10.3.5 Streamability of Stylesheet Functions

Under specific conditions, described in this section, a stylesheeet function can be used to process nodes from a streamed input document.

A stylesheet function is guaranteed-streamable if all the following conditions apply:

1. The streamable attribute of the xsl:function declaration is present with the value yes.
2. There is at most one parameter whose declared type permits nodes. The declared type permits nodes if the as attribute of the xsl:param element is either absent, or is set to a SequenceType that maps to a U-type that has a non-empty intersection with *U{N}* (see *19.2 Determining the Static Type of a Construct*).
3. At least one of the following conditions is true:
    1. There is no parameter whose declared type permits nodes.

        Note:

        In this case, the body of the function is immaterial.

    2. ~~Both~~ All of the following conditions are true:
        1. The sequence constructor forming the body of the function declaration has a sweep that is either motionless or consuming.
        2. At least one of the following conditions is true:
            1. The sequence constructor forming the body of the function declaration has a posture that is not roaming~~. either grounded or striding~~.
            2. The return type of the function, expressed in the as attribute of the xsl:function element, is present and is set to a SequenceType that maps to a U-type that has an empty intersection with *U{N}* (see *19.2 Determining the Static Type of a Construct*). ~~is an atomic or union type.~~

                Note:

                In this case the result of the function is atomized, and is therefore grounded.

19.8.7.12 Streamability of Function Calls

<snip />

DEFINITION: a **streaming argument** is the (only) argument of a streamable stylesheet function that has a SequenceType that maps to a U-type that has a non-empty intersection with *U{N}* (see *19.2 Determining the Static Type of a Construct*).

DEFINITION: A streamable function is said to be **analyzable** if it is not recursive. For this purpose a function F is recursive if it contains a call on F, or on a potential caller of F; a function is a potential caller of F if it either (a) contains a call on F, or (b) contains a call on a potential caller of F.

Note:

Any dynamic function call could also be treated as a potential caller of *F*; however, functions containing dynamic function calls that could jeopardize streamability will be deemed non-streamable as a consequence of other rules.

For a call to a stylesheet function, the following rules apply, in order:

1. If the referenced function is declared with streamable="yes" (and is analyzed as guaranteed streamable by the rules in 10.3.5 Streamability of Stylesheet Functions):
   1. ~~At most one argument of a streamable function is allowed to accept nodes; call this the streaming argument.~~
   2. ~~If the expression supplied as the streaming argument is grounded, then the general streamability rules apply. There is one operand role for each argument in the function signature, and its operand usage is the type-determined usage based on the declared type of that argument.~~
   3. ~~A streamable function is said to be analyzable if it is not recursive. For this purpose a function *F* is recursive if it contains a call on *F*, or on a potential caller of *F*; a function is a potential caller of *F* if it either (a) contains a call on *F*, or (b) contains a call on a potential caller of *F*.~~

      ~~**Note:**~~

      ~~Any dynamic function call could also be treated as a potential caller of *F*; however, functions containing dynamic function calls that could jeopardize streamability will be deemed non-streamable as a consequence of other rules.~~

   1. If the expression supplied as the streaming argument is roaming or free-ranging, then roaming and free-ranging.
   2. Determine the sweep *S1* of the streaming argument expression, if any:
      1. If the target function is analyzable, then *S1* is the larger sweep of the body of the function and the streaming argument expression.
      2. Otherwise, *S1* is consuming.
   3. Determine the sweep *S2* of the non-streaming arguments by applying the general streamability rules There is one operand role for each non-streaming argument in the function signature, and its operand usage is the type-determined usage based on the declared type of that argument.
   4. Determine *PF* as the posture of the target function body:
      1. If the return type is an atomic or union type, then *PF* is grounded.
      2. If the function non-analyzable, then *PF* is striding.
      3. Otherwise, *PF* is the posture of the sequence constructor making up the body of the target function.
   5. Consider *PA* the posture of the expression making up the streaming argument. Determine the posture *PC* of the function call by applying the first of the following that applies:
      1. If *PA* is grounded, then *PC* is grounded.
      2. If the target function is not analyzable:
         1. If *PA* is striding, then *PC* is *PF*.
         2. Otherwise, *PC* is roaming.
      3. If the target function is motionless:
         1. If *PF* is striding, then *PC* is *PA*.
         2. If *PF* is climbing or grounded, then *PC* is *PF*.
      4. If the streaming argument is striding, then *PC* is *PF*.
      5. Otherwise, *PC* is roaming.
   6. The sweep of the function call is the larger sweep of *S1* and *S2*, or free-ranging if both are consuming, the posture of the function call is *PC*.

   4. ~~If the target function is analyzable then the posture of the function call is the posture of the sequence constructor making up the body of the target function, and the sweep of the function call is the wider of the sweep of the expression supplied as the streaming argument, and the sweep of the sequence constructor making up the body of the target function.~~
   5. ~~If the expression supplied as the streaming argument is climbing or crawling, the function call is roaming and free-ranging.~~
   6. ~~Otherwise, the function call is striding and consuming.~~

2. Otherwise, the general streamability rules apply. There is one operand role for each argument in the function signature, and its operand usage is the type-determined usage based on the declared type of that argument.