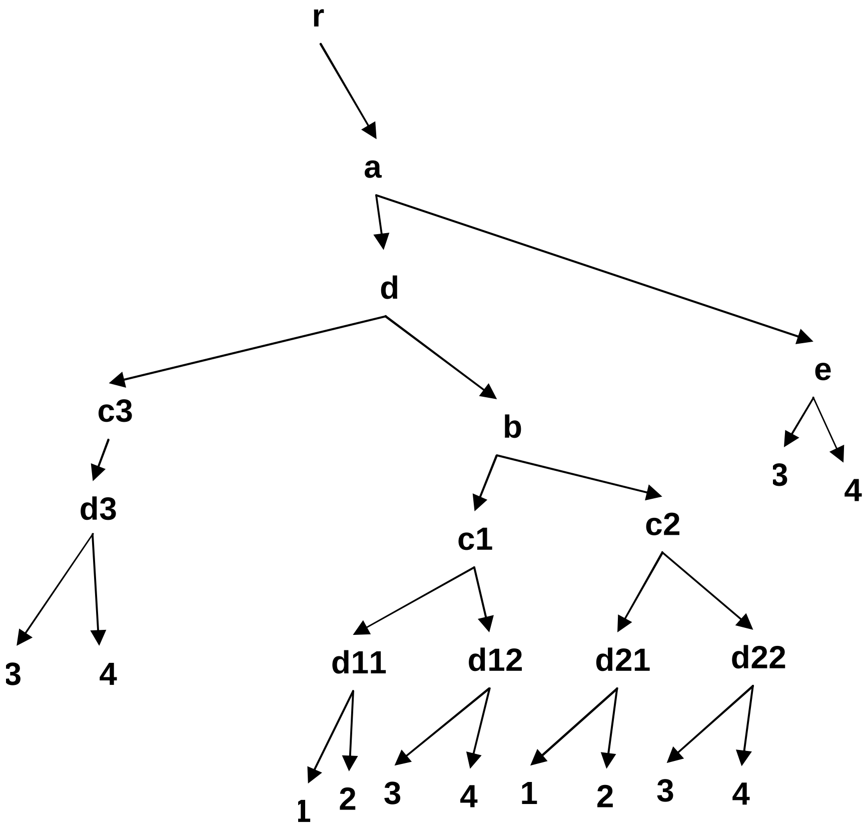
- **Keys and keyrefs may be defined in any element in the schema.**
- **A scoping node of a key (respectively, keyref) is an instance of the element within which the key (or the keyref) is defined.**
- **Consider references from selector-identified nodes of a keyref (i.e., nodes reachable by evaluating the selector expression of the keyref on a scoping node of the keyref) to selector-identified nodes of a key.**
- **These references are local to a scoping node of the keyref, that is:**
    - **Suppose n' is a selector-identified node of a keyref scoping node n.**
    - **Then n' is considered as referencing nodes that are selector-identified nodes of the key, and whose scoping node is a descendant of n.**
- **How many keyed items are referenced by a single instance of a foreign key?**
    - **In a valid document, every selector-identified node of a keyref references exactly one selector-identified node of a key.**
    - **For this, there is a mechanism that resolves conflicts:**
        - **Let n be a scoping node of a keyref KR that refers to a key K. There is a table, associated with n, which holds K's selector-identified nodes that may be referenced by KR's selector-identified nodes whose scoping node is n.**
        - **For each such node the table holds the node's key-sequence (i.e., the values of its fields). In order to construct the table for n, we compute the union of the tables of n's children (see the part highlighted in yellow below).**
        - **Also, if n is a scoping node of K, we add its selector-identified nodes, and key sequences, to the combined table.**
        - **Then, if the combined table contains two or more rows with the same key-sequence ks (and *different* nodes), this is considered a conflict.**
        - **The conflict is resolved as follows: All nodes with key-sequence ks that were added from the children's tables are removed (see the part highlighted in yellow below).**
        - **Consider, for example, the tree in the figure shown below. Suppose that "a" is a scoping node of a keyref. "e" is a selector-identified node of "a" (i.e., "e" is reachable from "a" via the selector expression of the keyref) . c1, c2 and c3 are scoping nodes of the relevant key. d11 and d12 are selector-identified nodes of c1. d21 and d22 are selector-identified nodes of c2. d3 is a selector-identified node of c3. Selector-identified nodes of "a" can reference d3 but they cannot reference d11, d12, d21 or d22. This is because when key information "percolates" bottom**

up, d11 and d21 cancel each other out and similarly d12 and d22 cancel each other out. This means that in a valid document, a selector-identified node of the keyref (whose scoping node is "a") cannot have a key-sequence of (1,2). It can have the key-sequence (3,4), but that would mean that it references d3 and not d12 or d22. In our example, "e" references d3 but does not reference d12 or d22. If we were to change the key-sequence of "e" from) 3,4) to (1,2) then the document would become invalid.



We base our conclusions mainly on the following excerpt from "XML Schema Part1: Structures Second Edition".

3.11.4 Identity-constraint Definition Validation Rules

Validation Rule: Identity-constraint Satisfied
For an element information item to be locally ·valid· with respect to an identity-constraint all of the following must be true:

1 The {selector}, with the element information item as the context node, evaluates to a node-set (as defined in [XPath]). [Definition:] Call this the target node set.

2 Each node in the ·target node set· is either the context node oran element node among its descendants.

3 For each node in the ·target node set· all of the {fields}, with that node as the context node, evaluate to either an empty node-set or a node-set with exactly one member, which must have a simple type. [Definition:] Call the sequence of the type-determined values (as defined in [XML Schemas: Datatypes]) of the [schema normalized value] of the element and/or attribute information items in those node-sets in order the key-sequence of the node.

4 [Definition:] Call the subset of the ·target node set· for which all the {fields} evaluate to a node-set with exactly one member which is an element or attribute node with a simple type the qualified node set. The appropriate case among the following must be true:

4.1 If the {identity-constraint category} is *unique*, then no two members of the ·qualified node set· have ·key-sequences· whose members are pairwise equal, as defined by Equal in [XML Schemas: Datatypes].

4.2 If the {identity-constraint category} is *key*, then all of the following must be true:

4.2.1 The ·target node set· and the ·qualified node set· are equal, that is, every member of the ·target node set· is also a member of the ·qualified node set· and *vice versa*.

4.2.2 No two members of the ·qualified node set· have ·key-sequences· whose members are pairwise equal, as defined by Equal in [XML Schemas: Datatypes].

4.2.3 No element member of the ·key-sequence· of any member of the ·qualified node set· was assessed as ·valid· by reference to an element declaration whose {nillable} is *true*.

4.3 If the {identity-constraint category} is *keyref*, then for each member of the ·qualified node set· (call this the keyref member), there must be a ·node table· associated with the {referenced key} in the [identity-constraint table] of the element information item (see Identity-constraint Table (§3.11.5), which must be understood as logically prior to this clause of this constraint, below) and there must be an entry in that table whose ·key-sequence· is equal to the keyref member's ·key-sequence· member for member, as defined by Equal in [XML Schemas: Datatypes].

Note: The use of [schema normalized value] in the definition of ·key sequence· above means that *default* or *fixed* value constraints may play a part in ·key sequence·s.

Note: Because the validation of *keyref* (see clause 4.3) depends on finding appropriate entries in a element information item's ·node table·, and ·node tables· are assembled strictly recursively from the node tables of descendants, only element information items within the sub-tree rooted at the element information item being ·validated· can be referenced successfully.

Note: Although this specification defines a ·post-schema-validation infoset· contribution which would enable schema-aware processors to implement clause 4.2.3 above (Element Declaration (§3.3.5)), processors are not required to provide it. This clause can be read as if in the absence of this infoset contribution, the value of the relevant {nillable} property must be available.

3.11.5 Identity-constraint Definition Information Set Contributions

Schema Information Set Contribution: Identity-constraint Table
[Definition:] An eligible identity-constraint of an element information item is one such that clause 4.1 or clause 4.2 of Identity-constraint Satisfied (§3.11.4) is satisfied with respect to that item and that constraint, or such that any of the element information item [children] of that item have an [identity-constraint table] property whose value has an entry for that constraint.

[Definition:] A node table is a set of pairs each consisting of a ·key-sequence· and an element node.

Whenever an element information item has one or more ·eligible identity-constraints·, in the ·post-schema-validation infoset· that element information item has a property as follows:

PSVI Contributions for element information items
`[identity-constraint table]`

one Identity-constraint Binding information item for each ·eligible identity-constraint·, with properties as follows:
PSVI Contributions for Identity-constraint Binding information items
`[definition]`
The ·eligible identity-constraint·.
`[node table]`
A ·node table· with one entry for every ·key-sequence· (call it k) and node (call it n) such that one of the following must be true:

1 There is an entry in one of the ·node tables· associated with the [definition] in an Identity-constraint Binding information item in at least one of the [identity-constraint table]s of the element information item [children] of the element information item whose ·key-sequence· is k and whose node is n;

2 n appears with ·key-sequence· k in the ·qualified node set· for the [definition].

provided no two entries have the same ·key-sequence· but distinct nodes. Potential conflicts are resolved by not including any conflicting entries which would have owed their inclusion to clause 1 above. Note that if all the conflicting entries arose under clause 1 above, this means no entry at all will appear for the offending ·key-sequence·.

Note: The complexity of the above arises from the fact that *keyref* identity-constraints may be defined on domains distinct from the embedded domain of the identity-constraint they reference, or the domains may be the same but self-embedding at some depth. In either case the ·node table· for the referenced identity-constraint needs to propagate upwards, with conflict resolution.