# 3 概念生成方法

本章我们先介绍RDF知识库的标注方法，在此基础上介绍图的频繁模式挖掘算法，

以及最核心的概念生成模型。

## 3.1 RDF知识库的标注方法

我们知道知识库用一组RDF三元组描述，三元组在上一章中介绍过，是由主语（subject）-谓语（predicate）-宾语（object）构成。通常情况下，一组RDF数据表示的是图集｛G1，G2，…｝，图G的顶点集合用Vg表示，边集合用Eg表示，描述集用Lab表示，因此我们有G =（V, E, L）。当我们对图集采用频繁子图的挖掘研究方法时，我们会得不到任何可用的结果，因为图集描述的是关于实体A与实体B之间的联系，任何联系只会出现一次，这种关联方式显然不可能存在一些频繁的模式。因此我们需要对原始数据图集进行标注，可以发现总共有三种标注方法：

实体-实体 我们对每个实体依次进行标号1，2，。。。n，即每一个实体映射到一个正整数，此整数仅表示顺序，并且当实体为初始结点时，标记为特殊标号0；对于每条边，依次进行标号1，2，。。。n，对应于描述谓语的顺序。分别对每一个图集标号，构成处理后的原始数据集。

实体-属性 同样，这种情况跟实体-实体的标号方式基本一致，只是对于属性的结点时标号和实体标号分开；边的标号形式与实体-实体一致。分别对每一个图集标号，构成处理后的原始数据集。

概念-概念 对每个实体依次进行若干个标号1，2，。。。n，此时每一个实体映射到若干个正整数，此整数表示的是概念集合中的顺序，即整个图转换为概念与概念之间的关系，每个结点有若干个概念；边的标号形式与实体-实体一致。分别对每一个图集标号，构成处理后的原始数据集。

对于前两种情况，我们找到的频繁出现的结构形式是一个实体与若干个实体有联系，或者是一个实体与若干个属性有联系，此时挖掘出来的结果突出的是此类实体联系频繁出现，此类实体相似；而最后一种情况，频繁出现的结构形式是一种概念与若干个概念有联系，结果突出的是概念之间的联系，此时实体已经升华为概念，有效地帮助识别概念常见组合方式。

## 3.2 RDF的频繁模式挖掘算法

频繁子图的严谨定义以及RDF的挖掘

我们先给出子图以及频繁子图的严格定义。

**子图** S =（Vs，Es，Ls）是G =（V，E，L）的子图当且仅当。进一步地，S是G的同构子图当且仅当满足的单射函数：(a) ，(b) 。这样，同构子图不再受到结点标记的限制。

**频繁子图** 假设图S是图G的同构子图，在一个标记的图数据集中，定义图S的支持度support为图S在D中出现的次数，频繁子图是指支持度大于或等于最小支持度阈值min\_support的图。

由此，我们有了问题的准确性描述，在利用上一节提到的标注方法标注的RDF数据图集中，找出其中的频繁子图。下面介绍应用于图的一般化挖掘算法，以及应用于RDF标注图的挖掘算法。

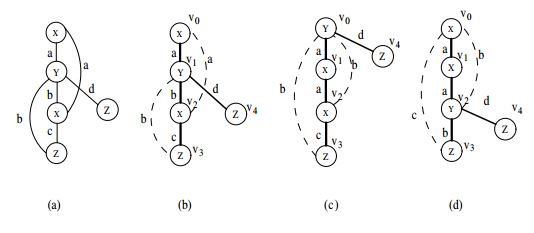
正如上一章相关技术所言，图的频繁子图挖掘一般采用基于Apriori方法和基于模式增长方法，并且由于Apriori方法只能采用图的广度优先遍历算法，受到较大的限制，所以本文主要基于改进后的模式增长算法进行频繁子图的挖掘。传统的模式增长算法有一个普遍的缺点是扩展图的低效率，每一次扩展的新图需要对之前已经找出的频繁子图一一比对来避免重复，这一过程的时间复杂度是。gSpan充分地考虑到了这一点，并将此过程的时间复杂度降为线性时间，从而极大地改进了算法的效率。

整体说来，模式增长的算法思想无非与两点有关，扩展与匹配。gSpan引入了DFSCode和最右扩展思想来保证每一个图只被搜索到一次，并且由于搜索扩展过程中DFSTree的产生使得匹配过程也变得额外简单。

### 3.2.1 扩展

首先，图的遍历方式主要有两种，深度优先遍历方式和广度优先遍历方式，每一个图遍历完成后依据遍历结点的顺序形成一棵DFS树，树的每一个结点表示图的一条边。根据图的DFS遍历顺序的不同，可以形成完全不同的深度优先遍历树。

在构造一棵DFS树的过程中，顶点用表示，i<j表示在之前被搜索到，边用label属性表示。每一次图结点和边的访问，使用下标记录下来，我们可以得到一个DFS下标（DFS subscripting），形如（vi，vj，la，lc，lb），其中vi，vj表示结点在整个图中的访问顺序，la，lb表示vi，vj对应结点的属性，lc表示连接vi，vj的边的属性。所以，一个图用结点访问顺序的方式表示将是一组DFS下标。补充说明图



因此，每一个图对应于不同的DFS树，每一棵树对应一组DFS下标。由此，DFS下标的纵向递进是一个图的搜索扩展过程，DFS下标的横向比较是图的访问顺序问题，简而言之，一个图只能选择用一棵DFS树表示，我们可以人为规定一个顺序来决定用哪一棵树表示，这就是接下来要介绍的最右扩展顺序。

**最右路径** 在一棵DFS树T中，遍历树T时记录下顶点的访问顺序，从v0到vn的最短路径称为最右路径。

**最右扩展** 最右扩展包括两种扩展方式，前向扩展和后向扩展方式。前向扩展指的是在最右路径上的顶点连接到另一个顶点的扩展；后向扩展指的是引入一条新边e添加到最右顶点的两个顶点之间。给定一个顶点v，它的后向扩展应该先于前向扩展，并且如果有多个后向扩展选择，先扩展到最右路径上先出现的顶点。

**DFS词典序** DFS词典序要求在满足最右扩展定义的边序前提下，同时满足序列序。序列序指的是当两条边拥有相同的下标时，优先搜索la小的边，la相等的情况下搜索lc小的边，最后如果前两者都想等时搜索lb小的边。

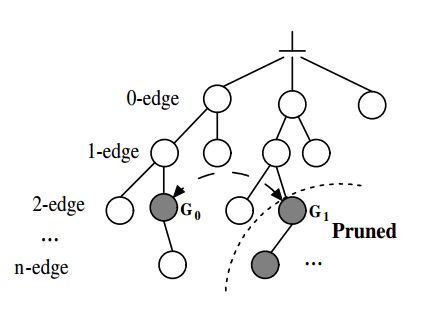
**DFS编码** 图的搜索过程满足DFS最右扩展定义的边序条件的一组下标称为DFS编码，如果同时满足DFS词典序的优先搜索条件，则此DFS编码称为图的最小DFS编码。

**DFS编码树** 由于每条边的候选集可能有多个，因此选择不同的候选边时实际上就是导致了不同的分支，每棵DFS编码树从根结点到叶结点都是一个DFS编码，也构成了一个图。

因此，我们有了图的优化扩展方法，在给定一个图进行频繁子图的挖掘过程中，只进行最小DFS编码的最右扩展，可以保证候选集的产生和图的完全扩展，以及更重要的不会产生重复的复制图，事实上这一点是由后面介绍的后剪枝策略保证的。

### 3.2.2 匹配

在搜索扩展的方式确定以后，我们可以准确无误地得到每条边的候选集。因此在原始图集中，按照DFS词典序，此时对所有图G产生第1条边的候选集，假设G1产生的候选集为，其中Sm表示DFS下标，Gn产生的候选集为，如果中有任意一个下标相同，则匹配成功，将下标记录下来，将所有匹配成功的第一条边的下标作为DFS编码树的一个分支，加入到初始队列中，并且图集变为，由上一个图集D中含有此下标的图G组合而成，判断n是否大于或等于最小支持度，如果满足加入分支，递归进行扩展-匹配-验证支持度-加入分支-验证重复，直到队列为空为止。



在DFS编码树中，在加入分支后的下一个步骤就是验证重复，这是gSpan算法与其它算法的核心区别点之一。模式增长算法有一个显著的问题是当起始边不同时开始扩展之后如何保证扩展完全图不一致，传统的做法是每扩展一次或者每加入一次分支判断扩展后的那个图是否已经曾经出现过，此时又要遍历一遍已经成功匹配的图集。然而一旦将图的每条边和每个顶点进行量化以后，可以直接比对将要加入分支的那条边是否满足DFS词典序，也就是说扩展后的图是否仍然是最小DFS编码。由此一来，验证重复的工作就变成了将扩展边与本图其它边的比较大小，如若不满足，则将该分支剪枝，并且该分支以下的搜索也立即停止。补充说明

### 3.2.3 RDF的同构与匹配

上面介绍的是图的一般频繁模式挖掘算法，但是RDF的频繁模式挖掘与之相比较有点差异。我们知道通过三类标注方式，RDF知识库数据集同样可以转换为图集D，前两类标注方式在匹配过程中会稍有不同，而概念-概念类标注方式则是在扩展部分有差别。

RDF的每一个描述在图中表现为一条边，有时候我们希望模式的定义是所有描述完全相同，包括主语，谓语，宾语，要完全一致才认为同构，我们称之为强同构方式，在实际方法中这一点表现为下标严格一致才匹配成功；但有时我们想把模式定义为一种关系集合，这时候便要求主语、谓语一致，宾语可以不一致，我们称之为弱同构方式，表现为匹配时忽略下标的lc。挖掘过程其他步骤与上节介绍的方法一致，最终得到一系列关系的集合，没有确定的宾语。

另一方面，由于概念-概念类的标注方式使得图集的每个顶点有多个属性，显然不能直接利用之前的方法。这一问题的关键在于把多个属性减小为一个属性，主语层面上根据主语属性一致性将一致的图集再组成一个图集，在这图集基础上进行此主语概念的挖掘；宾语层面上相当于增强了单边扩展的可能性，在扩展候选集时每一个属性都独立地作为一条边，然后进行匹配，匹配成功后原来的边加入到分支中，其它属性都不再拥有。最终的结果是得到概念与概念之间的描述。

## 3.3 概念生成模型

### 3.3.1 概念生成基本原理

首先，我们先定义一下概念以及概念生成的两种不同的方式。对于实体而言，概念指的是实体的上位词，如牵牛花是花，花是牵牛花的上位词，再如莎士比亚是作家，作家是莎士比亚的上位词。简而言之，在实体层面上概念是一种可以广泛描述的类型。因此，概念生成指的是使用知识库数据将该实体的概念导出；而另一方面，对于概念本身而言，此时知识库的数据描述的是关于概念之间的关系集合，通过频繁模式挖掘的方法，可以将频繁出现的概念与概念的关系模式导出，从而形成一个更广泛的概念集合。对应的，RDF的前两类标注方式是为实体层面上做准备，第三类标注方式是为概念关系层面上做预处理。

补充图说明概念生成方法

从上节我们可以得知一种挖掘图集的完全频繁子图的方法，无论是导入一个原始图集，或是给定一个标注好的RDF图集，都可以将所有频繁子图导出，并且将额外携带一些有用信息，比如支持度等。

### 3.3.2 实体集合的概念生成

在一个集合当中，频繁出现的结构是有趣的，它揭示了一种关联规则。关于兴趣度的度量方式主要有两类：支持度（support）和可信度（confidence），它们分别代表发现的规则的有用性和确定性。关联规则支持度大于最小支持度阈值表示该关联规则是足够有趣的，而可信度大于最小可信度阈值则表示该关联规则是足够真实的。

假设A是图G的频繁子图，具有支持度s，B在A的基础上扩展了一条新边，同时B也是图G的频繁子图，规则在图集D中存在，并且有支持度s’，其中s是D中包含出现的图的个数，因此，我们有

由此，当大于最小可信度阈值时，该规则是成立的，该公式可以作为扩展新边的依据。

对于实体的概念生成，除了已经有的部分上位词描述，还可以根据数据图集扩充一部分同类型的概念，这一过程需要用到上节介绍的频繁子图挖掘，在一个DFS编码树的分支中如果出现了概念这一条边，则表明该分支是有趣的，并且此分支可以转换为概念树的一个分支。

补充概念树模型

**概念树** 概念树由DFS编码树构造而成，它的每一支分支的叶节点都必须是描述概念的边，并且树的每一个结点都必须包括三要素：描述边信息、图集标注集合、该分支的可信度。其中描述边信息是DFS下标标注，和DFS编码树一致；图集标注集合指的是该边的支持度上具体的支持图标注；而分支的可信度是该分支的，在DFS编码树构造过程中计算得到。

具体的构造概念树的算法如下：在频繁子图的挖掘过程中有一个步骤是加入分支，此时构造概念树的结点的所有必需要素都具备，描述边信息很容易可以得到，图集标注集合也轻松地获得，而由于放宽了扩展新边的要求，该分支的可信度实际上是由概念树根结点来到此分支所经过分支的最小可信度。在频繁子图挖掘完成时，概念树也随之构造完毕，最后需要重新遍历一遍概念树，将叶节点非概念描述的完整分支剪枝（若从根节点到该叶节点存在其它分支，则将离叶节点最远的无其他分支的分支剪枝）。

因此，整个概念生成的问题就转换为分析概念树的简单问题了。根据概念树的树形结构，我们知道从根节点到每一个叶节点的路径都蕴含着一个概念，其中的含义是每一个分支都与上一个分支有紧密联系，且可信度大于最小可信度阈值证明该路径是真实有效的，而由这条路径构成的整个图是一个有趣的规则。我们认为叶节点的概念适用于根节点代表的所有图标注集合，作为其概念生成的依据，并且分支路径越长说明概念越匹配，因此作为根节点的图标注集合的其中一个标注S的最终可信度Q可以表示为：

其中，TS表示从根节点标注集合中含有S的出发能到达的所有叶节点。

最后，对每一个可能的标注进行概念树的挖掘，得到的结果是标注S到概念集合的映射，其中概念集合的每一个元素包括具体概念描述及其可信度。由此实体的概念生成过程也全部完成。

### 3.3.3关系集合的概念生成

进一步地，对于关系集合的概念生成，其实是在实体集合概念生成的工作基础上完成的。在上一节中，每一个实体都映射到一组概念中，然后利用RDF的第三类概念-概念标注方式，将实体之间的关系或者实体与属性之间的关系转换为概念之间的关系，其中映射为空的实体被消除。至此，原始数据图集的预处理就完成了。

然后，对转换好的数据图集进行频繁子图的挖掘，按照RDF特殊的同构和匹配方法，构造出DFS编码树，此时无需在构造概念树，因为关系集合的概念生成是一种模式发现，关系集合的整体构成了此概念的全部。因此，概念生成的问题也转换为分析DFS编码树的问题，同样，对于每一个根节点的标注，一直到叶节点的路径作为关系集合的一个概念，将DFS编码树遍历完成，得到的结果是标注S到概念集合的映射，其中概念集合的每一个元素是一条路径，代表的是一类频繁子图。由此关系集合的概念生成过程也全部完成。

## 3.4 本章小结

在本章中，我们对整个概念生成方法做了一番详细的讨论。首先，频繁子图的挖掘算法针对于所有预处理完毕的图集，通过应用最右扩展以及DFS下标方法，并且引入DFS编码和DFS词典序，构造出一棵DFS编码树，完美地解决了频繁子图挖掘的扩展与匹配问题，并且将复制图重复搜索复杂度时间降为线性时间。然后，对于RDF的频繁子图挖掘，对应于实体集合的概念生成和关系集合的概念生成，分别用不同的RDF标注方式将RDF知识库数据集转换为可模式挖掘的原始图集。最后，按照频繁子图挖掘算法，以及可信度计算，构造出一棵DFS编码树或者概念树，将概念生成的问题转换为分析概念树的简单问题。