Consistency Management with Repair Actions

MF1233055 周昊一

今日的许多大型软件系统都是以分布的方式开发的，而软件工程师在开发软件时也常常在开发过程中使用各种正式或者半正式的标记。因此如何维持系统的一致性就成了一个难题。

本文目的就是在于帮助软件工程师管理文档中的一致性。对一致性的管理主要可以分为三个部分：表示出一致性约束，一致性检查，以及对不一致性的修复。本文主要是在一个叫做xlinkit的工具的基础上，解决对不一致性的修复问题。

Xlinkit是一个用于检查分布式文档的一致性的框架，它包含一个用于表示文档之间约束的以一阶逻辑为基础的语言，以及文档管理机制和文档一致性检查引擎。

本文的工作基于xlinkit的输出内容，目的是在当不一致性被检测到时提供给用户一些可能的选择，包括添加，删除，更改三种操作，来修复不一致性。虽然更改操作可以表示为先删除，再添加，但是由于这是一个非常常用的操作，并且可以简化用户的理解，因此在保留了它作为一个原子操作。

该框架的主要内容可以分为两个部分。一个是repair administrator，用于通过静态分析来生成修复行为。另一个是repair manager，用于将结果呈现给用户，并接受用户的操作。

其中核心的部分是repair administrator，它所生成的修复行为首先要确保：1.除了生成的这些修复行为外没有其他的方法可以去除不一致性；2.生成的这些修复行为每一个都可以去除不一致性。在这个基础上，还要去除一些从用户的角度来说显然是错误的选项。Repair administrator的输入是xlinkit的公式（在文中给出了定义），然后它将这个公式映射为一个修复行为的集合的集合（由于公式中的or，and和imply，所以是多个集合的集合），文中给出了对应不同公式的映射规则。

随后文章结合一个实例讲解了修复行为的具体生成过程，甚至给出了一个该框架的具体实现例子。

对于未来的前景，该框架可以在此基础上进一步扩展为更高级的修复框架：通过一些约束与分析，力求最小化修复行为的集合，甚至消除用户的交互（自动的进行修复）。

当前该框架也存在一些缺陷，一个是没有考虑到文档的语法以及由其他的约束行为生成的修复行为，二是没有考虑一个操作可以修复多处不一致性的情况，三是输出的信息过于复杂。