软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(8):1992−2001 [doi: 10.3724/SP.J.1001.2012.04168]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

容错的网络声明式程序*

汪 芳[1,2,3+], Stéphane GRUMBACH[4]

[1](中国联合网络通信集团有限公司研究院,北京 100048)

[2](中国科学院 软件研究所 计算机科学国家重点实验室,北京 100190)

[3](中国科学院 研究生院,北京 100049)

[4](Institut National de Recherche en Informatique et en Automatique, France)

Fault Tolerant Network Declarative Programs

WANG Fang[1,2,3+], Stéphane GRUMBACH[4]

[1](China Unicom Research Institute, Beijing 100048, China)

[2](State Key Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[3](Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

[4](Institut National de Recherche en Informatique et en Automatique, France)

+ Corresponding author: E-mail: wangfang313@chinaunicom.cn

Wang F, Grumbach S. Fault tolerant network declarative programs. *Journal of Software*, 2012,23(8): 1992−2001 (in Chinese). http://www.jos.org.cn/1000-9825/4168.htm

Abstract: This paper introduces the syntax and the distributed fixpoint semantics of a rule-based declarative language, Netlog. The strongly well-behaved programs were defined, which were proven insensitive to bounded message loss.

Key words: network programming abstraction; declarative language; syntax; semantics; fault tolerance

摘 要: 介绍了基于递归规则的网络声明式语言 Netlog 的语法和分布式不动点语义,定义了强良好的程序,并证明了强良好的程序的计算结果对有限的消息丢失不敏感.

关键词: 网络编程抽象;声明式语言;语法;语义;容错性

中图法分类号: TP311 **文献标识码**: A

近年来,无线通信技术的发展渗透和微电子嵌入式计算设备种类的快速增加,加速了普适计算的发展.然而,资源的多样性、设备的不稳定性、网络的动态性以及数据的密集性等特点又为普适计算的发展带来各种挑战.开发和维护无线网络设备,需要开发者深入地了解不同设备的硬件配置、操作系统和技术标准,是一件极其复杂的工作,成为实现普适计算的一个瓶颈.现在,亟需一种面向应用的高层的网络编程抽象方案,将面向用户和应用的逻辑层与面向设备的物理层分开,使网络开发者从底层的细节中脱离出来,集中注意力于高层的应用逻辑上[1].

近几年来,人们尝试将声明式的数据查询语言引入网络编程.这个想法最初起源于传感器网络领域.Fung 等

* 基金项目: 国家自然科学基金(60833001); French Agence Nationale de la Recherche (ANR-09-BLAN-0131-01)
收稿时间: 2010-09-30; 定稿时间: 2011-01-26

人提出,可以将整个网络看成一个数据库,而本地通过声明式查询来实现与网络的互动[2].Cougar 系统[3]和TinyDB 系统[4]等无线传感器系统用 SQL 编写查询,提供了高效的数据查询和计算的解决方案.在对网络的拓扑结构和节点设备容载的全局认识的基础上,Strivastava 等人以集中式的方法设计分布式的查询执行计划,使子查询有的放矢,从而使执行计划得以优化[5].基于传感器数据的空间和时态的特征,声明式方法还被 Jeffery 等人运用于清理网络上不安全的数据[6].

Loo 等人提出用“声明式网络”[7],即用递归的声明式网络查询语言来实现较为复杂的网络通信算法,例如路由协议[8]和覆盖网络协议[7];同时,研究者也演示了如何用递归的查询语言来执行如异步系统诊断[9]、网络监控[10]以及自组织的网络协议[11]等网络操作.基于 Datalog 的优化方法,Loo 等人进而提出了声明式网络的分布式执行计划[12].这为表达网络上的各种复杂问题,如节点探索[13]、路由发现、根据服务级别的路径维护[14]、拓扑探索,包括物理拓扑探索[15]、安全的网络[16]以及自适应的移动 AdHoc 网络路由[17]提供了一种新方法.

与传统的网络编程语言相比,声明式网络语言具有简洁(比传统的命令式语言程序小 1~2 个数量级)、与底层实现无关、可形式化地定义语义等优点[18].Grumbach 等人定义了一种基于规则的递归的声明式网络语言 Netlog[19].Netlog 对数据库领域的演绎语言 Datalog[20]加以扩展,融入了一系列网络计算需要的原语.其最基本的特征是,程序是局部的,节点之间不共享存储.这个特征使得确定否定原子的值变得简单.另外,Netlog 还包括算术函数、聚合函数和非确定性选择操作用来定义网络上的计算.经典的递归规则语言在集中式环境下的不动点语义已经得到了充分的研究[20].而 Netlog 语言具有在网络环境下的分布式不动点语义,明确地定义了节点之间的通信,尤其是路由器的语义.

本文讨论 Netlog 程序的容错性.首先定义了强良好的 Netlog 程序,在其限制条件中,一些是对集中式环境下 Datalog 语言的限制的扩展,如安全性、膨胀性和单调性;另一些是针对网络环境下 Netlog 语言的限制,如通信的谨慎性和原子性.本文证明了强良好的程序在网络上的计算结果不受有限的消息丢失的影响.

本文第 1 节介绍 Netlog 语言的计算模型.第 2 节介绍 Netlog 语言的语法和分布式不动点语义.第 3 节定义强良好的 Netlog 程序.第 4 节证明此类程序的容错性.第 5 节进行总结.

1 计算模型

Netlog 语言的计算模型以分布式计算的消息传递系统[21]为基础.网络的拓扑结构由图 $\boldsymbol{G}=(V,Link)$定义.其中,V 是网络中节点的集合,$Link$ 是节点间的双向通信连接的集合.节点有唯一的身份标识,即 ID,取自 1,2,…,$|V|$.数据以关系表的形式分布在各个节点上,节点不与其他节点共享数据或者控制其他节点.节点之间依靠消息进行通信,消息中含有消息的内容和消息的目标地址,形式为〈消息内容,目标地址〉.其中,消息的内容是由 Netlog 规则演绎的关系的实例,即事实.消息的目标地址为某个节点的 ID,或 ngh(此时,消息发送给所有邻居节点),或 all(此时,消息广播给所有节点).将发生在节点内的计算事件和节点间传递消息的通信事件区分开来.节点上的一个计算事件连同一个通信事件构成一个回合,而一系列回合构成一个执行序列.

节点有 3 个模块:路由器、引擎和本地存储,具有完全相同的行为.在回合 l,节点α的行为如下:(1) 路由器将接收到的消息放入接收队列 $\mathcal{R}^{\alpha}(l+1)$,并根据目标地址分成两组:(i) 如果目标地址是$\alpha$或 all,那么消息的内容组成 $\mathcal{L}^{\alpha}(l+1)$,在下一回合开始时送给演绎引擎;(ii) 如果目标地址是其他节点的 ID 或 all,那么这些消息组成 $\mathcal{F}^{\alpha}(l+1)$,并被加入到发送队列 $\mathcal{P}^{\alpha}(l+1)$中.同时,路由器将发送队列 $\mathcal{P}^{\alpha}(l)$中的消息发送给其他节点:如果消息的目标地址是 all,那么发送给所有邻居节点;否则,在本地路由表中查找发往目标地址的路径信息,将消息发给该路径的下一跳.如果查找失败,则采取寻路失败策略,如丢弃消息.(2) 演绎引擎从本地数据存储中载入被 $\mathcal{L}^{\alpha}(l)$中的事实触发的程序,循环地演绎程序的规则,直到不再演绎新的事实或者不再删除事实,更新本地数据库,并将部分演绎的事实作为消息送给路由器的发送队列 $\mathcal{P}^{\alpha}(l+1)$.(3) 数据存储保存两类数据:(i) 节点相关的数据,即网络信息(例如拓扑、路由、通信代价等)或者与应用相关的数据;(ii) 程序.

2 Netlog 语言

Netlog 语言具有明确定义的语法和分布式不动点语义,并且在网络模拟平台以及无线传感器网络上测试了分布式生成树、简化的 AODV、DSDV、OLSR、连通支配集等协议[18,19].

2.1 语 法

区分两个类:有序自然数$(N,\leqslant)$(用作节点的标识)和算术类$(R,\leqslant,+,\times)$.实际 Netlog 语言使用标准编程语言的数据类型.当不重要或者上下文清晰时,本文忽略这些数据类型.

- 常量:即在 N 和 R 类上的常量,或者只用作地址的常量 *all* 或 *ngh*.
- 变量:假设给定一个无限集的变量,记为 $x,y,z,\ldots$.
- 算数函数:在 R 类上的$+,-,\times,\div$.
- 聚合函数:在任何类上的 Min,Max 或计数函数(记为#),或在 R 类上的平均值函数(记为 *Avg*).
- 项:简单项或者复杂项.简单项是常量或者变量,复杂项为聚合项、任择项、地址项或算数项.聚合项的形式为 $Aggr(x)$,其中,$Aggr$ 表示一个聚合函数,x 是变量.任择项的形式为$\Diamond x$,其中,x 是变量.地址项的形式为$@t$,其中,t 是 N 类上的项,或者是常量 *all* 或 *ngh*.算数项的形式为 $t_1\theta t_2$,其中,t_1,t_2 是 R 类上的项,θ表示一个算数函数.聚合项和任择项只能出现在规则头中.
- 关系原子:形式为 $R(t_1,\ldots,t_n)$的表达式,其中,R 是元为 n 的关系记号,$t_1,\ldots,t_n$ 是恰当的类上的项.
- 比较原子:形式为 $t_1\theta t_2$ 的表达式,其中,θ是$=,>,\geqslant,<,\leqslant$或$\neq$,$t_1,t_2$ 是相同类上的项.
- 赋值原子:形式为 $x:=t$ 的表达式,其中,x 是简单变量,t 是 x 不出现在其中的项.
- 正字段或原子:关系原子、比较原子或赋值原子.
- 否定字段:形式为$\neg A$,其中,A 为关系原子.
- 全称字段:形式为$\neg A$,其中,A 为关系原子,其部分参数未指明,记为“_”.
- 删除字段:形式为$!A$,其中,A 是关系原子.
- 字段:正字段、否定字段、全称字段或删除字段.
- 计时声明:间歇声明或时效声明.间歇声明是形式为$[inv:t]$的表达式,时效声明是形式为$[exp:t]$的表达式,其中 t 在 N 类上($[inv:t]$表示每隔 t 个单位时间触发该规则,$[exp:t]$表示演绎的事实的有效时间).
- 存储规则:形式为$\downarrow A\ T_1 :- L_1;\ldots;L_l\ T_2$ 的表达式,其中,$l\geqslant 0$.规则头,即 $A\ T_1$ 包含一个关系原子 A 和一个时效声明 T_1,这两者都可以为空.规则身,即 $L_1;\ldots;L_l\ T_2$,包含 l 个字段 $L_1,\ldots,L_l$ 和一个间歇声明 T_2,这两者都可以为空.存储规则需符合以下条件:
  1) 规则中所有的变量存在于规则身中的正字段、删除字段或者赋值原子的左边.
  2) 规则头中的变量不同时存在于头中的简单项、聚合项或任择项中(例如,$R(x,y,\Diamond x,\mathrm{Min}(y))$不允许作为规则头).
  3) 规则身不包含聚合项或任择项,并且包括最多 1 个地址项$@x$,且 x 是变量.
- 发送规则:形式为$\uparrow A :- L_1;\ldots;L_l\ T$.的表达式,除了存储规则的条件以外,还需满足以下条件:

  A 中最多有 1 个参数可以是地址项.
- 存储并发送规则:形式为$\uparrow\downarrow A\ T_1 :- L_1;\ldots;L_l\ T_2$ 的表达式,需满足发送规则的条件.
- 规则:存储规则、发送规则、存储并发送规则.
- 确定性规则:不含任择项的规则;否则,为非确定性规则.
- 程序:规则的有限集合.

将程序 P 的存储规则的子集记为 $P_\downarrow$,将 P 的发送规则的子集记为 $P_\uparrow$.存储并发送规则同时属于这两个子集.

2.2 分布式不动点语义

假设网络在某个足够长的时间段内是稳定的,直到程序的分布式不动点收敛.因此,网络拓扑在这段时间内不变,并且可以忽略计时声明.下面的定义中先忽略↑/↓操作符,即忽略节点之间的通信,定义在单个节点上

Netlog 程序的执行效果.

给定关系模式(shema)S,其中包含关系 $Link$ 和 $Route$.给定 S 中的关系的解释的有限集合,即 S 的实例 I.对任何关系 $R\in S$,令 R^I 为 I 对 R 的解释.以下将$(\alpha_1,\ldots,\alpha_n)\in R^I$ 写作 $R(\alpha_1,\ldots,\alpha_n)\in I$.

给定变量的有限集合 V,V 上的赋值是从 V 到 $N\cup R\cup\{all,ngh\}$的一个映射.给定规则 r,r 中包含且仅包含 S 中的关系,令 $Var(r)$为 r 中的变量的集合,令 $\mathcal{V}(Var(r))$为 $Var(r)$上的赋值的集合.对任何$\sigma\in\mathcal{V}(Var(r))$,将$\sigma$的域扩展到 r 的规则身中的项:

- σ对常量解释为该常量;
- 对于含有算数函数θ的项,$\sigma(t_1\theta t_2)=\sigma(t_1)\theta\sigma(t_2)$.

令 I 为节点α上的 S 的一个实例.I 和σ满足 r 的规则身的定义如下:

- 对关系原子,(I,σ) :– $R(t_1,\ldots,t_n)$,当且仅当 $R(\sigma(t_1),\ldots,\sigma(t_n))\in I$;
- 对比较原子,(I,σ) :– $t_1\theta t_2$,当且仅当 R :– $\sigma(t_1)\theta\sigma(t_2)$;
- 对赋值原子,(I,σ) :– $x:=t$,当且仅当$\sigma(x)=\sigma(t)$;
- 对否定字段,(I,σ) :– $\neg R(t_1,\ldots,t_n)$,当且仅当 $R(\sigma(t_1),\ldots,\sigma(t_n))\notin I$;
- 对全称字段,(I,σ) :– $\neg R(t_1,\ldots,\_,\ldots,t_n)$,当且仅当对任意常量 C,$R(\sigma(t_1),\ldots,C,\ldots,\sigma(t_n))\notin I$;
- 对删除字段,(I,σ) :–$!R(t_1,\ldots,t_n)$,当且仅当 $R(\sigma(t_1),\ldots,\sigma(t_n))\in I$;
- 对 r 的规则身(记为 $body_r$),假设 $body_r$ 为 $L_1;\ldots;L_l$,那么(I,σ) :– $body_r$ 当且仅当对任何 $i\in[1,l]$都有(I,σ) :– L_i,并且,如果$@x$ 是 $body_r$ 中的地址项,那么$\sigma(x)=\alpha$.

令 $STVar(head_r)$为 r 的规则头(记为 $head_r$)中不出现在聚合项或任择项中的变量的集合,令 $NAVar(head_r)$为 $head_r$ 中不出现在聚合项中的变量的集合,令$\tau\in\mathcal{V}(NAVar(head_r))$,记$\tau$的域,即 $NAVar(head_r)$为 $dom(\tau)$.τ的延伸的定义如下,它将τ的域扩展到 $Var(r)$:

$$[\tau]_{I,r}=\{\sigma|\sigma\in\mathcal{V}(Var(r));\text{对于所有 }x\in\mathrm{dom}(\tau)\text{都有}\sigma(x)=\tau(x);\text{并且}(I,\sigma)\text{ :– }body_r\}.$$

下面假设$[\tau]_{I,r}\neq\varnothing$.$\tau(head_r)$的定义:

- 如果 $head_r$ 不包含聚合项,即 $Var(head_r)=NAVar(head_r)$,形式为 $R(x_1,\ldots,x_n,\Diamond z_1,\ldots,\Diamond z_l)$,那么,
$$\tau(head_r)=R(\tau(x_1),\ldots,\tau(x_n),\tau(z_1),\ldots,\tau(z_l)).$$
- 如果 $head_r$ 的形式为 $R(x_1,\ldots,x_n,Aggr(y_1),\ldots,Aggr(y_m),\Diamond z_1,\ldots,\Diamond z_l)$,那么,
$$\tau(head_r)=R(\tau(x_1),\ldots,\tau(x_n),Aggr\{\{\sigma(y_1)|\sigma\in[\tau]_{I,r}\}\},\ldots,Aggr\{\{\sigma(y_m)|\sigma\in[\tau]_{I,r}\}\},\tau(z_1),\ldots,\tau(z_l)),$$

其中,$\{\{\cdot\}\}$表示多重集,$Aggr$ 表示多重集上的聚合函数.

规则头中的地址项$@x$ 的赋值和变量 x 的赋值一致,并保留记号@.

下面给出程序 P 在实例 I 上执行的正结果的集合 Δ_P^+ 和负结果的集合 Δ_P^- 定义.先对规则 r 定义集合 $PPC_r(I)$:

$$PPC_r(I)=\{\tau(head_r)|\tau\in\mathcal{V}(NAVar(head_r)),[\tau]_{I,r}\neq\varnothing\}.$$

对非确定性规则 r,可能的正结果的集合为满足函数依赖 $STVar(head_r)\rightarrow NAVar(head_r)$的 $PPC_r(I)$的子集,即包含对所有任择项作唯一的选择的事实的子集;对确定性规则 r,只有 1 个正结果的集合等于 $PPC_r(I)$.假设 $P=\{r_i|i\in[1,\zeta]\}$,令 $\mathcal{P}r_i(I)$为满足上述条件的 $PPC_r(I)$的子集的集合,那么 $\mathcal{P}_P(I)=\left\{\bigcup_{i=1}^{\zeta}J_{r_i}\mid J_{r_i}\in\mathcal{P}_{r_i}(I)\right\}$. 定义程序 P 的正结果的集合为

$$\Delta_P^+(I)\doteq J,\text{其中},J\in\mathcal{P}_P(I),$$

其中,$\doteq$ 表示非确定性映射.对确定性程序,$\doteq$ 可以用=代替.如果程序 P 中不含任择项,那么,

$$\Delta_P^+(I)=\{\tau(head_r)|r\in P,\tau\in\mathcal{V}(NAVar(head_r)),[\tau]_{I,r}\neq\varnothing\}.$$

类似地,定义程序 P 的负结果集合为

$$\Delta_P^-(I)=\{\sigma(A)|r\in P,(I,\sigma)\text{ :– }body_r,!A\text{ 在 }body_r\text{ 中}\}.$$

显然,$\Delta_P^-(I)\subseteq I$.

下面考虑节点之间的通信,给出 Netlog 语言的分布式不动点语义的定义.假设网络上的所有节点都已载入程序 P.先考虑其中一个节点α上的计算和通信.节点上的计算被定义为两个函数:存储函数 $\Psi_P^{\downarrow}$,即需要更新的本地实例和发送函数 $\Psi_P^{\uparrow}$,即需发送给其他节点的消息的集合.它们以节点上的本地实例 I 和接收到的事实的集合 L 作为参数.

- $\Psi_P^{\downarrow}(I,L) \doteq \Delta_{P\downarrow}^{+}(I \cup L) \cup (I \setminus \Delta_P^{-}(I \cup L))$;
- $\Psi_P^{\uparrow}(I,L) \doteq \{(R(\alpha_1,...,\alpha_n),d) \mid R(\alpha_1,...,\alpha_n) \in \Delta_{P\uparrow}^{+}(I \cup L), d = ngh$;或$R(\alpha_i,...,@\alpha_i,...,\alpha_n) \in \Delta_{P\uparrow}^{+}(I \cup L), d = \alpha_i\}$.

在程序的执行的定义之前,需要先定义事实以及实例上的偏序关系.

定义 1. 令 R 是元为 n 的关系,它的一些属性(假设其标记的集合为 $\mathcal{A}\subseteq[1,n]$)在偏序域$(D,\prec)$上.对事实 $R(\alpha_1,\ldots,\alpha_n)$和 $R(\alpha_1',...,\alpha_n')$,如果对任何 $i\in\mathcal{A}$ 都有 $\alpha_i \prec \alpha_i'$,那么 $R(\alpha_1,\ldots,\alpha_n) \prec R(\alpha_1',...,\alpha_n')$.对于 D 上的实例 I 和 I',如果对任何事实 $p\in I$ 都存在 $p'\in I'$满足 $p\prec p'$,那么 $I\prec I'$.如果 $I\prec I'$,并且 $I'\prec I$,那么记为 $I\fallingdotseq I'$.

在每一个回合,节点循环地执行存储和发送函数直到不动点.

定义 2. 给定 Netlog 程序 P、节点α上的实例 I^{α}和接收到的事实的集合 L^{α},P 在α上关于 I^{α}和 L^{α}的单回合执行是一个序列$(I_i^{\alpha},\mathcal{P}_i^{\alpha})$,其中,$(I_i^{\alpha})$是本地实例的序列,$(\mathcal{P}_i^{\alpha})_i$是累积的消息的集合的序列:

- $I_0^{\alpha} \doteq \Psi_P^{\downarrow}(I^{\alpha},L^{\alpha})$;
- $I_{i+1}^{\alpha} \doteq \Psi_P^{\downarrow}(I_i^{\alpha},\varnothing), i\geqslant 0$;
- $\mathcal{P}_0^{\alpha} \doteq \Psi_P^{\uparrow}(I^{\alpha},L^{\alpha})$;
- $\mathcal{P}_{i+1}^{\alpha} \doteq \Psi_P^{\uparrow}(I_i^{\alpha},\varnothing) \cup \mathcal{P}_i^{\alpha}, i\geqslant 0$.

对每一个这样的序列,令 u 为满足 $I_{u+1}^{\alpha} \fallingdotseq I_u^{\alpha}$ 的最小数,那么$(I_u^{\alpha},\mathcal{P}_u^{\alpha})$为程序 P 在α上关于 I^{α}和 L^{α}的单回合不动点.如果每一个单回合执行都收敛,那么 P 在α上关于 I^{α}和 L^{α}的单回合计算终止.

注意,接收到的事实参加计算但是不存储到节点上,而需要发送的消息累积在 $\mathcal{P}_i^{\alpha}$ 中而不参加计算.

下面考虑节点间的通信.节点α与网络之间的数据交换由通信函数 $\mathcal{R}^{\alpha}$定义.它定义了节点α接收到的消息和其他节点发送的消息的对应.

定义 3. $\mathcal{R}^{\alpha}(l+1)$为节点$\alpha$在回合 l 接收到的消息.根据消息的地址,$\mathcal{R}^{\alpha}(l+1)$被划分为两个集合:接收的事实的集合 $\mathcal{L}^{\alpha}(l+1)$和需转发的消息集合 $\mathcal{F}^{\alpha}(l+1)$.

- $\mathcal{L}^{\alpha}(0)=\varnothing$;
- $\mathcal{L}^{\alpha}(l)=\{fact \mid (fact,dest)\in\mathcal{R}^{\alpha}(l), dest\in\{\alpha,ngh,all\}\}, l\geqslant 1$;
- $\mathcal{F}^{\alpha}(l)=\{(fact,dest) \mid (fact,dest)\in\mathcal{R}^{\alpha}(l), dest\notin\{\alpha,ngh\}\}, l\geqslant 1$.

回合 l 开始时, 节点α上有本地实例 $I^{\alpha}(l)$、接收到的事实的集合 $\mathcal{L}^{\alpha}(l)$和需要发送的消息的集合 $\mathcal{P}^{\alpha}(l)$.在回合 l,节点α执行本地计算直到单回合执行序列收敛,如果达到不动点$(I_{u_l}^{\alpha}(l),\mathcal{P}_{u_l}^{\alpha}(l))$,则产生一个新的本地实例 $I^{\alpha}(l+1)=I_{u_l}^{\alpha}(l)$,以及需要发送给其他节点的消息集合 $\mathcal{P}^{\alpha}(l+1)=\mathcal{P}_{u_l}^{\alpha}(l)\cup\mathcal{F}^{\alpha}(l+1)$.同时,节点发送 $\mathcal{P}^{\alpha}(l)$中的消息.路由器查找本地路由表得到到目标地址的路径,将消息发送给下一跳节点;并将接受到的消息 $\mathcal{R}^{\alpha}(l+1)$划分为两组:在回合 $l+1$ 将参加本地计算的事实的集合 $\mathcal{L}^{\alpha}(l+1)$和需要转发的消息集合 $\mathcal{F}^{\alpha}(l+1)$.

程序在每个节点上一系列回合的执行序列以及程序的分布式不动点定义如下:

定义 4. 假设程序 P 在网络 G 的节点α上关于实例 I^{α}和一系列回合接收到的消息集合$(\mathcal{R}^{\alpha}(l))_l$计算,那么 P 在α上关于 I^{α}和$(\mathcal{R}^{\alpha}(l))_l$的执行是一个序列$(I^{\alpha}(l),\mathcal{P}^{\alpha}(l))_l$,其中,

- $I^{\alpha}(0)=I^{\alpha}$;
- $\mathcal{P}^{\alpha}(0)=\varnothing$;

对 $l\geqslant 0$,如果 P 在α上关于 $I^{\alpha}(l)$和 $\mathcal{L}^{\alpha}(l)$的单回合计算终止,并且$(I_{u_l}^{\alpha}(l),\mathcal{P}_{u_l}^{\alpha}(l))$为单回合不动点,那么,

- $I^{\alpha}(l+1)=I_{u_l}^{\alpha}(l)$;

- $\mathcal{P}^{\alpha}(l+1)=\mathcal{P}^{\alpha}_{u_l}(l)\cup\mathcal{F}^{\alpha}(l+1)$.

其中,$\mathcal{F}^{\alpha}(l)$和 $\mathcal{L}^{\alpha}(l)$在定义 3 定义.如果存在 $w\in N$,使得对每个 $w'\geqslant w$ 都有 $I^{\alpha}(w')\fallingdotseq I^{\alpha}(w)$,那么 $I^{\alpha}(w)$为 P 在α上关于 I^{α}和$(\mathcal{R}^{\alpha}(l))_l$ 的不动点.如果对于每个节点$\alpha\in V$,P 在α上关于 I^{α}和$(\mathcal{R}^{\alpha}(l))_l$ 的执行收敛于不动点 $I^{\alpha}(w_{\alpha})$,那么 P 在 G 上关于 I 和$(\mathcal{R}^{\alpha})_{\alpha\in V}$的计算终止,不动点的集合$(I^{\alpha}(w_{\alpha}))_{\alpha\in V}$为程序 P 的分布式不动点.

对于同步且通信无错的系统,它的通信函数是明确的,我们将其记为$(\mathcal{R}^{\alpha}_{syn})_{\alpha\in V}$.

性质 1. 对于同步且通信无错的系统,假设程序 P 在网络 G 上关于实例 I 和通信函数$(\mathcal{R}^{\alpha}_{syn})_{\alpha\in V}$的计算在节点$\alpha\in V$ 上的执行序列为$(I^{\alpha}_{syn}(l),\mathcal{P}^{\alpha}_{syn}(l))_l$,那么,

$$R^{\alpha}_{syn}(l+1)=\left\{(fact,dest)\ \middle|\ \begin{array}{l}\exists\beta\ \text{s.t.}\ Link(\beta,\alpha)\in I^{\beta}_{syn}(l);\ (fact,dest)\in P^{\beta}_{syn}(l);\ \text{and}\\ \text{if}\ dest\notin\{\alpha,ngh,all\},\ \text{then}\ Route(\beta,\alpha,dest)\in I^{\beta}_{syn}(l).\end{array}\right\},l\geqslant 0.$$

3 强良好的 Netlog 程序

本节定义强良好的 Netlog 程序,它们在多项式时间内终止计算(假设网络拓扑在某个足够长的时间段内是稳定的).对强良好的 Netlog 程序的限制条件中:一些是对集中式环境下 Datalog 语言的限制的扩展,如安全性、膨胀性和单调性;而另一些是针对网络环境下 Netlog 语言的限制,如通信的谨慎性和原子性.

首先定义安全的程序,这类程序在任何网络上都可以在有限域上计算.先定义域受限的赋值如下:给定变量的有限集合 V 和有限域 D,V 上的受限于 D 的赋值是从 V 到 D 的映射.定义程序的域受限的语义如下:

令 $\mathcal{V}_D(V)$为 V 上的受限于 D 的赋值的集合.给定规则 r,令$\tau\in\mathcal{V}_D(NAVar(head_r))$.定义$\tau$的延伸如下:

$$[\tau]_{D,I,r}=\{\sigma|\sigma\in\mathcal{V}_D(Var(r)),(I,\sigma)\ :\!-\ body_r\ \text{且对于所有}\ x\in\mathrm{dom}(\tau),\sigma(x)=\tau(x)\}.$$

基于以上定义,可以类似地定义规则头的域受限的赋值,由此可以类似地定义程序的域受限的分布式不动点.给定程序 P 和实例 I,将 P 或者 I 中出现的常量的集合记为 $cons(P,I)$.定义安全的程序如下:

定义 5. 对于程序 P,如果存在 $k\in N$,使得 P 在任何网络 G 上关于任何实例 I 和任何通信函数$(\mathcal{R}^{\alpha})_{\alpha\in V}$的计算都具有一个域 D,满足$|D|\leqslant|cons(P,I)|^k$,且每一个节点$\alpha\in V$上的每一个回合的单回合执行和一个受限于 D 的单回合执行一致,那么 P 是安全的.

定义拟膨胀的程序如下:

定义 6. 如果程序 P 对于任何实例 I 和 L 满足 $I\prec\Psi^{\downarrow}_P(I,L)$,那么 P 是拟膨胀的.

性质 2. 拟膨胀的程序 P 在任何网络 G 上关于任何实例 I 和任何通信函数$(\mathcal{R}^{\alpha})_{\alpha\in V}$的计算满足:对于任何$\alpha\in V$、任何 $l'>l\geqslant 0$ 和任何 $i'\geqslant i\geqslant 0$,都有 $I^{\alpha}(l)\prec I^{\alpha}_i(l)\prec I^{\alpha}_{i'}(l)\prec I^{\alpha}(l')$.

在定义程序的单调性之前,需要先将偏序关系$\prec$延伸到包含实例和消息的集合的结构上.

定义 7. 对于节点α上的结构$(I_1,\mathcal{P}_1)$和$(I_2,\mathcal{P}_2)$,其中,I_1 和 I_2 为实例,$\mathcal{P}_1$ 和 $\mathcal{P}_2$ 为消息的集合,如果 $I_1\prec I_2$ 并且对任何消息$(f_1,d_1)\in\mathcal{P}_1$,都存在消息$(f_2,d_2)\in\mathcal{P}_2$ 满足 $f_1\prec f_2$,且 d_1 和 d_1 满足以下任一条件,那么$(I_1,\mathcal{P}_1)\prec(I_2,\mathcal{P}_2)$:

- 如果 $d_1=\beta$,那么 $d_2=\beta$,或者 $d_2=ngh$ 并且 $Link(\alpha,\beta)\in I_2$,或者 $d_2=all$;
- 如果 $d_1=ngh$,那么 $d_2=ngh$,或者 $d_2=all$;
- 如果 $d_1=all$,那么 $d_2=all$.

如果$(I_1,\mathcal{P}_1)\prec(I_2,\mathcal{P}_2)$且$(I_2,\mathcal{P}_2)\prec(I_1,\mathcal{P}_1)$,那么记$(I_1,\mathcal{P}_1)\fallingdotseq(I_2,\mathcal{P}_2)$.

定义 8. 给定程序 P,对任何节点α上的实例 I_1,I_2 和接收到的事实的集合 L_1,L_2,如果 $I_1\prec I_2$ 并且 $L_1\prec L_2$,那么,$(\Psi^{\downarrow}_P(I_1,L_1),\Psi^{\uparrow}_P(I_1,L_1))\prec(\Psi^{\downarrow}_P(I_2,L_2),\Psi^{\uparrow}_P(I_2,L_2))$, 则 P 是$\prec$保持的.

定义 9. 对于程序 P,令 S 为 P 的发送规则的规则头中的关系的集合.如果对于任何实例 I 以及 S 上的实例 L 都有 $\Psi^{\uparrow}_P(I,L)=\Psi^{\uparrow}_P(I,\varnothing)$,那么 P 具有谨慎的通信.

定义 10. 对于程序 P,令 S 为 P 的发送规则的规则头中的关系的集合.如果对于任何实例 I,S 上的实例 L 和

任何 $f\in\Psi_P^{\downarrow}(I,L)$,都存在 $p\in L$ 使得 $f\in\Psi_P^{\downarrow}(I,\{p\})$,那么 P 具有原子的通信.

定义 11. 如果程序是安全的、拟膨胀的和$\prec$保持的,并且具有谨慎的和原子的通信,那么它是强良好的.

4 容错性

本节讨论 Netlog 程序对于网络通信错误的容错能力.为了便于讨论,假设程序是确定性的.另外,假设网络在足够长的时间段内是稳定的,并且假设消息延迟的时间有常数的上限.

定理 1. 强良好的程序在网络上的计算结果不受有限个数的消息丢失的影响.

定理 1 的证明依赖于下面几个引理.将节点在一个回合内的最长计算时间称为节点单回合计算复杂度.

引理 1. 给定网络 G,安全且拟膨胀的程序在 G 上计算,其节点单回合计算复杂度多项式地依赖于实例大小.

证明:令 P 为一个安全且拟膨胀的程序.令 P 在网络 G 上关于实例 I 计算.因为 P 是拟膨胀的,由性质 2,在每个节点α的每个回合 l 的实例序列$(I_i^{\alpha}(l))_i$ 是关于$\prec$递增的.因为 P 是安全的,由定义 5,存在 $k\in N$ 和域 D,满足 $|D|\leqslant|cons(P,I)|^k$ 且 $cons(P,I)\subseteq D$,使得在每个节点α的每个回合 l 的单回合执行序列$(I_i^{\alpha}(l),\mathcal{P}_i^{\alpha}(l))_i$和一个受限于 D 的单回合执行序列一致.所以,每个回合 l 的序列$(I_i^{\alpha}(l))_i$和序列$(\mathcal{P}_i^{\alpha}(l))_i$关于$\subseteq$都有多项式的势的上限.显然,它们关于$\prec$都有多项式的势的上限.所以,序列$(I_i^{\alpha}(l))_i$在多项式多的步内收敛. □

基于引理 1,当强良好的程序在网络上的计算时,令单位时间为节点单回合计算时间的上限,即为一个依赖于实例大小的常数.那么,每个节点的每个单回合计算需要最多 1 个单位时间;同时,假定每个单回合计算的时间下限为$\kappa,\kappa\leqslant 1$.为简单起见,对节点上的实例和消息的集合的结构$(I_1,\mathcal{P}_1)$和$(I_2,\mathcal{P}_2)$,当上下文明显 $I_1\prec I_2$ 时,以下将$(I_1,\mathcal{P}_1)\prec(I_2,\mathcal{P}_2)$简写成 $\mathcal{P}_1\prec\mathcal{P}_2$.

引理 2. 强良好的程序 P 在任何网络 G 上关于任何实例 I 和任何通信函数$(\mathcal{R}^{\alpha})_{\alpha\in V}$的计算都满足:对任何节点$\alpha\in V$上的任何回合 l 和 l',如果 $l'\geqslant l$,那么 $\lim_{i\to\infty}\mathcal{P}_i^{\alpha}(l)\prec\lim_{i\to\infty}\mathcal{P}_i^{\alpha}(l')$.

证明:假设强良好的程序 P 在网络 G 上计算.因为程序是拟膨胀的,由性质 2,对每个节点α上的任何回合 l 和 l',如果 $l'\geqslant l$,那么对任何 $i\geqslant 0$, $I_i^{\alpha}(l)\prec I^{\alpha}(l')$.又因为强良好的程序具有谨慎的通信并且是$\prec$保持的,所以 $\mathcal{P}_0^{\alpha}(l)=\Psi_P^{\uparrow}(I^{\alpha}(l),\varnothing)\prec\Psi_P^{\uparrow}(I^{\alpha}(l'),\varnothing)=\mathcal{P}_0^{\alpha}(l')$.于是,对任何 $i\geqslant 0$,

$$\mathcal{P}_{i+1}^{\alpha}(l)=\Psi_P^{\uparrow}(I_i^{\alpha}(l),\varnothing)\cup\mathcal{P}_i^{\alpha}(l)\prec\Psi_P^{\uparrow}(I_i^{\alpha}(l'),\varnothing)\cup\mathcal{P}_i^{\alpha}(l')=\mathcal{P}_{i+1}^{\alpha}(l').$$

因此,$\lim_{i\to\infty}\mathcal{P}_i^{\alpha}(l)\prec\lim_{i\to\infty}\mathcal{P}_i^{\alpha}(l')$. □

引理 3. 强良好的程序 P 在同步且通信无错的网络 G 上关于任何实例 I 的计算都满足:对任何节点$\alpha\in V$上的任何回合 l 和 l',如果 $l'\geqslant l$,那么 $\mathcal{L}^{\alpha}(l)\prec\mathcal{L}^{\alpha}(l')$.

证明:假设强良好的程序 P 在同步且通信无错的网络 G 上计算.对任何节点$\alpha\in V$上的任何事实 $f\in\mathcal{L}^{\alpha}(l)$,都存在$\beta\in V\setminus\{\alpha\}$以及回合 $t<l$,使得存在包含 f 的消息 $m\in\lim_{i\to\infty}\mathcal{P}_i^{\beta}(t)$.

由引理 2,对所有 $t'\geqslant t$,$\lim_{i\to\infty}\mathcal{P}_i^{\beta}(t)\prec\lim_{i\to\infty}\mathcal{P}_i^{\beta}(t')$.所以存在消息 $m'\in\lim_{i\to\infty}\mathcal{P}_i^{\alpha}(t+l'-l)$,满足 $m\prec m'$.假设 m 包含事实 f',那么 $f\prec f'$.又由性质 1,$f'\in\mathcal{L}^{\alpha}(l')$.因此,$\mathcal{L}^{\alpha}(l)\prec\mathcal{L}^{\alpha}(l')$. □

引理 4. 强良好的程序 P 在有限个消息丢失的网络 G 上关于任何实例 I 的计算都满足:对任何节点$\alpha\in V$上的任何回合 l,以及任何事实 $p\in\mathcal{L}^{\alpha}(l)$和任何回合 $t<l$,都存在 $t'\geqslant t$,使得$\{p\}\prec\mathcal{L}^{\alpha}(t')$.

证明:假设强良好的程序 P 在有限个消息丢失的网络 G 上计算.对任何节点$\alpha\in V$上的任何事实 $f\in\mathcal{L}^{\alpha}(l)$,都存在$\beta\in V\setminus\{\alpha\}$以及回合 s,使得存在包含 f 的消息 $m\in\lim_{i\to\infty}\mathcal{P}_i^{\beta}(s)$.由引理 2 可知,对所有 $s'\geqslant s+j$,都有

$$\lim_{i\to\infty}\mathcal{P}_i^{\beta}(s)\prec\lim_{i\to\infty}\mathcal{P}_i^{\beta}(s+j)\prec\lim_{i\to\infty}\mathcal{P}_i^{\beta}(s').$$

又因为消息延迟有常数的上限,所以对任何 $t\geqslant l$ 都存在 $s''\geqslant s+j$ 使得存在消息 $m''\in\lim_{i\to\infty}\mathcal{P}_i^{\beta}(s'')$满足 $m\prec m''$,并且使得存在回合 $t'\geqslant t$ 满足 $p'\in\mathcal{L}^{\alpha}(t')$,其中,$p'$为消息 m''包含的事实.那么 $p\prec p'$,即$\{p\}\prec\mathcal{L}^{\alpha}(t')$. □

引理 5. 假设强良好的程序 P 在网络 G 上关于实例 I 计算,当网络上消息丢失个数为 j 时,P 的计算在节点α

上的执行序列为$(I_{ls}^{\alpha}(l),\mathcal{P}_{ls}^{\alpha}(l))_l$;而当网络同步且通信无错时,$P$ 的计算在节点α上的执行序列为$(I_{syn}^{\alpha}(l),\mathcal{P}_{syn}^{\alpha}(l))_l$.那么,对于任何节点$\alpha\in V$上的任何回合 l,

(1) 存在 $h\geqslant l$,使得 $I_{ls}^{\alpha}(l)\prec I_{syn}^{\alpha}(h)$;

(2) 存在 $s\geqslant l$,使得 $I_{syn}^{\alpha}(l)\prec I_{ls}^{\alpha}(s)$.

证明:假设当网络上消息丢失个数为 j 时,P 的计算在节点$\alpha\in V$ 上的回合 l 的单回合不动点为$(I_{ls,u(\alpha,l)}^{\alpha}(l),\mathcal{P}_{ls,u(\alpha,l)}^{\alpha}(l))$;而当网络同步且通信无错时,$P$ 的计算在节点$\alpha\in V$ 上的回合 l 的单回合不动点为$(I_{syn,v(\alpha,l)}^{\alpha}(l),\mathcal{P}_{syn,v(\alpha,l)}^{\alpha}(l))$.

证明(1):对时刻 w 进行归纳,以κ为递增的单位.

基础:对当网络上消息丢失个数为 j 时,P 的计算在节点$\alpha\in V$ 上的 0 时刻的回合 l_0^{α},$I_{ls}^{\alpha}(l_0^{\alpha})=I_{ls}^{\alpha}(0)=I_{syn}^{\alpha}(0)$,$\mathcal{P}_{ls,u(\alpha,l_0^{\alpha})}^{\alpha}(l_0^{\alpha})=\mathcal{P}_{ls,u(\alpha,0)}^{\alpha}(0)=\mathcal{P}_{syn,v(\alpha,0)}^{\alpha}(0)=\varnothing$,且 $L_{ls}^{\alpha}(l_0^{\alpha})=L_{ls}^{\alpha}(0)=L_{syn}^{\alpha}(0)=\varnothing$.

递归:假设对当网络上消息丢失个数为 j 时,P 的计算在节点$\alpha\in V$ 上的 w 时刻的回合 l_w^{α}:

(a) 存在 $h\geqslant l_w^{\alpha}$,使得 $I_{ls}^{\alpha}(l_w^{\alpha})\prec I_{syn}^{\alpha}(h)$,且 $\mathcal{P}_{ls,u(\alpha,l_w^{\alpha}-1)}^{\alpha}(l_w^{\alpha}-1)\prec\mathcal{P}_{syn,v(\alpha,h-1)}^{\alpha}(h-1)$;

(b) 对任何 $l\leqslant l_w^{\alpha}$ 和任何 $p\in L_{ls}^{\alpha}(l)$,存在 $u\geqslant l$ 满足 $\{p\}\prec L_{syn}^{\alpha}(u)$.

以下证明:对当网络上消息丢失个数为 j 时,P 的计算在节点$\alpha\in V$ 上的 $w+\kappa$时刻的回合 $l_{w+\kappa}^{\alpha}$:

(a′) 存在 $h'\geqslant l_{w+\kappa}^{\alpha}$,使得 $I_{ls}^{a}(l_{w+\kappa}^{\alpha})\prec I_{syn}^{\alpha}(h')$,且 $\mathcal{P}_{ls,u(\alpha,l_{w+\kappa}^{\alpha}-1)}^{\alpha}(l_{w+\kappa}^{\alpha}-1)\prec\mathcal{P}_{syn,v(\alpha,h'-1)}^{\alpha}(h'-1)$;

(b′) 对任何 $l\leqslant l_{w+\kappa}^{\alpha}$ 和任何 $p\in L_{ls}^{\alpha}(l)$,存在 $u'\geqslant l$ 满足 $\{p\}\prec L_{syn}^{\alpha}(u')$.

如果回合 $l_{w+\kappa}^{\alpha}$ 即回合 l_w^{α},则已证;否则,回合 $l_{w+\kappa}^{\alpha}$ 即回合 $l_w^{\alpha}+1$,下面证明此时情形(a′)和情形(b′)成立:

证明(a′):由情形(a),存在 $h\geqslant l_w^{\alpha}$,使得 $I_{ls}^{\alpha}(l_w^{\alpha})\prec I_{syn}^{\alpha}(h)$. 因为 P 具有原子的通信,所以对任何 $f\in I_{ls,0}^{\alpha}(l_w^{\alpha})$,即 $f\in\Psi_P^{\downarrow}(I_{ls}^{a}(l_w^{\alpha}),L_{ls}^{a}(l_w^{\alpha}))$,都有 $p_f\in L_{ls}^{a}(l_w^{\alpha})$ 使得 $f\in\Psi_P^{\downarrow}(I_{ls}^{a}(l_w^{\alpha}),\{p_f\})$. 又由情形(b),存在 $u_f\geqslant l_w^{\alpha}$ 满足 $\{p_f\}\prec L_{syn}^{\alpha}(u_f)$. 令 $h'=\max\{h,\max\{u_f\mid f\in I_{ls,0}^{\alpha}(l_w^{\alpha})\}\}$.因为 P 是拟膨胀的,由性质 2,$I_{syn}^{\alpha}(h)\prec I_{syn}^{\alpha}(h')$,所以,$I_{ls}^{a}(l_w^{\alpha})\prec I_{syn}^{\alpha}(h')$. 由引理 3 可知,$L_{syn}^{\alpha}(u_f)\prec L_{syn}^{\alpha}(h')$,所以 $\{p_f\}\prec L_{syn}^{\alpha}(h')$. 因为 P 是$\prec$保持的,所以,$\Psi_P^{\downarrow}(I_{ls}^{a}(l_w^{\alpha}),\{p_f\})\prec\Psi_P^{\downarrow}(I_{syn}^{\alpha}(h'),L_{syn}^{\alpha}(h'))=I_{syn,0}^{\alpha}(h')$.所以,对任何 $f\in I_{ls,0}^{\alpha}(l_w^{\alpha})$ 都存在 $f'\in I_{syn,0}^{\alpha}(h')$ 满足 $f\prec f'$,即 $I_{ls,0}^{\alpha}(l_w^{\alpha})\prec I_{syn,0}^{\alpha}(h')$. 因为 P 具有谨慎的通信并且是$\prec$保持的,所以 $\mathcal{P}_{ls,0}^{\alpha}(l_w^{\alpha})=\Psi_P^{\uparrow}(I_{ls}^{\alpha}(l_w^{\alpha}),\varnothing)\prec\Psi_P^{\uparrow}(I_{syn}^{\alpha}(h'),\varnothing)=\mathcal{P}_{syn,0}^{\alpha}(h')$.

于是,对任何 $i\geqslant 0$,$I_{ls,i+1}^{\alpha}(l_w^{\alpha})=\Psi_P^{\downarrow}(I_{ls,i}^{\alpha}(l_w^{\alpha}),\varnothing)\prec\Psi_P^{\downarrow}(I_{syn,i}^{\alpha}(h'),\varnothing)=I_{syn,i+1}^{\alpha}(h')$,并且

$$\mathcal{P}_{ls,i+1}^{\alpha}(l_w^{\alpha})=\Psi_P^{\uparrow}(I_{ls,i}^{\alpha}(l_w^{\alpha}),\varnothing)\cup\mathcal{P}_{ls,i}^{a}(l_w^{\alpha})\prec\Psi_P^{\uparrow}(I_{syn,i}^{\alpha}(h'),\varnothing)\cup\mathcal{P}_{syn,i}^{\alpha}(h')=\mathcal{P}_{syn,i+1}^{\alpha}(h').$$

所以,$I_{ls,u(\alpha,l_w^{\alpha})}^{\alpha}(l_w^{\alpha})\prec I_{syn,v(\alpha,h')}^{\alpha}(h')$,即 $I_{ls}^{a}(l_w^{\alpha}+1)\prec I_{syn}^{\alpha}(h'+1)$,并且 $\mathcal{P}_{ls,u(\alpha,l_w^{\alpha})}^{\alpha}(l_w^{\alpha})\prec\mathcal{P}_{syn,v(\alpha,h')}^{\alpha}(h')$.

证明(b′):因为回合 l_w^{α} 最迟在 $w+\kappa$时刻结束,所以α在回合 l_w^{α} 收到的消息最迟由另一个节点β在回合 $l_{w+\kappa}^{\alpha}$ 发出,即对任何事实 $p\in L_{ls}^{a}(l_w^{\alpha}+1)$,存在节点$\beta\in V\backslash\{\alpha\}$和回合 $l_p\leqslant l_{w+\kappa}^{\beta}$,使得存在消息$(p,d)\in\mathcal{P}_{ls,u(\beta,l_p-1)}^{\beta}(l_p-1)$. 由引理 2 可知,因为 $l_p\leqslant l_{w+\kappa}^{\beta}$,所以 $\mathcal{P}_{ls,u(\beta,l_p-1)}^{\beta}(l_p-1)\prec\mathcal{P}_{ls,u(\beta,l_{w+\kappa}^{\beta}-1)}^{\beta}(l_{w+\kappa}^{\beta}-1)$. 由情形(a′),存在 $h'\geqslant l_{w+\kappa}^{\alpha}$ 使得

$$\mathcal{P}_{ls,u(\beta,l_{w+\kappa}^{\beta}-1)}^{\beta}(l_{w+\kappa}^{\beta}-1)\prec\mathcal{P}_{syn,v(\beta,h'-1)}^{\beta}(h'-1).$$

由引理 2 可知,对任何 $h''\geqslant h'$,$\mathcal{P}_{syn,v(\beta,h'-1)}^{\beta}(h'-1)\prec\mathcal{P}_{syn,v(\beta,h''-1)}^{\beta}(h''-1)$,所以,$\mathcal{P}_{ls,u(\beta,l_p-1)}^{\beta}(l_p-1)\prec\mathcal{P}_{syn,v(\beta,h''-1)}^{\beta}(h''-1)$. 因此,存在 $h''\geqslant h'$以及消息$(p',d')\in\mathcal{P}_{syn,v(\beta,h''-1)}^{\beta}(h''-1)$ 满足$(p,d)\prec(p',d')$,并且使得α在回合 $u'\geqslant l_{w+\kappa}^{\alpha}-1$接收消息$(p,d)$,即 $p'\in L_{syn}^{\alpha}(u'+1)$,则 $\{p\}\prec L_{syn}^{\alpha}(u'+1)$.联合情形(b),可得情形(b′).

证明(2):对 l 进行归纳.

基础:对当网络同步且通信无错时 P 的计算在每个节点$\alpha\in V$ 上的回合 0,$I_{syn}^{\alpha}(0)=I_{ls}^{a}(0)$,

$$\mathcal{P}^{\alpha}_{syn,u(\alpha,0)}(0)=\mathcal{P}^{\alpha}_{ls,v(\alpha,0)}(0)=\varnothing,\text{且 } L^{\alpha}_{syn}(0)=L^{\alpha}_{ls}(0)=\varnothing.$$

递归:假设对当网络同步且通信无错时 P 的计算在每个节点$\alpha\in V$上的回合 l:

(c) 存在 $s\geqslant l$,使得 $I^{\alpha}_{syn}(l)\prec I^{a}_{ls}(s)$, 且 $\mathcal{P}^{\alpha}_{syn,v(\alpha,l-1)}(l-1)\prec\mathcal{P}^{\alpha}_{ls,u(\alpha,s-1)}(s-1)$;

(d) 对任何 $t\leqslant l$ 和任何 $p\in L^{\alpha}_{syn}(t)$, 存在 $v\geqslant t$ 满足 $\{p\}\prec L^{\alpha}_{syn}(v)$.

以下证明:对当网络同步且通信无错时 P 的计算在每个节点$\alpha\in V$上的回合 l+1:

(c′) 存在 $s'\geqslant l$+1,使得 $I^{\alpha}_{syn}(l+1)\prec I^{a}_{ls}(s')$, 且 $\mathcal{P}^{\alpha}_{syn,v(\alpha,l)}(l)\prec\mathcal{P}^{\alpha}_{ls,u(\alpha,s'-1)}(s'-1)$;

(d′) 对任何 $t\leqslant l$+1 和任何 $p\in L^{\alpha}_{syn}(t)$, 存在 $v\geqslant t$ 满足 $\{p\}\prec L^{\alpha}_{syn}(v)$.

证明(c′):由情形(c),存在 $s\geqslant l$,使得 $I^{\alpha}_{syn}(l)\prec I^{a}_{ls}(s)$. 因为 P 具有原子的通信,所以对任何 $f\in I^{\alpha}_{syn,0}(l)$, 即 $f\in\Psi^{\downarrow}_{P}(I^{\alpha}_{syn}(l),L^{\alpha}_{syn}(l))$, 都有 $p_f\in L^{\alpha}_{syn}(l)$, 使得 $f\in\Psi^{\downarrow}_{P}(I^{\alpha}_{syn}(l),\{p_f\})$. 又由情形(d),存在 $v'_f\geqslant l$ 使得存在 $p'\in L^{\alpha}_{ls}(v'_f)$ 满足 $p_f\prec p'$.由引理 4,存在最小的 v_f 满足 $v_f\geqslant\max\{v'_f,s\}$, 使得存在 $p''\in L^{\alpha}_{ls}(v_f)$ 满足 $p'\prec p''$,即 $\{p_f\}\prec L^{\alpha}_{ls}(v_f)$.

因为 P 是拟膨胀的,由性质 2, $I^{\alpha}_{ls}(s)\prec I^{\alpha}_{ls}(v_f)$, 所以 $I^{\alpha}_{syn}(l)\prec I^{\alpha}_{ls}(v_f)$.

因为 P 是$\prec$保持的,所以 $\Psi^{\downarrow}_{P}(I^{\alpha}_{syn}(l),\{p_f\})\prec\Psi^{\downarrow}_{P}(I^{\alpha}_{ls}(v_f),L^{\alpha}_{ls}(v_f))=I^{\alpha}_{ls,0}(v_f)$.

又因为 $\{f\}\prec\Psi^{\downarrow}_{P}(I^{\alpha}_{syn}(l),\{p_f\})$, 所以 $\{f\}\prec I^{\alpha}_{ls,0}(v_f)$.

令 $s'=\max\{v_f\mid f\in I^{\alpha}_{syn,0}(l)\}$. 因为 P 是拟膨胀的,由性质 2,对任何 $f\in I^{\alpha}_{syn,0}(l)$ 都有 $I^{\alpha}_{ls,0}(v_f)\prec I^{\alpha}_{ls,0}(s')$. 又因为 $\{f\}\prec I^{\alpha}_{ls,0}(v_f)$, 即 $\{f\}\prec I^{\alpha}_{ls,0}(s')$, 所以 $I^{\alpha}_{syn,0}(l)\prec I^{\alpha}_{ls,0}(s')$.

同情形(a′)的证明,有 $I^{\alpha}_{syn}(l+1)\prec I^{\alpha}_{ls}(s'+1)$ 和 $\mathcal{P}^{\alpha}_{syn,v(\alpha,l)}(l)\prec\mathcal{P}^{\alpha}_{ls,u(\alpha,s')}(s')$.

证明(d′):因为对任何 $p\in L^{\alpha}_{syn}(l+1)$,存在$\beta\in V\backslash\{\alpha\}$和回合 $l_p\leqslant l$ 满足存在$(p,d)\in\mathcal{P}^{\beta}_{syn,v(\beta,l_p-1)}(l_p-1)$. 由情形(c),存在 $s\geqslant l$,使得 $\mathcal{P}^{\beta}_{syn,v(\beta,l-1)}(l-1)\prec\mathcal{P}^{\beta}_{ls,u(\beta,s-1)}(s-1)$. 所以存在$(p',d')\in\mathcal{P}^{\beta}_{ls,u(\beta,s-1)}(s-1)$ 且 $p'\in L^{\alpha}_{ls}(s+1)$, 满足$(p,d)\prec(p',d')$,即 $\{p\}\prec L^{\alpha}_{ls}(s+1)$. 联合情形(d),可得情形(d′). □

定理 1 的证明:假设通信函数$(\mathcal{R}^{\alpha})_{\alpha\in V}$定义了有限的消息丢失.由引理 5,良好的程序 P 在任何网络 G 上关于任何实例 I 和$(\mathcal{R}^{\alpha})_{\alpha\in V}$ 的计算在任何节点$\alpha\in V$ 上的执行序列$(I^{\alpha}(l),\mathcal{P}^{\alpha}(l))_l$ 收敛,当且仅当 P 在 G 上关于 I 和通信函数$(\mathcal{R}^{\alpha}_{syn})_{\alpha\in V}$的计算在$\alpha$上的执行序列$(I^{\alpha}_{syn}(l),\mathcal{P}^{\alpha}_{syn}(l))_l$收敛,并且具有相等的不动点. □

5 总 结

Netlog 是一种基于规则的递归的网络声明式语言,用来表达较为复杂的网络计算问题.Netlog 语言:(1) 融入了丰富的网络应用必需的原语;(2) 具有定义明确的分布式不动点语义;(3) 已经在 Netquest 虚拟器上应用,并在网络模拟器和 iMode 设备的实验平台上测试.本文定义了强良好的 Netlog 程序,并证明其对通信具有容错性.在对强良好的 Netlog 程序的限制条件中:一些是对集中式环境下 Datalog 语言的限制的扩展,如安全性、膨胀性和单调性;另一些是针对网络环境下 Netlog 语言的限制,如通信的谨慎性和原子性.下一阶段我们考虑 Netlog 程序对节点上数据的容错性,以及可适应高度动态网络的 Netlog 程序.

致谢 感谢林惠民教授和吴志林博士参与 Netlog 语言的研究工作,并给予宝贵的建议.

References:

[1] Marron J, Minder D. Embedded WiSeNts Research Roadmap. Berlin: Springer-Verlag, 2006.

[2] Fung WF, Sun D, Gehrke J. Cougar: The network is the database. In: Proc. of the SIGMOD Conf. 2002. 621. [doi: 10.1145/564691.564775]

[3] Demers AJ, Gehrke J, Rajaraman R, Trigoni A, Yao Y. The cougar project: A work-in-progress report. SIGMOD Record, 2003, 32(4):53–59. [doi: 10.1145/959060.959070]

[4] Madden S, Franklin MJ, Hellerstein JM, Hong W. Tinydb: An acquisitional query processing system for sensor networks. ACM Trans. on Database Systems, 2005,30(1):122 −173. [doi: 10.1145/1061318.1061322]

[5] Srivastava U, Munagala K, Widom J. Operator placement for in-network stream query processing. In: Proc. of the 24th ACM Symp. on Principles of Database Systems. 2005. 250−258. [doi: 10.1145/1065167.1065199]

[6] Jeffery SR, Alonso G, Franklin MJ, Hong W, Widom J. Declarative support for sensor data cleaning. pervasive computing. In: Proc. of the 4th Int'l Conf. 2006. 83−100. [doi: 10.1007/11748625_6]

[7] Loo BT, Condi T, Hellerstein JM, Maniatis P, Roscoe T, Stoica I. Implementing declarative overlays. In: Proc. of the 20th ACM Symp. on Operating Systems Principles. Brighton, 2005. 75−90. [doi: 10.1145/1095810.1095818]

[8] Loo BT, Hellerstein JM, Stoica I, Ramakrishnan R. Declarative routing: Extensible routing with declarative queries. In: Proc. of the ACM SIGCOMM 2005 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. Philadelphia, 2005. 289−300. [doi: 10.1145/1080091.1080126]

[9] Abiteboul S, Abrams Z, Haar S, Milo T. Diagnosis of asynchronous discrete event systems: Datalog to the rescue! In: Proc. of the 24th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems. Baltimore, 2005. 358−367. [doi: 10.1145/1065167.1065214

[10] Reiss F, Hellerstein JM. Declarative network monitoring with an underprovisioned query processor. In: Proc. of the ICDE. 2006. 56. [doi: 10.1109/ICDE.2006.46]

[11] Grumbach S, Lu JL, Qu WW. Self-Organization of wireless networks through declarative local communication. In: Proc. of the OTM, On the Move Conf. LNCS 4805, 2007. 497−506. [doi: 10.1007/978-3-540-76888-3_72]

[12] Loo BT, Condie T, Garofalakis MN, Gay DE, Hellerstein JM, Maniatis P, Ramakrishnan R, Roscoe T, Stoica I. Declarative networking: Language, execution and optimization. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Chicago, 2006. 97−108. [doi: 10.1145/1142473.1142485]

[13] Alonso G, Kranakis E, Sawchuk C, Wattenhofer R, Widmayer P. Probabilistic protocols for node discovery in ad hoc multi-channel broadcast networks. In: Proc. of the 2nd Int'l Conf. on Ad-Hoc, Mobile, and Wireless Networks (ADHOC-NOW). 2003. 104−115.

[14] Bejerano Y, Breitbart Y, Orda Y, Rastogi R, Sprintson A. Algorithms for computing QoS paths with restoration. IEEE/ACM Trans. on Network, 2005,13(3):648 −661. [doi: 10.1109/TNET.2005.850217]

[15] Bejerano Y, Breitbart Y, Garofalakis MN, Rastogi R. Physical topology discovery for large multi-subnet networks. In: Proc. of the INFOCOM. 2003. 342−352. [doi: 10.1109/INFCOM.2003.1208686]

[16] Abadi M, Loo BT. Towards a declarative language and system for secure networking. In: Proc. of the 3rd USENIX Int'l Workshop on Networking Meets Databases (NETB 2007). Berkeley, 2007. 1−6.

[17] Liu C, Mao Y, Oprea M, Basu P, Loo BT. A declarative perspective on adaptive MANET routing. In: Proc. of the ACM Workshop on Programmable Routers for Extensible Services of Tomorrow (PRESTO 2008). New York: ACM Press, 2008. 63−68. [doi: 10.1145/1397718.1397733]

[18] Bauderon M, Grumbach S, Gu D, Qi X, Qu W, Suo K, Zhang Y. Programming iMote networks made easy. In: Proc. of the 4th Int'l Conf. on Sensor Technologies and Applications (SENSORCOMM 2010). Venice, Mestre: CPS Press, 2010. 539−544. [doi: 10.1109/SENSORCOMM.2010.87]

[19] Grumbach S, Wang F. Netlog, a rule-based language for distributed programming. In: Proc. of the 12th Int'l Symp. on Practical Aspects of Declarative Languages (PADL 2010). Madrid: Springer-Verlag, 2010. 88−103. [doi: 10.1007/978-3-642-11503-5_9]

[20] Abiteboul S, Hull R, Vianu V. Foundations of Databases. Boston: Addison-Wesley, 1995.

[21] Attiya H, Welch J. Distributed Computing: Fundamentals, Simulations and Advanced Topics. Wiley-Interscience, 2004.

汪芳(1981－),女,安徽安庆人,博士,工程师,CCF 会员,主要研究领域为数据库理论,查询语言,并行计算.

Stéphane GRUMBACH(1962－),男,博士,教授,博士生导师,主要研究领域为数据库理论,查询语言,逻辑和复杂度,网络,分布式编程.