:triple>
21  <tns:triple>
22    <tns:subject type="uri">http://example.org/CRC</tns:subject>
23    <tns:predicate type="uri">http://example.org/hasValue</tns:predicate>
24    <tns:object type="variable">crc</tns:object>
25  </tns:triple>
26  <tns:triple>
27    <tns:subject type="uri">http://example.org/SEQTYPE</tns:subject>
```

```

28     <tns:predicate type="uri">http://example.org/hasValue</tns:predicate
29     >
30     <tns:object type="variable">seqtype</tns:object>
31 </tns:triple>
32 <tns:triple>
33     <tns:subject type="uri">http://example.org/EMAIL</tns:subject>
34     <tns:predicate type="uri">http://example.org/hasValue</tns:predicate
35     >
36     <tns:object type="variable">email</tns:object>
37 </tns:triple>
38 </tns:RDFGraphPattern>

```

Este código corresponde a la anotación en formato XML. Por claridad, una representación más compacta del mismo patrón sería:

```

?package t:isA t:Package .
t:GOTERMS t:hasValue ?goterms .
t:ASYNc t:hasValue ?async .
t:CRC t:hasValue ?crc .
t:SEQTYPE t:hasValue ?seqtype .
t:EMAIL t:hasValue ?email

```

El símbolo ? representa que el identificador siguiente se refiere a una variable. Así vemos que, en la primera tripleta, aparece la variable *package*. Esta variable representa un valor que aún no ha sido determinado, pero que lo estará cuando se realice el *binding*. Lo mismo ocurre con el resto de variables *goterms*, *async*, *crc*, *seqtype* y *email*. Así vemos que este patrón captará un grafo RDF que contenga, al menos, seis tripletas que coincidan con el patrón propuesto.

También podemos encontrar grafos RDF (RDFG), que corresponden al marcado en los lugares de la red. Un ejemplo de dichas anotaciones podría ser el correspondiente al marcado en el lugar *goterms_default*, que se presenta en el siguiente bloque:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tns:RDFGraph
3   xmlns:tns="http://gidhe.es/edugonza/XMLSchema/RDFGSchema"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:schemaLocation="http://gidhe.es/edugonza/XMLSchema/RDFGSchema">
6   <tns:triple>
7     <tns:subject type="uri">http://example.org/Package01</tns:subject>
8     <tns:predicate type="uri">http://example.org/isA</tns:predicate>
9     <tns:object type="uri">http://example.org/Package</tns:object>
10  </tns:triple>
11  <tns:triple>
12    <tns:subject type="uri">http://example.org/GOTERMS</tns:subject>
13    <tns:predicate type="uri">http://example.org/hasValue</tns:predicate
14    >
15    <tns:object type="bnode">goterms</tns:object>
16  </tns:triple>
17 </tns:RDFGraph>

```

Esto se puede resumir, de forma compacta en:

```

t:Package01 t:isA t:Package .
t:GOTERMS t:hasValue _:goterms

```

En esta anotación, como se puede observar, se hace uso de *Blank nodes*, que representan parámetros que se tratarán con el *SMT solver*. El tipo de cada parámetro se define en el fichero *parameters.smt2* en formato SMT-Lib v2:

```

1      (declare-fun text () Bool)
2      (declare-fun xml () Bool)
3      (declare-fun goterms () Int)
4      (declare-fun async () Int)
5      (declare-fun crc () Int)
6      (declare-fun seqtype () Int)
7      (declare-fun email () Int)
8      (declare-fun sequence () Int)
9      (declare-fun input_datatype () Int)

```

Otro tipo de anotaciones que podemos encontrar en la red son las guardas paramétricas en las transiciones. El siguiente bloque muestra un ejemplo, correspondiente a la transición *get_XML_Result*:

```

1      (= $xml$ true)
2      (= $status$ true)

```

Esto quiere decir que la transición se disparará si no existe una contradicción entre las declaraciones paramétricas existentes en el estado de partida, y las declaraciones que establecen que el valor de la variable *xml* es *true* y el de *status* es *true* también.

Respecto a las consultas CTL que se realizan sobre el ejemplo, vamos a fijarnos en la correspondiente a la fórmula: $EF(RDFG_{Text_Result} \vee RDFG_{GFF_Result} \vee RDFG_{XML_Result} \vee RDFG_{Status} \vee RDFG_{Job_ID})$. Cuando creamos esta fórmula en el editor *Web*, se genera un fichero XML cuyo contenido es el siguiente:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tns:CTL-Formula
3   xmlns:tns="http://gidhe.es/edugonza/XMLSchema/FormulaSchema"
4   xmlns:tns2="http://gidhe.es/edugonza/XMLSchema/RDFGSchema"
5   xmlns:tns3="http://gidhe.es/edugonza/XMLSchema/RDFGPSchema">
6   <tns:Mono value="ef">
7     <tns:Dual value="or">
8       <tns:leftFormula>
9         <tns:Dual value="or">
10          <tns:leftFormula>
11            <tns:Dual value="or">
12              <tns:leftFormula>
13                <tns:Dual value="or">
14                  <tns:leftFormula>
15                    <tns:Literal>
16                      <tns:rdfg inPlace="netId1337458105565_node_1609">
17                        </tns:rdfg>
18                      </tns:Literal>
19                    </tns:leftFormula>
20                  <tns:rightFormula>
21                    <tns:Literal>
22                      <tns:rdfg inPlace="netId1337458105565_node_1605">
23                        </tns:rdfg>
24                      </tns:Literal>
25                    </tns:rightFormula>
26                  </tns:Dual>
27                </tns:leftFormula>
28              <tns:rightFormula>
29                <tns:Literal>
30                  <tns:rdfg inPlace="netId1337458105565_node_1607">

```

```

31         </tns:rdfg>
32     </tns:Literal>
33 </tns:rightFormula>
34 </tns:Dual>
35 </tns:leftFormula>
36 <tns:rightFormula>
37     <tns:Literal>
38         <tns:rdfg inPlace="netId1337458105565_node_1595">
39             </tns:rdfg>
40         </tns:Literal>
41     </tns:rightFormula>
42 </tns:Dual>
43 </tns:leftFormula>
44 <tns:rightFormula>
45     <tns:Literal>
46         <tns:rdfg inPlace="netId1337458105565_node_1615">
47             </tns:rdfg>
48         </tns:Literal>
49     </tns:rightFormula>
50 </tns:Dual>
51 </tns:Mono>
52 </tns:CTL-Formula>

```