

(12) 发明专利申请

(10) 申请公布号 CN 103198114 A

(43) 申请公布日 2013. 07. 10

(21) 申请号 201310108699. 9

(22) 申请日 2013. 03. 29

(71) 申请人 电子科技大学

地址 611731 四川省成都市高新区(西区)西  
源大道 2006 号

(72) 发明人 唐雪飞 陈科

(74) 专利代理机构 成都宏顺专利代理事务所

(普通合伙) 51227

代理人 周永宏

(51) Int. Cl.

G06F 17/30 (2006. 01)

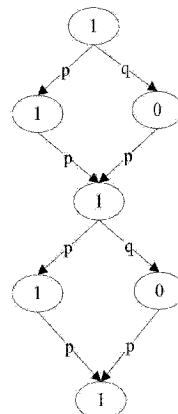
权利要求书2页 说明书8页 附图2页

(54) 发明名称

一种基于重叠有向图的 WEB 服务匹配方法

(57) 摘要

本发明提供了一种基于重叠有向图的 WEB 服务匹配方法,涉及计算机信息分析与数据处理领域,通过将大量的 WEB 服务结点的描述文件(WSDL)载入服务器端内存,形成海量 WEB 描述源数据,为了匹配到指定的服务,建立了空位种子匹配模型,并在此基础上构建重叠有向图,通过连通性和重叠数两个关键指标对种子进行评价,找到最优种子,最终完成 WEB 服务的匹配,是一种快速寻找合适的 WEB 服务方法。本发明可广泛应用于 WEB 服务结点大量存在,对 WEB 服务寻找要求较高的环境中,如面向服务的架构(SOA)、云计算等环境中。



1. 一种基于重叠有向图的 WEB 服务匹配方法,其特征在于:包括以下步骤:

步骤 1:将 WEB 服务的描述文件即 WSDL 文件通过网络载入服务器端的内存,形成 WEB 描述源数据;

步骤 2:建立空位种子匹配模型,将 WEB 服务描述源数据与目标服务数据以空位种子的形式进行描述,得到空位种子匹配框架;

步骤 3:构建面向不同空位种子的重叠有向图,针对每一个重叠有向图计算其连通性和重叠数;

步骤 4:以最大的连通性和最小的重叠数为标准寻找最优空位种子,并通过重叠有向图权重 ODW 描述最优空位种子;

步骤 5:将最优空位种子与目标 WEB 服务描述源数据进行匹配,得到最优的 WEA 服务匹配结果,找到需要的 WEB 服务结点。

2. 根据权利要求 1 所述的一种基于重叠有向图的 WEB 服务匹配方法,其特征在于:步骤 1 的具体方法为:采用超文本传输协议连接所有需要的远程 WSDL 文件地址,将 WEB 服务描述文件通过文档对象树模型进行解析,其解析格式为超文件标记格式,针对解析的文档对象树,将所有包含 `<wsdl:documentation>` 的结点载入服务器端的内存。

3. 根据权利要求 1 或 2 所述的一种基于重叠有向图的 WEB 服务匹配方法,其特征在于:步骤 2 中所述的空位种子,其定义为:

一个空位种子 S 是定义在字符集  $A = \{1, *\}$  上的,并规定以 1 开始,1 结束的固定模式串;其中 1 表示匹配,\* 是一个通配符,表示在该位置可以是 1 匹配或 0 失配;S 表示为:

$$S = (1)^{i_1} (*)^{k_1} (1)^{i_2} (*)^{k_2} \dots (1)^{i_{n-1}} (*)^{k_{n-1}} (1)^{i_n}$$

其中  $(1)^j$  和  $(*)^k$  表示连续 j 个 1 和连续 k 个 \*;

空位种子的长度 (length) 是 S 中所有 1 和 \* 的个数,表示为  $|S|$ :

$$\text{length}(S) = |S| = \sum_{j=1}^n i_j + \sum_{j=1}^{n-1} k_j$$

空位种子的权值 w 定义为所有 1 的个数:

$$\text{weight}(S) = w = \sum_{j=1}^n i_j$$

空位种子的模式是将 S 中所有 \* 替换为 1 或 0 形成的表示串,总共有  $2^{|S|-w}$  个不同的模式。

4. 根据权利要求 1 或 2 所述一种基于重叠有向图的 WEB 服务匹配方法,其特征在于:步骤 3 包括以下步骤:

a) 建立结构有向图:自上而下的节点是按照种子对应元素从左到右的 1 或 0,其中 1 代表节点 1,\* 代表两个节点:1 和 0,每条有向边标有相似度 p 或 q=1-p;

b) 标记步骤:从顶部节点到底部按顺序从序号 1 开始标记每个节点,放到节点值后面用括号括起来,然后删除标记 p 或 q,同时,将首节点序号标记在每条有向边上;

c) 重叠步骤:重叠步骤是一系列重复迭代的过程,依次以每层节点值为 1 的节点作为开始节点构造结构有向图,下一步从第二层节点中值 1 节点开始,以此类推,最终停留在 L-1 层,经过重叠步骤后,得到空位种子的重叠有向图模型。

5. 根据权利要求 4 所述的一种基于重叠有向图的 WEB 服务匹配方法,其特征在于:重

叠有向图的连通性定义为：

有两个有公共结点的边  $e_1$  和  $e_2$ , 即  $e_1$  的结束节点是  $e_2$  的开始节点, 设  $B_1$  是  $e_1$  标记的集合,  $B_2$  是  $e_2$  标记的集合, 称  $B_1$  和  $B_2$  具有连通性或是连通的当且仅当  $B_1 \cap B_2 \neq \emptyset$ ;

重叠有向图的重叠数的定义为：

设  $B$  是边  $e$  的标记集合, 定义重叠数 :

$$oc(e) = |B|。$$

6. 根据权利要求 1 所述的一种基于重叠有向图的 WEB 服务匹配方法, 其特征在于 : 计算重叠有向图权重 ODW 包括以下步骤 :

a) 构建重叠惩罚函数 :

$$\phi(c, p) = \begin{cases} p^c & (e \text{ 的结束节点为 } 1) \\ (1-p)^c & (e \text{ 的结束节点为 } 0) \end{cases}$$

其中  $c=oc(e)$ , 可见当  $p<1$  时,  $\phi(c, p)$  是  $c$  的减函数 ;

b) 构建奖励边连通性的“奖励函数”:

$$\varpi_e(\varphi, p) = \begin{cases} \varphi + p & (e \text{ 与 } e' \text{ 连通且 } e' \text{ 的结束结点为 } 1) \\ \varphi + (1-p) & (e \text{ 与 } e' \text{ 连通且 } e' \text{ 的结束结点为 } 0) \\ \varphi & (e \text{ 与 } e' \text{ 不连通}) \end{cases}$$

其中  $\varphi$  是重叠惩罚函数  $\phi(c, p)$  的值, 对于结点  $N$ , 定义  $w(N) = w(e) * w(e')$ , 用于计算结点的权值 ;

c) 构建重叠有向图权重 :

$$ODW(G) = \sum_{j=1}^{\ell} w(N_j)$$

根据此模型, 对于相同长度和权重的空位种子, 更优质的种子有更高的重叠有向图权重, 在计算候选种子的  $ODW(G)$  后就可以找到优质种子。

7. 根据权利要求 1 或 6 所述的一种基于重叠有向图的 WEB 服务匹配方法, 其特征在于 步骤 5 包括以下步骤 :

a) 首先确定一个终止值  $d$ 、连续种子长度  $w$  和一个阈值  $t$ ,  $d$  值通常是基于统计学的原理指明一个预期的终止  $E$  值, 然后在考虑搜索背景性质的基础上计算出合适的  $d$  值 ;

b) 根据空位种子模式对内存中的 WEB 服务描述源数据进行局部匹配, 当有一个匹配串的分值高于  $t$  时, 则找到了一个选中的字串, 即增强点 ;

c) 当一个分值高于  $t$  的字串选中后, 进行基于动态规划算法的局部延伸寻优, 规定比对的最低分值是  $d$ , 当比对延伸时会遇到一些负的分值, 使得比对的分值下降, 当下降的分值小于  $d$  时, 命中的延伸就会终止, 这时这段比对中具有最高分值的片段便成为一段命中的匹配, 从而找到符合要求的 WEB 服务结点。

## 一种基于重叠有向图的 WEB 服务匹配方法

### 技术领域

[0001] 本发明属于计算机信息分析与数据处理领域,具体涉及一种基于重叠有向图的 WEB 服务匹配方法。

### 背景技术

[0002] Web 服务(WebService)是基于 XML 和 HTTP 的一种服务,每个 web 服务对应了一个 WSDL (WebServicesDescriptionLanguage) 描述文件。WSDL 描述文件是一个用来描述 Web 服务和说明客户端如何与 Web 服务通信的 XML 语言文件。

[0003] 每一个 WSDL 描述文件都封装了若干可供调用接口及其说明文档,但当 WSDL 描述文件大量存在时,如何迅速找到需要的服务接口,是需要解决的一个问题。

#### [0004] 1. WEB 服务描述

[0005] 面向服务体系架构(Service-OrientedArchitecture, SOA)带来了一种新的集成思想,根据它可以构造出灵活的以服务为中心的架构。SOA 凭借其松耦合的特性,可以把企业现有的应用作为服务来发布,服务拥有独立于硬件平台、操作系统和编程语言的接口。因此,不同系统之间可以通过这种定义良好的服务的交互实现系统之间的通信,不仅可以实现海量数据集成与异构业务系统的协同工作,还可以按照模块化的方式来添加新服务或更新现有服务,以解决新的业务需求。

[0006] SOA 的核心就是 WEB 服务,它向外界暴露出一个能够通过 Web 进行调用的应用程序接口(ApplicationProgramInterface, API),能够用编程的方法通过 Web 来调用这个应用程序。Web 服务平台是一套标准,它定义了应用程序如何在 Web 上实现互操作性。你可以用任何你喜欢的语言,在任何你喜欢的平台上写 Web 服务,只要我们可以通过 Web 服务标准对这些服务进行查询和访问。

[0007] Web 服务平台需要一套协议来实现分布式应用程序的创建。任何平台都有它的数据表示方法和类型系统。要实现互操作性,Web 服务平台必须提供一套标准的类型系统,用于沟通不同平台、编程语言和组件模型中的不同类型系统。在传统的分布式系统中,基于界面(interface)的平台提供了一些方法来描述界面、方法和参数(如 COM 和 COBAR 中的 IDL 语言)。同样的,Web 服务平台也必须提供一种标准来描述 Web 服务,让客户可以得到足够的信息来调用这个 Web 服务。

[0008] 如何把来自客户端的关键词语与网络中服务端大量存在的 WSDL 描述文件进行匹配并找到这些关键词语最匹配的 WSDL 描述文件从而实现合适的 web 服务是网络信息搜索技术领域亟待解决的问题。

#### [0009] 2. WEB 服务匹配

[0010] 在网络上的 WEB 服务的数量可能会非常大,如何根据关键字或基本信息寻找到匹配的 WEB 服务,是一个十分重要的问题。将 WEB 服务的 WSDL 描述文件作为数据源,将需要匹配的关键字和基本信息作为基本要素,寻找合适的 WEB 服务的过程,即 WEB 服务的匹配。

[0011] 为了在海量数据源的情况下迅速找到合适的 WEB 服务(即 WSDL 描述文件),本发明

引入基于空位种子的信息局部搜索方法,完成 WEB 服务的匹配。本方法主要基于现有技术“序列比对方法”对匹配序列进行评估。

[0012] 序列比对的定义是将两条序列字符进行两两匹配。在匹配过程中,可能出现以下三种情况:(1)一个字符代替了另一个字符,即发生了突变而产生字符失配;(2)插入一个或多个字符;(3)删除一个或多个字符。因此,在序列比对过程中,并不是简单的字符一一对应,除了匹配和失配以外,还会在序列中引入空位(用符号“-”表示)来反映第 2、3 种变化。

[0013] 如两个序列 AATCTATA 和 AAGATA,下面给出了其中 3 种序列比对情况:

[0014] AATCTATA      AATCTATA      AATCTATA

[0015] AAG-AT-A      AA-G-ATA      AA--GATA

[0016] 为了对序列比对结果进行量化,引入分数机制对序列比对进行打分,以得到最优的序列比对。引入空位后,根据不同的比对情况,可以得到序列比对得分。打分规则其定义如下:

[0017]

$$Score(S_1, S_2) = \sum_{i=1}^n \begin{cases} +k_1, & S_1[i] = S_2[i] \\ -k_2, & S_1[i] \neq S_2[i] \\ -k_3, & S_1[i] = - \text{ 或 } S_2[i] = - \end{cases}$$

[0018] Score(S<sub>1</sub>, S<sub>2</sub>) 表示两个序列 S<sub>1</sub>,S<sub>2</sub> 的匹配得分,i 为自然数用于表示序列中的某个位置,n 代表较长序列的最大字符数,所述字符可以为中文字符也可以为英文字符。

[0019] 其中 k<sub>i</sub> ≥ 0 (1 ≤ i ≤ 3) 分别代表了匹配、失配和空位的情况。从打分公式可以看出,匹配将得到一个正的分数,而失配或空位将得到 0 分或负分。由于可能的序列比对情况千变万化,不同的比对可能得到相同的分数。

[0020] WEB 服务技术广泛地应用在分布式计算、WebService、SOA 等现代信息技术中,在 WEB 服务海量存在的环境中,如何根据关键字快速找到匹配的 WEB 服务,是现代 WEB 框架发展的重要手段,但传统的关键字查找的方法效率低下,无法适应海量 WEB 信息查询的情况。

## 发明内容

[0021] 为了克服传统 WEB 服务匹配方法的弊端,本发明设计了一种基于重叠有向图的 WEB 服务匹配方法,采用基于空位种子的匹配技术和重叠有向图模型,对 WEB 服务进行匹配,大大提高 WEB 服务匹配速率和处理效率。

[0022] 本发明的技术方案为:一种基于重叠有向图的 WEB 服务匹配方法,包括以下步骤:

[0023] 步骤 1:将 WEB 服务的描述文件即 WSDL 文件通过网络载入服务器端的内存,形成 WEB 描述源数据;

[0024] 步骤 2:建立空位种子匹配模型,将 WEB 服务描述源数据与目标服务数据以空位种子的形式进行描述,得到空位种子匹配框架;

[0025] 步骤 3:构建面向不同空位种子的重叠有向图,针对每一个重叠有向图计算其连通性和重叠数;

[0026] 步骤 4:以最大的连通性和最小的重叠数为标准寻找最优空位种子,并通过重叠有向图权重 ODW 描述最优空位种子;

[0027] 步骤 5 : 将最优空位种子与目标 WEB 服务描述源数据进行匹配, 得到最优的 WEA 服务匹配结果, 找到需要的 WEB 服务结点。

[0028] 作为优选: 步骤 1 的具体方法为: 采用超文本传输协议连接所有需要的远程 WSDL 文件地址, 将 WEB 服务描述文件通过文档对象树模型进行解析, 其解析格式为超文件标记格式, 针对解析的文档对象树, 将所有包含 `<wsdl:documentation>` 的结点载入服务器端的内存。

[0029] 作为优选: 步骤 2 中所述的空位种子, 其定义为:

[0030] 一个空位种子 S 是定义在字符集  $A = \{1, *\}$  上的, 并规定以 1 开始, 1 结束的固定模式串; 其中 1 表示匹配, \* 是一个通配符, 表示在该位置可以是 1 匹配或 0 失配; S 表示为:

$$[0031] S = (1)^{i_1} (*)^{k_1} (1)^{i_2} (*)^{k_2} \dots (1)^{i_{n-1}} (*)^{k_{n-1}} (1)^{i_n}$$

[0032] 其中  $(1)^j$  和  $(*)^k$  表示连续 j 个 1 和连续 k 个 \*;

[0033] 空位种子的长度 (length) 是 S 中所有 1 和 \* 的个数, 表示为  $|S|$ :

$$[0034] \text{length}(S) = |S| = \sum_{j=1}^n i_j + \sum_{j=1}^{n-1} k_j$$

[0035] 空位种子的权值 w 定义为所有 1 的个数:

$$[0036] \text{weight}(S) = w = \sum_{j=1}^n i_j$$

[0037] 空位种子的模式是将 S 中所有 \* 替换为 1 或 0 形成的表示串, 总共有  $2^{|S|-w}$  个不同的模式。

[0038] 作为优选: 步骤 3 包括以下步骤:

[0039] a) 建立结构有向图: 自上而下的节点是按照种子对应元素从左到右的 1 或 0, 其中 1 代表节点 1, \* 代表两个节点: 1 和 0, 每条有向边标有相似度 p 或  $q=1-p$ ;

[0040] b) 标记步骤: 从顶部节点到底部按顺序从序号 1 开始标记每个节点, 放到节点值后面用括号括起来, 然后删除标记 p 或 q, 同时, 将首节点序号标记在每条有向边上;

[0041] c) 重叠步骤: 重叠步骤是一系列重复迭代的过程, 依次以每层节点值为 1 的节点作为开始节点构造结构有向图, 下一步从第二层节点中值 1 节点开始, 以此类推, 最终停留在 L-1 层, 经过重叠步骤后, 得到空位种子的重叠有向图模型。

[0042] 作为优选: 重叠有向图的连通性定义为:

[0043] 有两个有公共结点的边  $e_1$  和  $e_2$ , 即  $e_1$  的结束节点是  $e_2$  的开始节点, 设  $B_1$  是  $e_1$  标记的集合,  $B_2$  是  $e_2$  标记的集合, 称  $B_1$  和  $B_2$  具有连通性或是连通的当且仅当  $B_1 \cap B_2 \neq \emptyset$ ;

[0044] 重叠有向图的重叠数的定义为:

[0045] 设 B 是边 e 的标记集合, 定义重叠数:

$$[0046] \text{oc}(e) = |B|.$$

[0047] 作为优选: 计算重叠有向图权重 ODW 包括以下步骤:

[0048] a) 构建重叠惩罚函数:

[0049]

$$\varphi(c, p) = \begin{cases} p^c & (e \text{ 的结束节点为 } 1) \\ (1-p)^c & (e \text{ 的结束节点为 } 0) \end{cases}$$

[0050] 其中  $c=oc(e)$ , 可见当  $p<1$  时,  $\varphi(c,p)$  是  $c$  的减函数;

[0051] b) 构建奖励边连通性的“奖励函数”:

[0052]

$$\varpi_{e'}(\varphi, p) = \begin{cases} \varphi + p & (e \text{ 与 } e' \text{ 连通且 } e' \text{ 的结束结点为 } 1) \\ \varphi + (1-p) & (e \text{ 与 } e' \text{ 连通且 } e' \text{ 的结束结点为 } 0) \\ \varphi & (e \text{ 与 } e' \text{ 不连通}) \end{cases}$$

[0053] 其中  $\varphi$  是重叠惩罚函数  $\varphi(c,p)$  的值, 对于结点 N, 定义  $w(N)=w(e)*w(e')$ , 用于计算结点的权值;

[0054] c) 构建重叠有向图权重:

$$[0055] ODW(G) = \sum_{j=1}^{\varepsilon} w(N_j)$$

[0056] 根据此模型, 对于相同长度和权重的空位种子, 更优质的种子有更高的重叠有向图权重, 在计算候选种子的  $ODW(G)$  后就可以找到优质种子。

[0057] 作为优选: 步骤 5 包括以下步骤:

[0058] a) 首先确定一个终止值  $d$ 、连续种子长度  $w$  和一个阈值  $t$ ,  $d$  值通常是基于统计学的原理指明一个预期的终止  $E$  值, 然后在考虑搜索背景性质的基础上计算出合适的  $d$  值;

[0059] b) 根据空位种子模式对内存中的 WEB 服务描述源数据进行局部匹配, 当有一个匹配串的分值高于  $t$  时, 则找到了一个选中的字串, 即增强点;

[0060] c) 当一个分值高于  $t$  的字串选中后, 进行基于动态规划算法的局部延伸寻优, 规定比对的最低分值是  $d$ , 当比对延伸时会遇到一些负的分值, 使得比对的分值下降, 当下降的分值小于  $d$  时, 命中的延伸就会终止, 这时这段比对中具有最高分值的片段便成为一段命中的匹配, 从而找到符合要求的 WEB 服务结点。

[0061] 本发明的有益效果为: 本发明设计了一种在大量的 WSDL 描述文件中匹配目标的方法, 提出了空位种子模型, 并通过重叠有向图寻找最优空位种子, 完成对目标 WEB 服务的匹配。当需要在海量 WSDL 描述文件中寻找目标时, 克服了传统的依次逐一匹配方式存在的速度慢的缺点, 本发明通过设计性能优良的空位种子, 利用种子模型去匹配 WSDL 描述文件, 从而大大提高匹配效率, 进而有效地提高 Web 服务的性能, 这对日益发展的互联网服务扩展有着重要的意义。

## 附图说明

[0062] 图 1 为空位种子  $1*1*1$  的结构有向图;

[0063] 图 2 为空位种子  $11**1$  的结构有向图;

[0064] 图 3 为空位种子  $1*1*1$  结构有向图的首次修改;

[0065] 图 4 为空位种子  $11**1$  结构有向图的首次修改;

[0066] 图 5 为空位种子  $1*1*1$  的重叠有向图;

[0067] 图 6 为空位种子  $11**1$  的重叠有向图。

## 具体实施方式

[0068] 为了便于本领域技术人员的理解, 下面结合附图和具体的实施例对本发明做进一

步的说明。

[0069] 本发明将按照 WSDL 描述文件载入、空位种子的建立、重叠有向图的生成、连通性和重叠数的计算、基于最优空位种子的匹配五个步骤进行详细描述：

[0070] 步骤 1：服务器端从网络中的 WEB 服务器载入大量的 WSDL 描述文件在服务器端形成海量 WSDL 描述文件源数据，此步骤为现有技术。

[0071] 步骤 1 的具体方法为：采用超文本传输协议连接所有需要的远程 WSDL 文件地址，将 WEB 服务描述文件通过文档对象树模型进行解析，其解析格式为超文件标记格式，针对解析的文档对象树，将所有包含 <wsdl:documentation> 的结点载入服务器端的内存。

[0072] 步骤 2：空位种子的建立

[0073] 1) 空位种子按照以下公式进行定义：

[0074] 一个空位种子 S 是定义在字符集  $A = \{1, *\}$  上的，并规定以 1 开始，1 结束的固定模式串。其中 1 表示匹配，\* 是一个通配符，表示在该位置可以是 1( 匹配 ) 或 0( 失配 )。S 可以表示为：

$$[0075] S = (1)^{i_1} (*)^{k_1} (1)^{i_2} (*)^{k_2} \cdots (1)^{i_{n-1}} (*)^{k_{n-1}} (1)^{i_n}$$

[0076] 举个实施例：如  $1**1*1$  就是一个空位种子。

[0077] 其中  $(1)^i$  和  $(*)^k$  表示连续  $i$  个 1 和连续  $k$  个 \*， $i_j$  表示  $i$  的不同取值， $j \in [1, n]$ ，其中  $n$  为自然数； $k_j$  表示  $k$  的不同取值， $j \in [1, n-1]$  其中  $n$  为自然数。可以看出，空位种子 S 由若干个连续多个 1 组成的“1 块 (1blocks)”和连续多个 \* 组成的“\* 块 (\*blocks)”构成。由于规定 S 必须以 1 开头和 1 结尾，所以“1 块”的个数等于“\* 块”的个数加 1。

[0078] 空位种子 S 的长度 (length) 是 S 中所有 1 和 \* 的个数，表示为  $|S|$ ：

$$[0079] \text{length}(S) = |S| = \sum_{j=1}^n i_j + \sum_{j=1}^{n-1} k_j$$

[0080] 空位种子的权值 w 定义为所有 1 的个数：

$$[0081] \text{weight}(S) = w = \sum_{j=1}^n i_j$$

[0082] 空位种子的模式是将 S 中所有 \* 替换为 1 或 0 形成的表示串。显然，总共有  $2^{|S|-w}$  个不同的模式。

[0083] 步骤 3：重叠有向图的生成：

[0084] 为了快速找到最优或接近最优的空位种子，我们引入了种子重叠的概念。种子的命中可以重叠，但重叠的命中将检测出同一处的匹配。因此，种子灵敏度反比于重叠命中数，即良好的种子应该有低的重叠命中数。即使不同的空位种子具有相同的长度和权重，其仍有不同的灵敏度和重叠性。

[0085] 1) 构造空位种子结构有向图：

[0086] 由种子结构产生的自上而下的有向图。自上而下的节点是按照种子对应元素从左到右的 1 或 0，其中 1 代表节点 1，\* 代表两个节点：1 和 0。每条有向边标有相似度 p 或  $q=1-p$ ，读作“以概率 p ( 或 q ) 出现”。

[0087] 为了解释不同的种子具有相同的长度和权重的不同结构，我们不采用最简单的空位种子  $1*1$  作为我们的第一个例子，因为其只有一个候选种子，即  $1*1$ ，其长度为 3 和权重为 2。我们以长度为 5 权重为 3 的空位种子开始。这里有两个候选种子（除去对称的）： $1*1*1$

和 11\*\*1。

[0088] 图 1 和图 2 显示了种子 1\*1\*1 和 11\*\*1 对应的两个不同种子结构有向图。

[0089] 根据空位种子的定义,每一个结构有向图都由节点 1 开始和结束。对于一个长度 L 和重量 w 的空位种子,从顶级节点到底部有 L 层,每一层都至少有一个节点,如节点 1。下层节点以边 p(下层节点为 1) 或边 q(下层节点为 0) 与上层节点相连。显然,在一个基本结构有向图中有  $w+2(L-w)$  个节点。

[0090] 空位种子的结构有向图是一个用来检测相同长度和权重种子之间的结构差异的基本模型。节点中的值为 1 或 0,称为节点值。空位种子的重叠有向图,是对结构有向图的扩展。为了使有向图保存更多的种子结构信息,我们需要对它进行修改,包括标记步骤和重叠步骤。

[0091] 2) 标记步骤

[0092] 首先,我们从顶部节点到底部按顺序从序号 1 开始标记每个节点,放到节点值后面用括号括起来,然后删除标记 p 或 q,因为我们可以从下层节点(1 对应 p, 0 对应 q)分析得到;同时,将首节点序号标记在每条有向边上。由于这是建立重叠有向图的第一步,因此所有的边都以起始节点(1) 进行标记,从第二步开始,起始节点将改变为下层中值为 1 的节点。

[0093] 图 3 和图 4 显示了空位种子 1\*1\*1 和 11\*\*1 的修改结果。结构有向图的开始节点是顶级节点。这种有向图可以作为重叠有向图的初级版本,即无重叠有向图。

[0094] 3) 重叠步骤

[0095] 重叠步骤是一系列重复迭代的过程。迭代过程以步骤为基础,依次以每层节点值为 1 的节点作为开始节点构造结构有向图,但起始节点采用下层中值为 1 的点。结构有向图的首次修改是从第一层开始进行,因此下一步必须从第二层节点中值 1 节点开始,以此类推。

[0096] 当到最后一层时,我们可以看出它第一层相同,因此我们停在 L-1 层。

[0097] 经过重叠步骤操作后就创建了重叠有向图。

[0098] 图 5 和图 6 是空位种子 1\*1\*1 和 11\*\*1 的有向图完整版本。重叠有向图的层数:  $N(layers)=2L-2$ 。

[0099] 步骤 4:计算连通性和重叠数

[0100] 1) 连通性计算:有两个有公共结点的边  $e_1$  和  $e_2$ ,即  $e_1$  的结束节点是  $e_2$  的开始节点,设  $B_1$  是  $e_1$  标记的集合,  $B_2$  是  $e_2$  标记的集合。称  $B_1$  和  $B_2$  具有连通性或是连通的当且仅当  $B_1 \cap B_2 \neq \emptyset$ 。

[0101] 例如,在空位种子 1\*1\*1 的重叠有向图中,公共结点节点(3)的两条边有连通性,因为  $B_1=\{1\}$  且  $B_2=\{1\}$  那么  $B_1 \cap B_2 = \{1\} \neq \emptyset$ ,但对于具有公共节点(2)的两条边(最左边)不连通,因为  $B_1=\{1\}$  且  $B_2=\{2\}$  那么  $B_1 \cap B_2 = \emptyset$ 。

[0102] 路径上所有的边都是连通的路径称为连通的路径。当且仅当一条连通的路径中的 L 条边都包含相同的某个标记,才说路径中包含空位种子的模式。

[0103] 2) 重叠数计算:设 B 是边 e 的标记集合,定义重叠数:

[0104]  $oc(e)=|B|$

[0105] 连通性和重叠数是判断空位种子质量的两个主要因素。如果某空位种子有高的连通性,称其是优质的,因为可以检测出更多的连通路径从而可以找到更多的模式。另一方面,优质空位种子要有低的重叠数,因为重叠的命中将检测出同一处的匹配,那么低重叠数将有高的敏感度。

[0106] 步骤 5 :基于最优空位种子的匹配

[0107] 1) 边权重的计算 :边的权重是连通性和重叠数的函数。定义如下 :

[0108]

$$w(e) = \sum_{e' \in \{ \text{joint edges of } e \}} \varpi_{e'} \{ \varphi[oc(e), p], p \}$$

[0109] 其中  $oc(e)$  是  $e$  的重叠数,  $\varphi(\bullet, p)$  是“重叠惩罚函数”;

[0110] 2) 定义重叠惩罚及连通奖励函数 :

[0111]

$$\varphi(c, p) = \begin{cases} p^c & (e \text{ 的结束节点为 } 1) \\ (1-p)^c & (e \text{ 的结束节点为 } 0) \end{cases}$$

[0112] 其中  $c=oc(e)$ 。可见当  $p<1$  时,  $\varphi(c, p)$  是  $c$  的减函数。

[0113]  $\varpi_{e'}(\bullet, p)$  是奖励边连通性的“奖励函数”,也就是说,如果  $e$  与其具有公共结点的边是连通的则  $w(e)$  应该增加。设  $e'$  是  $e$  的某个具有公共结点的边,在本发明中,定义  $\varpi_{e'}(\bullet, p)$  如下 :

[0114]

$$\varpi_{e'}(\varphi, p) = \begin{cases} \varphi + p & (e \text{ 与 } e' \text{ 连通且 } e' \text{ 的结束结点为 } 1) \\ \varphi + (1-p) & (e \text{ 与 } e' \text{ 连通且 } e' \text{ 的结束结点为 } 0) \\ \varphi & (e \text{ 与 } e' \text{ 不连通}) \end{cases}$$

[0115] 其中  $\varphi$  是重叠惩罚函数  $\varphi(c, p)$  的值。

[0116] 对于结点  $N$ ,定义  $w(N)=w(e)*w(e')$ ,用于计算结点的权值。

[0117] 3) 计算重叠有向图权重

[0118] 重叠有向图权重定义为图中每条边的权值之和。给定一个重叠有向图  $G$ ,设  $e$  是  $G$  的边数那么重叠有向图的权重  $ODW(G)$  定义如下 :

$$[0119] ODW(G) = \sum_{j=1}^e w(N_j)$$

[0120] 某重叠有向图的  $ODW(G)$  值主要基于空位种子的结构,重叠数,连通性和相似度等等。由于涉及更多的参数且允许用户自定义函数来计算,重叠有向图模型将更加完整,精确而灵活。

[0121] 根据此模型,对于相同长度和权重的空位种子,更优质的种子有更高的重叠有向图权重,在计算候选种子的  $ODW(G)$  后就可以找到优质种子。

[0122] 4) 基于最优空位种子的匹配

[0123] 按照重叠有向图权重最大值针对每一个候选种子进行计算,寻找最优空位种子,并以些为标准进行 WEB 服务描述匹配,其特征在于包括以下步骤 :

[0124] a) 首先确定一个终止值 d、连续种子长度 w 和一个阈值 t。d 值通常是基于统计学的原理指明一个预期的终止 E 值，然后在考虑搜索背景性质的基础上计算出合适的 d 值。

[0125] b) 根据空位种子模式对内存中的 WEB 服务描述源数据进行局部匹配，当有一个匹配串的分值高于 t 时，则找到了一个选中的字串，即增强点。

[0126] c) 当一个分值高于 t 的字串选中后，进行局部延伸寻优，规定比对的最低分值是 d，当比对延伸时会遇到一些负的分值，使得比对的分值下降，当下降的分值小于 d 时，命中的延伸就会终止，这时这段比对中具有最高分值的片段便成为一段命中的匹配，从而找到符合要求的 WEB 服务结点。

[0127] 本领域的普通技术人员将会意识到，这里所述的实施例是为了帮助读者理解本发明的原理，应被理解为本发明的保护范围并不局限于这样的特别陈述和实施例。本领域的普通技术人员可以根据本发明公开的这些技术启示做出各种不脱离本发明实质的其它各种具体变形和组合，这些变形和组合仍然在本发明的保护范围内。

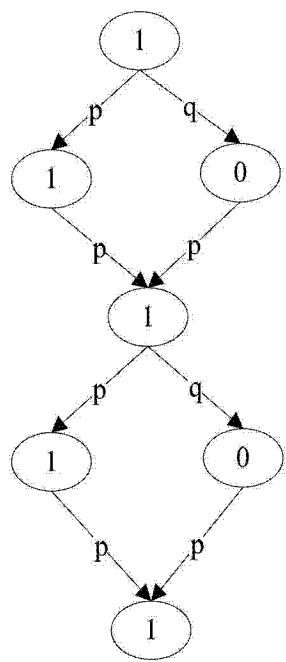


图 1

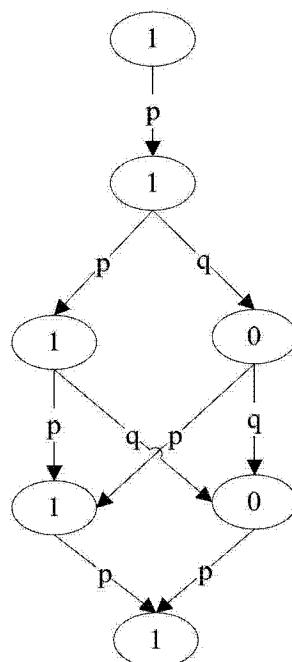


图 2

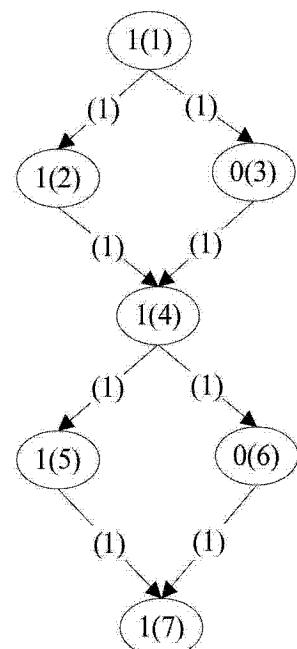


图 3

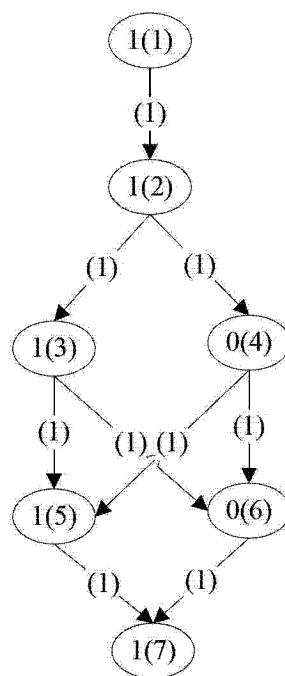


图 4

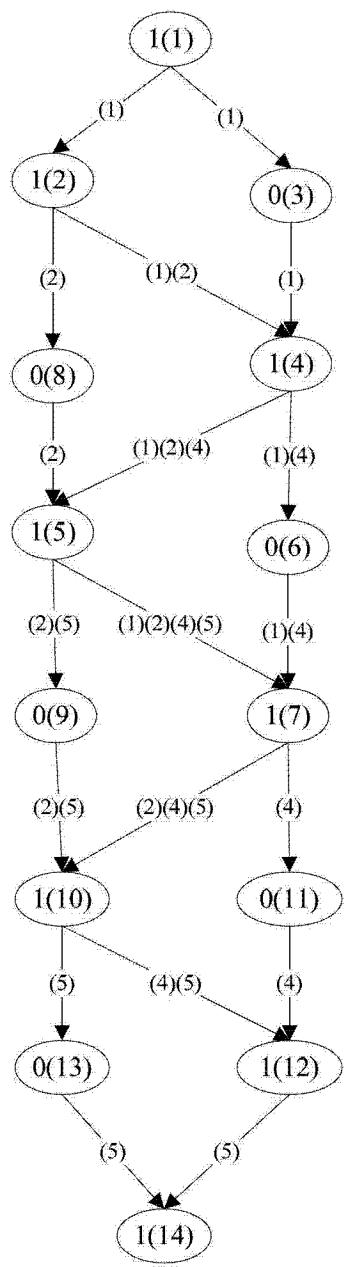


图 5

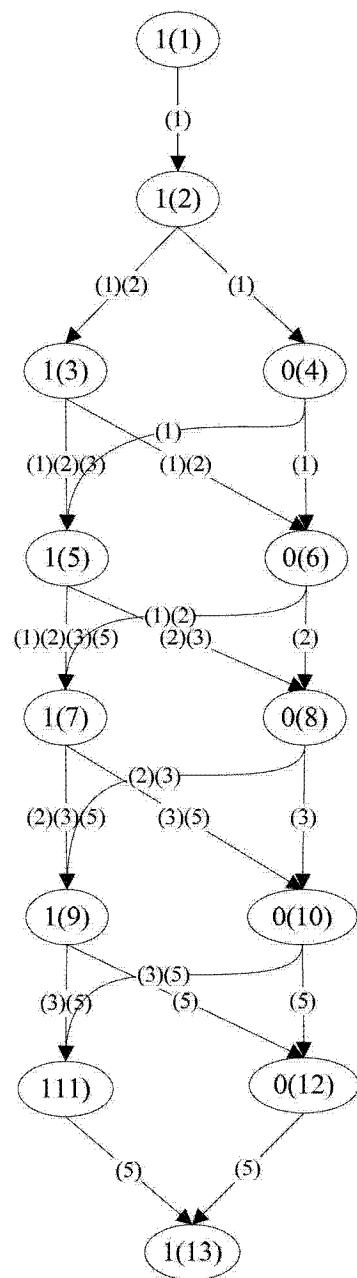


图 6