上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

THESIS OF BACHELOR

论文题目： 随机进程模型研究

学生姓名： 孙晨鸽

学生学号： 516030910421

专　　业： 计算机科学与技术

指导教师： 傅育熙教授

学院(系)： 电子信息与电气工程学院

随机进程模型研究

摘　要

中文摘要应该将学位论文的内容要点简短明了地表达出来，应该包含论文中的基本信息，体现科研工作的核心思想。摘要内容应涉及本项科研工作的目的和意义、研究方法、研究成果、结论及意义。注意突出学位论文中具有创新性的成果和新见解的部分。摘要中不宜使用公式、化学结构式、图表和非公知公用的符号和术语，不标注引用文献编号。硕士学位论文中文摘要字数为 500 字左右，博士学位论文中文摘要字数为 800 字左右。英文摘要内容应与中文摘要内容一致。

摘要页的下方注明本文的关键词（4 ~ 6 个）。

关键词：上海交大, 饮水思源, 爱国荣校

A SAMPLE DOCUMENT FOR LaTeX-BASED SJTU THESIS TEMPLATE

ABSTRACT

Shanghai Jiao Tong University (SJTU) is a key university in China. SJTU was founded in 1896. It is one of the oldest universities in China. The University has nurtured large numbers of outstanding figures include JIANG Zemin, DING Guangen, QIAN Xuesen, Wu Wenjun, WANG An, etc.

SJTU has beautiful campuses, Bao Zhaolong Library, Various laboratories. It has been actively involved in international academic exchange programs. It is the center of CERNet in east China region, through computer networks, SJTU has faster and closer connection with the world.

Key words: SJTU, master thesis, XeTeX/LaTeX template

目　录

第一章 引言

1.1 背景与现状

1.2 并发进程模型

1.3 传值并发模型

1.4 随机进程模型

1.5 目的与意义

第二章 随机传值进程模型

2.1 Random VPC 语法

根据 Uniform Approach，我们将 Uniform Approach 中的 CCS 部分替换为 VPC，很容易得到 Random VPC 的语法。

$$T ::= \bigoplus_{i\in I} p_i\tau.T_i \mid \sum_{i\in I}\varphi_i\lambda_i.T_i \mid T \mid T' \mid (c)T \mid \varphi T \mid !a(x).T \mid !\bar{a}(t).T$$

Uniform Approach 修改了 CCS 的 prefix 操作符，在 CCS 的基础上增加了 $p\tau.T$，使 τ 操作成为一个随机性的操作。

在扩展 VPC 时，我们也将重点关注 prefix 操作符。

在 VPC 的 Symbolic Semantics 中，对于 VPC 的典型 prefix 运算 $\varphi\lambda.T \xrightarrow{\lambda}_{\varphi} T$，我们可以通过 Uniform Approach 的方法将它扩展为一个概率的操作 $p\tau.\varphi\lambda.T \xrightarrow{p\tau}_{\top}\xrightarrow{\lambda}_{\varphi} T$。它的语义就会变成：在概率 p 下，我们会经过一个内部 τ 通道进入一个 VPC $\varphi\lambda.T$，若 $\mathsf{Th}\vdash\varphi$（即一阶理论 Th 下，φ 为真），则我们可以经过 λ 通道到达 T，其中 $\lambda\in\{a(x),\bar{a}(x)\mid a\in\mathcal{N}, x\in\mathsf{V}_{\Sigma}, t\in\mathsf{T}_{\Sigma}\}\cup\{\tau\}$。

2.1.1 if then else 语法

这个语法在 VPC 和 CCS 里看起来都非常格格不入，我们尝试用 VPC 的语法来表示所有的 if then else，并给出相应的证明。

在 VPC 中，$\varphi T \equiv if\ \varphi\ then\ T$，$(\varphi T \mid \neg\varphi S) \equiv if\ \varphi\ then\ T\ else\ S$。那么 $\varphi T \xrightarrow{\lambda}_{\neg\varphi}$?，φT 是否等价于 $(\varphi T \mid \neg\varphi 0)$?

Proposition 1 $\varphi 0 = 0$

Proof. $\varphi 0 = \varphi 0 + 0 = \varphi 0 + \top 0 = \varphi 0 + (\varphi\vee\neg\varphi)0 = \varphi 0 + \varphi 0 + \neg\varphi 0 = \varphi 0 + \neg\varphi 0 = (\varphi\vee\neg\varphi)0 = \top 0 = 0$

Proposition 2 $\varphi T = (\varphi T \mid \neg\varphi 0)$

Proof. $\varphi T = (\varphi T \mid 0) = (\varphi T \mid \neg\varphi 0)$ according to Proposition 1.

Proposition 3 $S = \varphi T$，那么 $S \xrightarrow{\epsilon}_{\neg\varphi} 0$

2.1.2 操作子的定义与条件等价集

对于 $S = p\tau.\varphi T + (1-p)\tau.0$，它的意义是在 p 的概率下选择内部通道 τ，若 $\mathsf{Th}\vdash\varphi$，则到达 T 状态。在 VPC 的 Symbolic Semantic 下，它会经过这样的转移：$S \xrightarrow{p\tau}\xrightarrow{\epsilon}_{\varphi} T$。考虑是否可以将其写成这种形式：$S \xRightarrow{p}_{\varphi} T$。

Definition 1 若 $S, T \in \mathcal{T}_{\mathbb{VPC}}$，$S \xrightarrow{\tau}_{\varphi_1}\xrightarrow{\tau}_{\varphi_2}\cdots\xrightarrow{\tau}_{\varphi_n} T$，则可以记成 $S \Rightarrow_{\varphi_1\varphi_2\cdots\varphi_n} T$。

Definition 2 若 $S,T \in \mathcal{P}_{RCCS}$，$S \overset{p_1\tau}{\rightarrow}\overset{p_2\tau}{\rightarrow} \cdots \overset{p_n\tau}{\rightarrow} T$，则记为 $S \overset{p_1p_2\cdots p_n}{\Rightarrow} T$。

以上，定义 1 在 VPC 中有相同的定义，而 Uniform Approach 中没有类似定义 2 的定义，但在方便表达的情境下，我们新增了这样的记号。

Definition 3 若 $S,T \in \mathcal{T}_{RVPC}$，$S \overset{p_1\tau}{\rightarrow}_{\varphi_1} \overset{p_2\tau}{\rightarrow}_{\varphi_2} \cdots \overset{p_n\tau}{\rightarrow}_{\varphi_n} T$，则可以记成 $S \overset{p_1p_2\cdots p_n}{\Rightarrow}_{\varphi_1\varphi_2\cdots\varphi_n} T$。若 $\varphi = \varphi_1\varphi_2\cdots\varphi_n, p = p_1p_2\cdots p_n$，则 $S \overset{p}{\Rightarrow}_{\varphi} T$。

Definition 4 若 $S = \varphi p\tau.T$，那么 $S \overset{p\tau}{\rightarrow}_{\varphi} T$

Definition 5 $A, A' \in \mathcal{T}_{RVPC}$，若 $\varphi A\mathcal{E}\varphi A'$，则 $A' \in [A]_{\varphi\mathcal{E}}$。$[A]_{\varphi\mathcal{E}}$ 称为等价关系 $\mathcal{E}$ 在条件 φ 下包含 A 的等价集。

定义 5 的产生是由于，在 symbolic bisimulation 中，我们考虑用必要条件下的动作去模拟充分条件下的动作。举个简单的例子对于 $A(x) = (x \geq 5)\tau.B(s(x))$ 和 $A'(x) = (x \geq 3)\tau.B(s(x))$，$A(x)$ 和 $A'(x)$ 是不符号互模拟的，因为不存在 $x \geq 3$ 的划分，使得划分中的每一项推出 $x \geq 5$ 的一个划分中的每一项。但是 $(x \geq 5)A(x)$ 和 $(x \geq 5)A'(x)$ 却是互模拟的。在 $[A]_{\varphi\mathcal{E}}$ 内部，我们可以使 φ 对等价性的影响透明。

2.2 Epsilon Tree

Definition 6 smybolic bisimulation 在 $\mathcal{T}_{VPC}$ 上的定义 VPC Page 181（此处略）

Definition 7 若 $A \in \mathcal{T}_{RVPC}, A$ 的 silent tree t 满足如下定义：

- 每一个节点都被标记成 $\mathcal{T}_{RVPC}$ 的一个元素，A 是根节点。
- 边被标记成 (φ, p)，其中 $p \in (0,1]$，φ 是一个布尔表达式。如果一条从 A' 到 A'' 的有向边被标记成 (φ, p)，表示 $A' \overset{p\tau}{\rightarrow}_{\varphi} A''$。特别的，如果存在 $A' \overset{\tau}{\rightarrow}_{\varphi} A''$，那么 A' 到 A'' 的边标记为 $(\varphi, 1)$；存在 $A' \overset{p\tau}{\rightarrow} A''$，$A'$ 到 A'' 的边标记为 $(\top, 1)$。

Definition 8 A 的 silent tree 是一个 $\varphi\mathcal{E} - tree\ t^A_{\varphi\mathcal{E}}$ 当且仅当树中的所有结点属于 $[A]_{\varphi\mathcal{E}}$。

Definition 9 φq-transition。$A \in \mathcal{T}_{RVPC}, \mathcal{B} \in (\mathcal{T}_{RVPC}/\varphi\mathcal{E})\backslash\{[A]_{\varphi\mathcal{E}}\}$，对 $t^A_{\varphi\mathcal{E}}$ 的每一个叶子结点 $L \xrightarrow{\coprod_{i\in I} p_i\tau}_{\coprod_{i\in I}\psi_i} \coprod_{i\in[k]} L_i$, 存在 $i \in I, L_i \in \mathcal{B}$ 且 $\mathsf{Th} \vdash \varphi \Rightarrow \psi_i$。定义 $\mathsf{P}_{\varphi}(L \xrightarrow{\coprod_{i\in I} p_i\tau}_{\coprod_{i\in I}\psi_i} \mathcal{B}) = \sum\{p_i \mid L \overset{p_i\tau}{\rightarrow}_{\psi_i} L_i \in \mathcal{B} \wedge i \in I \wedge \mathsf{Th} \vdash \varphi \Rightarrow \psi_i\}$。定义 $\mathsf{P}_{\varphi\mathcal{E}}(L \xrightarrow{\coprod_{i\in I} p_i\tau}_{\coprod_{i\in I}\psi_i} \mathcal{B}) = \mathsf{P}_{\varphi}(L \xrightarrow{\coprod_{i\in I} p_i\tau}_{\coprod_{i\in I}\psi_i} \mathcal{B})/(1 - \mathsf{P}_{\varphi}(L \xrightarrow{\coprod_{i\in I} p_i\tau}_{\coprod_{i\in I}\psi_i} [A]_{\varphi\mathcal{E}}))$。当 $\mathsf{P}_{\varphi\mathcal{E}}(L \xrightarrow{\coprod_{i\in I} p_i\tau}_{\coprod_{i\in I}\psi_i} \mathcal{B}) = q$ 时，A 到 $\mathcal{B}$ 有 φq-transition。写作 $A \rightsquigarrow_{\varphi\mathcal{E}} \overset{q}{\rightarrow}_{\varphi} \mathcal{B}$。

Definition 10 一个 $\mathcal{T}_{RVPC}$ 上的等价关系 $\mathcal{E}$ 是一个 symbolic bisimulation，当且仅当，$A\mathcal{E}B$ 则（A, B 对称）

- 若 $A \rightsquigarrow_{\varphi\mathcal{E}} \overset{a(x)}{\rightarrow}_{\varphi} \mathcal{C} \in \mathcal{T}/\varphi\mathcal{E}, \mathcal{C} \neq [A]_{\mathcal{E}}$，则存在 φ 的划分 $\{\varphi_i\}_{i\in I}$，和集合

$\{B \rightsquigarrow_{\varphi_i\mathcal{E}} \xrightarrow{a(x)}_{\psi_i} C\}$，使得对 $i \in I$，$\mathsf{Th} \vdash \varphi_i \Rightarrow \psi_i$。

- 若 $A \rightsquigarrow_{\varphi\mathcal{E}} \xrightarrow{\bar{a}(t)}_{\varphi} C \in \mathcal{T}/\varphi\mathcal{E}$，则存在 φ 的划分 $\{\varphi_i\}_{i\in I}$，和集合 $\{B \rightsquigarrow_{\varphi_i\mathcal{E}} \xrightarrow{\bar{a}(t_i)}_{\psi_i} C\}$，使得对 $i \in I$，$\mathsf{Th} \vdash (\varphi_i \Rightarrow \psi_i) \wedge (t = t_i)$。
- 若 $A \rightsquigarrow_{\varphi\mathcal{E}} \xrightarrow{q}_{\varphi} C \in \mathcal{T}/\mathcal{E}$，则 $B \rightsquigarrow_{\varphi\mathcal{E}} \xrightarrow{q}_{\varphi} C$

2.3 Random VPC 的等价性

Proposition 4 symbolic bisimulation 具有传递性。

Proof. 即证明若 $\mathcal{E}$ 是一个 symbolic bisimulation，$A\mathcal{E}B, B\mathcal{E}C$，则 $A\mathcal{E}C$。

- 若 $A \rightsquigarrow_{\varphi\mathcal{E}} \xrightarrow{\lambda} C \in \mathcal{T}_{RVPC}/\varphi\mathcal{E}$，由于 $A\mathcal{E}B$ 根据定义存在 φ 的划分 $\{\varphi_i\}$ 和集合 $\{B \rightsquigarrow_{\varphi_i\mathcal{E}} \xrightarrow{\lambda}_{\psi} C|\mathsf{Th} \vdash \varphi_i \Rightarrow \psi_i\}$。对于每一个 $B \rightsquigarrow_{\varphi_i\mathcal{E}} \xrightarrow{\lambda}_{\psi_i} C$，$t^B_{\varphi_i\mathcal{E}}$ 上的每一个结点一定属于 $[B]_{\psi_i\mathcal{E}}$，所以 $t^b_{\varphi_i\mathcal{E}}$ 实际上是 $t^b_{\psi_i\mathcal{E}}$ 的子树，$B \rightsquigarrow_{\varphi_i\mathcal{E}} \xrightarrow{\lambda}_{\psi_i} C$ 中的起点终点对 $\{(M,N)|(M \in t^B_{\varphi_i\mathcal{E}}) \vee (N \in C) \vee (M \xrightarrow{\lambda}_{\psi_i} N))\}$ 是 $B \rightsquigarrow_{\psi_i\mathcal{E}} \xrightarrow{\lambda}_{\psi_i} C$ 中的起点终点对 $\{(M',N')|(M' \in t^B_{\psi_i\mathcal{E}}) \vee (N' \in C) \vee (M' \xrightarrow{\lambda}_{\psi_i} N'))\}$ 的子集。由于 $B\mathcal{E}C$，对于每一个 $B \rightsquigarrow_{\psi_i} \xrightarrow{\lambda}_{\psi_i} C$，根据定义存在 ψ_i 的划分 $\{\psi_{i,j}\}$ 和集合 $S = \{C \rightsquigarrow_{\psi_{i,j}\mathcal{E}} \xrightarrow{\lambda}_{\phi_{i,j}} C|\mathsf{Th} \vdash \psi_{i,j} \Rightarrow \phi_{i,j}\}$，那么存在 S 的子集 S' 的可以模拟 $B \rightsquigarrow_{\varphi_i\mathcal{E}} \xrightarrow{\lambda}_{\psi_i} C$。
  我们只需要再构建 φ 的划分 $\{\varphi'_i\}_{i\in I}$ 使其一一对应 S' 中涉及的条件 $\{\phi_{i,k}\}_{k\in K} \subset \{\phi_{i,j}\}_{j\in J}$ 即可。由于存在 ψ_i 的划分 $\{\psi_{i,j}\}_{j\in J}$ 满足 $\psi_{i,j} \Rightarrow \phi_{i,j}\}_{j\in J}$，那么我们可以构造 $Con = \{\psi_{i,k}\}_{k\in[K-1]} \cup \{\bigvee_{j\in J/[K-1]} \psi_{i,j}\}$，对 $c_k \in Con, \bigvee_{k\in K} c_k \Leftrightarrow \psi_i$，且 Con 中元素两两互斥。进而，我们用 $(\bigvee Con_i) \wedge \varphi_i$ 代替 φ_i，即可得到最终的划分。
- φq-transition 的传递性与以上过程相似。

Proposition 5 如果每一个 $\mathcal{T}_{RVPC}$ 上的等价关系 $\mathcal{E}_i$ 都是 symbolic bisimulation，那么 $(\bigcup_{i\in I} \mathcal{E}_i)^*$ 是一个 symbolic bisimulation。

Proof. 令 $\mathcal{E} = (\bigcup_{i\in I} \mathcal{E}_i)$。若因为 $\varphi A_0 \mathcal{E}_1 \varphi A_1 \mathcal{E}_2 \cdots \mathcal{E}_k \varphi A_k$，$(\varphi A_0, \varphi A_k) \in \mathcal{E}$，由于互模拟具有传递性，通过依次证明 $A_0\mathcal{E}A_1, A_1\mathcal{E}A_2 \cdots$，我们可以证明 A_0 和 A_k 互模拟。

我们可以通过 A_0 的 $\varphi\mathcal{E}$-tree，t_{A_0} 递归的构建 A_1 的 $\varphi\mathcal{E}$-tree，t_{A_1}，进而对 $A_0 \rightsquigarrow_{\varphi\mathcal{E}} \xrightarrow{\lambda}_{\varphi} C$ 构造出 $\{A_1 \rightsquigarrow_{\varphi_i\mathcal{E}} \xrightarrow{\lambda}_{\psi_i} C|\mathsf{Th} \vdash (\varphi_i \Rightarrow \psi_i) \vee (\{\varphi_i\}$ 是 φ 的一个划分$)\}$。对于每次递归的 t_{A_0} 的根结点，分以下情况讨论：

- **Case 1 t_{A_0} 的根结点只有一个儿子 A'_0。**
  - **Case 1.1** $A'_0 \in [A_0]_{\varphi\mathcal{E}_1}$。则根据 A'_0 构建 A_1 的 $\varphi\mathcal{E}$-tree。
  - **Case 1.2** $A'_0 \notin [A_0]_{\varphi\mathcal{E}_1}$。根据定义存在划分 $\{\varphi_i\}$ 和集合 $\{A_1 \rightsquigarrow_{\varphi_i\mathcal{E}} \xrightarrow{\lambda}_{\psi_i} \in$

$[A_0']_{\varphi\mathcal{E}}|\mathsf{Th}\vdash\varphi_i\Rightarrow\psi_i\}$。这里我们构建了一个 A_1 的 $\varphi\mathcal{E}_1$-tree, t'_{A_1}。对 t'_{A_1} 的叶子结点 $B\xrightarrow{\lambda}_{\psi_i}B'$ 中的 $B'\in[A_0']_{\varphi\mathcal{E}_1}$，根据 A_0' 的 $\varphi\mathcal{E}$-tree 构建 B' 的 $\varphi\mathcal{E}$-tree。

- **Case 2** A_0 **有** h **个儿子** $A_0^1,\cdots,A_0^h$。
  - **Case 2.1** $\forall j\in[h],A_0^j\mathcal{E}_1A_0$。我们根据 A_0^1 构建 t_{A_1}。
  - **Case 2.2 存在** $A_0^1\notin[A_0]_{\varphi\mathcal{E}_1}$。令 $q=\mathsf{P}_{\varphi\mathcal{E}_1}(A_0\xrightarrow{\coprod_{i\in[h]}p_i\tau}_{\coprod_{i\in I}\psi_i}[A_0^1]_{\varphi\mathcal{E}_1})$，有 $A_1\rightsquigarrow_{\varphi\mathcal{E}_1}\xrightarrow{q}_{\varphi}[A_0^1]_{\varphi\mathcal{E}_1}$。对于 A_1 的 $\varphi\mathcal{E}_1$-tree，t'_{A_1} 的叶子结点 N，$N\xrightarrow{\coprod_{i\in[h]}p_i\tau}_{\coprod_{i\in I}\psi_i}\coprod_{i\in I}N_i'\in[A_0^1]_{\varphi\mathcal{E}_1}$ 中的 N_i'，根据 A_0^1 来构造 N_i' 的 $\varphi\mathcal{E}$-tree。
- **Case 3** t_{A_0} **的根结点** $A_0\xrightarrow{\lambda}_{\varphi}L'\in\mathcal{C}$。根据定义存在划分 $\{\varphi_i\},\mathsf{Th}\vdash\bigvee_{i\in I}\varphi_i\Leftrightarrow\varphi$ 和集合 $\{A_1\rightsquigarrow_{\varphi_i\mathcal{E}_1}\xrightarrow{\lambda}_{\psi_i}\in\mathcal{C}|\mathsf{Th}\vdash\varphi_i\Rightarrow\psi_i\}$，由于 $\mathcal{E}_1\in(\bigcup_{i\in I}\mathcal{E}_i)^*$，我们可以得到存在集合 $\{A_1\rightsquigarrow_{\varphi_i\mathcal{E}}\xrightarrow{\lambda}_{\psi_i}\in\mathcal{C}|\mathsf{Th}\vdash\varphi_i\Rightarrow\psi_i\}$。

Definition 11 如果 $\mathcal{E}_i$ 是一个 $\mathcal{T}_{RVPC}$ 上的 symbolic bisimulation，那么观察等价性 $\approxeq^s_{\mathsf{Th}}$ 定义等价为最大的 symbolic bisimulation。

以下 $\approxeq^s_{\mathsf{Th}}$ 均为 $\mathcal{T}_{RVPC}$ 上的 symbolic bisimulation。

Proposition 5 $\approxeq^s_{\mathsf{Th}}$ 是一个等价关系。

Proposition 6 $\approxeq^s_{\mathsf{Th}}$ 具有同余性。

Proof. $\approxeq^s_{\mathsf{Th}}$ 对于随机选择和非确定性选择的封闭性比较容易证明。

对条件操作运算的封闭性在于，$S\approxeq^s_{\mathsf{Th}}T$ 可以推出 $\varphi S\approxeq^s_{\mathsf{Th}}\varphi T$ 对 $\varphi S\rightsquigarrow_{\psi\approxeq^s_{\mathsf{Th}}}\xrightarrow{\lambda}_{\psi}\mathcal{C}\in\mathcal{T}/\psi\approxeq^s_{\mathsf{Th}}$，有 $\mathsf{Th}\vdash\varphi\psi$，且 $S\rightsquigarrow_{\psi\approxeq^s_{\mathsf{Th}}}\xrightarrow{\lambda}_{\psi}\mathcal{C}\in\mathcal{T}/\psi\approxeq^s_{\mathsf{Th}}$，根据定义，存在 ψ 的划分 $\{\psi_i\}_{i\in I}$ 和集合 $\{T\rightsquigarrow_{\psi_i\approxeq^s_{\mathsf{Th}}}\xrightarrow{\lambda}_{\phi_i}\mathcal{C}|\mathsf{Th}\vdash\psi_i\Rightarrow\phi_i\}$，又因为 $\mathsf{Th}\vdash\varphi\psi$，所以存在 $j\in J\subset I,\mathsf{Th}\vdash\varphi\psi_j$，$\{\varphi T\rightsquigarrow_{\psi_j\approxeq^s_{\mathsf{Th}}}\xrightarrow{\lambda}_{\phi_j}\mathcal{C}|\mathsf{Th}\vdash\psi_j\Rightarrow\phi_j\}$，重新构造 ψ 的划分为 $\{\psi_j\}_{j\in[J-1]}\cup\{\bigvee_{i\in[I]/[J-1]}\psi_i\}$。q-transition 的证明是类似的。

Composition 和 Localization 的证明与 Uniform Approach 中的基本类似，因为 Random VPC 没有对 CCS 的 Composition 和 Localization 做改动。

（由于时间原因这里证明的比较草率，中期检查之后会继续完善）

Proposition 7 若 $S\approxeq^s_{\mathsf{Th}}T$，则对于每一个代换 $\rho,S\rho\approxeq^s_{\mathsf{Th}}T\rho$

Proof. 可以通过 VPC 的 Lemma 3 证明。

第三章　随机传值进程模型的应用

Random VPC 可以用于对云计算协议的形式刻画。在此我们对一种云计算协议：Gossip-Style Membership Protocol 进行建模和模拟实现。

3.1　Gossip Style Membership 协议

3.1.1　Gossip 协议

Gossip 协议分为 push gossip 和 pull gossip 两种。

- **Push Gossip:**
  消息的发送者周期性的随机选择 k 个目标节点发送 Gossip Message。接收到 Gossip Message 的节点可以根据本地时间，周期性的选择 b 个目标节点发送 Gossip Message。在发送过程中，已经拥有 Gossip Message 的节点仍然可以被选为目标节点。
- **Pull Gossip:**
  每个节点周期性的向 k 个目标节点发送 Gossip Query，收到 Gossip Query 的节点若拥有 Gossip Message 的节点会向发送 Query 的节点返回 Gossip Message 的拷贝。

在超过 $\frac{n}{2}$ 的节点拥有 Gossip Message 时，可以证明此时选择 Pull Gossip 会比 Push Gossip 传播的更快 [此处应有引用]。因此在使用 Gossip 协议时，常使用 Push Gossip 与 Pull Gossip 的混合：在消息传播 $\frac{n}{2}$ 节点之前使用 Push Gossip，在消息传播 $\frac{n}{2}$ 之后使用 Pull Gossip。

3.1.2　Membership 协议

在数据中心中会频繁的出现节点的失效，我们会为每一个节点维护一个 Membership List 来记录数据中心中其他节点的运行情况。Membership Protocol 是一个通过网络检测节点失效和传播失效信息的机制，常用的 Membership Protocol 有 Heartbeating Protocol, Gossip-Style Membership Protocol, SWIM Failure Detector Protocol 等。

3.2　Gossip Style Membership 的实现

3.2.1　Push Gossip 协议的实现

由于 Push Gossip 实现难度较小，并且单一的 Push Gossip 协议仍然可以达到 $O(\log N)$ 时间复杂度的消息传播，我们以 Push Gossip 协议的实现作为 Random

VPC 的 demo。其中 Gossip 单节点的通道示意图如图 3–1。以 Gossip 协议作为通信协议的四节点的 Peer-to-Peer Sysyem 的示意图如图 3–2。

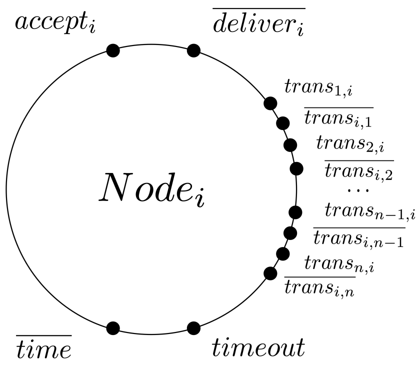

图 3–1 **Gossip** 节点示意图

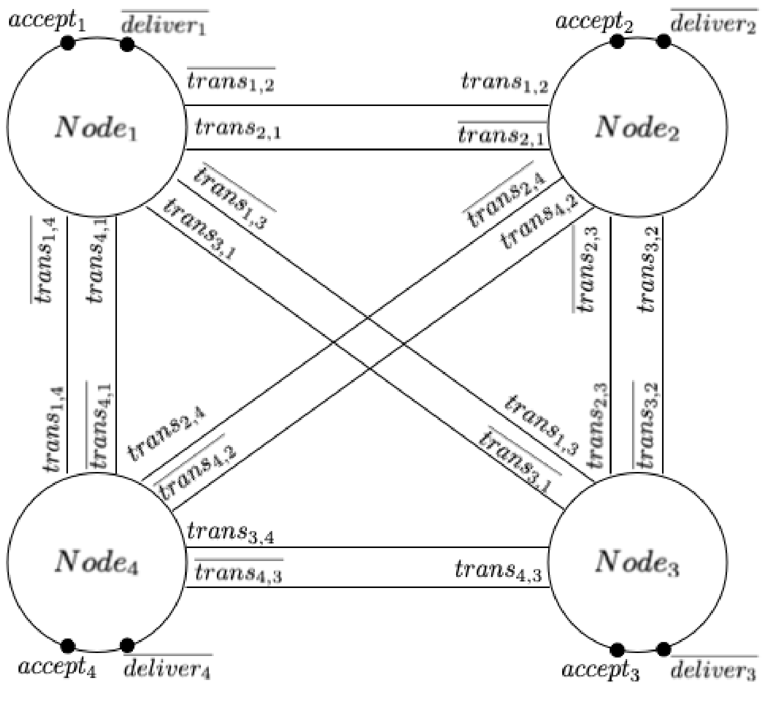

图 3–2 以 **Gossip** 协议作为通信协议的四节点的 **Peer-to-Peer Sysyem** 示意图 (省略 **Timer**)

Gossip 节点 Node 的状态有以下几种，分别对应 Gossip 协议中节点的状态:

表 3–1 节点状态对应[1]

节点状态	Gossip 状态
Node	可接受系统外信息（除此状态外其余均不可接受外界信息）
DeliveringNode	可向系统外传递信息
UnInfectiousNode	未获取 Gossip Message
InfectiousNode	已获取 Gossip Message
GossipingNode	可向系统内部特定 b 个其他节点发送 Gossip Message

一个基于 Gossip 协议的 Peer-to-Peer System 的定义如下。

$$Node_i \stackrel{def}{=} accept_i(x).DeliveringNode_i(x)$$

$$DeliveringNode_i(x) \stackrel{def}{=} \overline{deliver_i}(x).InfectiousNode_i(x)$$

$$InfectiousNode_i(x) \stackrel{def}{=} timeout.(\bigoplus_{perm \in \mathsf{PERM}_i} p_{perm}\tau.GossipingNode_{i,perm}(x))$$

$$GossipingNode_{i,perm}(x) \stackrel{def}{=} \overline{trans_{i,perm_1}}(x).\cdots\overline{trans_{i,perm_b}}(x).\overline{time}.InfectiousNode_i(x)$$

$$UnInfectiousNode_i \stackrel{def}{=} \sum_{j \in \mathsf{N}/\{i\}} trans_{j,i}(x).DeliveringNode_i(x)$$

$$GossipSystem_{\mathsf{N}} \stackrel{def}{=} (Node_1 \mid UnInfectiousNode_2 \mid \cdots \mid UnInfectiousNode_n) \backslash \{trans_{i,j} \mid i \in \mathsf{N} \wedge j \in \mathsf{N} \wedge i \neq j\} \cup \{time, timeout\}$$

在这个 p2p-system 中共有 n 个节点，标号为 N，每一个拥有 gossip message 的节点周期性的向 $\mathsf{k}(\mathsf{k} \le n-1)$ 个其他节点发送 gossip message。其中 PERM_i 为 $\mathsf{N}/\{i\}$ 中任选 k 个元素的全排列。上述定义基于一系列理想条件的假设：

(1) 网络传输可靠。然而现实中的网络传输存在丢包、延迟、比特反转等问题，我们可以通过 ACK 机制来解决 [此处应有引用]。对不可靠网络下的 Gossip 协议，我们在上述理想条件下增加对网络传输的过程的建模，建模过程在 Milner 的 CCS 中有提及 [此处应有引用]。

(2) 节点不会损坏。在现实中节点（也就是主机），是有寿命的，即在运行了一定时间后主机就会宕机，对于多个节点构成的 p2p 系统，存在节点失效的概率只会更高 [此处应有引用：可参照 MTTF]，在后文对 Membership 的建模过程中我们会增加对这个问题的解决方案。

(3) 系统中只存在一个消息的传输。对于单个消息源的多个消息，我们仍然可以看作单个 Gossip Message 一起发送；对于多个消息源，我们可以为每个节点提供消息队列机制，来保存多个消息。在后续的 Membership 建模过程中还会提到。

为了方便建模，我们假设一个节点选择任意其他节点作为 Gossip 的目标节

点的概率是相同的，即 $p_{perm} = \frac{1}{A_{n-1}^{\mathsf{k}}} = \frac{(n-\mathsf{k}-1)!}{(n-1)!}$ 为固定值，当 $\mathsf{k}=2$ 时，$p_{perm} = \frac{1}{(n-1)(n-2)}$。

3.2.2 Push Gossip 协议的等价性

Gossip 协议的目的是为了在系统内对节点进行多播，我们可以定义一个多播的 Specification，同样我们只关注一个消息的多播：

定义 3.1

$$MulticastSpec_{\mathsf{N}} \stackrel{def}{=} accept_1(x).\overline{deliver_1}(x).Multicast_{\mathsf{N},\{1\}}(x)$$
$$Multicast_{\mathsf{N},\mathsf{KNOWN}}(x) \stackrel{def}{=} \tau.Multicasting_{\mathsf{N},\mathsf{KNOWN}}(x)$$
$$Multicasting_{\mathsf{N},\mathsf{KNOWN}}(x) \stackrel{def}{=} \overline{deliver_i}(x).Multicasting_{\mathsf{N},\mathsf{KNOWN}\cup\{i\}}(x),$$
$$i \in \mathsf{N} - \mathsf{KNOWN} \wedge |\mathsf{N}| \neq |\mathsf{KNOWN}|$$

定义 3.2

$$\begin{aligned}
&GossipSystem_{\mathsf{N},(a,b,c)}(x)\\
&= (\overbrace{InfectiousNode(x) \mid \cdots \mid InfectiousNode(x)}^{a}\\
&|\overbrace{DeliveringNode(x) \mid \cdots \mid DeliveringNode(x)}^{b}\\
&|\overbrace{UnInfectiousNode \mid \cdots \mid UnInfectiousNode}^{c})\\
&\backslash\{trans_{i,j} \mid i \in \mathsf{N} \wedge j \in \mathsf{N} \wedge i \neq j\} \cup \{time, timeout\}
\end{aligned}$$

定理 3.1 $GossipSystem_{\mathsf{N}} \approxeq^{s}_{\mathsf{Th}} MulticastSpec_{\mathsf{N}}$，当 $\top \in \mathsf{Th}$ 时成立。

证明 每一个节点能且仅能执行一次 deliver 操作，且 deliver 的顺序不影响功能，所以我们可以规定 $A \xrightarrow{\overline{deliver_i}(t_1)}_{\top} B$ 和 $C \xrightarrow{\overline{deliver_j}(t_2)}_{\top} B$ 是互模拟的当且仅当 $t_1 = t_2$，对下标是否一致不作要求。因此在后文的证明中忽略了下标。

我们可以通过构建等价集，并证明等价集是一个 symbolic bisimulation 关系，来证明 $GossipSystem_{\mathsf{N}} \approxeq^{s}_{\mathsf{Th}} MulticastSpec_{\mathsf{N}}$。

构造等价集

$$\begin{aligned}
\mathcal{S} = \{&(GossipSystem_{\mathsf{N}}, MulticastSpec_{\mathsf{N}}),\\
&(GossipSystem_{\mathsf{N},(n,0,0)}(x), Multicasting_{\mathsf{N},\mathsf{N}}(x))\}\\
&\cup \{(GossipSystem_{\mathsf{N},(a,b,c)}(x), Multicasting_{\mathsf{N},\mathsf{KNOWN}}(x)) \mid |\mathsf{KNOWN}| = a \wedge b \neq 0\}\\
&\cup \{(GossipSystem_{\mathsf{N},(a,b,c)}(x), Multicast_{\mathsf{N},\mathsf{KNOWN}}(x)) \mid |\mathsf{KNOWN}| = a \wedge b = 0\}
\end{aligned}$$

我们来依次证明 $\mathcal{S}$ 中的每一对等价关系为 symbolic bisimulation:

- $GossipSystem_{\mathsf{N},(n,0,0)}(x)$ 和 $Multicasting_{\mathsf{N},\mathsf{N}}(x)$ 的 $\top\mathcal{S}$-tree 分别只有一个根节点，且不能转移至其他等价集，他们的互模拟性是显然的，实际上 $GossipSystem_{\mathsf{N},(n,0,0)}(x) = Multicasting_{\mathsf{N},\mathsf{N}}(x) = 0$。
- 对于 $(GossipSystem_{\mathsf{N},(a,b,c)}(x), Multicasting_{\mathsf{N},\mathsf{KNOWN}}(x)) \mid |\mathsf{KNOWN}| = a \wedge b \neq 0$:
  - **Case 1:** $a < n-1$。 $GossipSystem_{\mathsf{N},(a,b,c)}(x)$ 的 $\top\mathcal{S}$-tree t 如图所示:

    对 t 的叶子结点有 $GossipSystem_{\mathsf{N},(a,n-a,0)}(x) \xrightarrow{\overline{deliver}(x)}_{\top} GossipSystem_{\mathsf{N},(a+1,n-1-a,0)}(x)$。即 $(GossipSystem_{\mathsf{N},(a,b,c)}(x) \rightsquigarrow_{\top\mathcal{S}} \xrightarrow{\overline{deliver}(x)}_{\top} [GossipSystem_{\mathsf{N},(a+1,b',c')}(x)]_{\top\mathcal{S}}, b'+c' = b+c-1$。

    我们可以用 $Multicasting_{\mathsf{N},\mathsf{KNOWN}} \rightsquigarrow_{\top\mathcal{S}} \xrightarrow{\overline{deliver}(x)}_{\top} Multicasting_{\mathsf{N},\mathsf{KNOWN}'}$ 来模拟。
  - **Case 2:** $a = n-1$。 $GossipSystem_{\mathsf{N},(n-1,1,0)}(x)$ 的 $\top\mathcal{S}$-tree t 只有一个根节点:

    $(GossipSystem_{\mathsf{N},(n-1,1,0)}(x) \rightsquigarrow_{\top\mathcal{S}} \xrightarrow{\overline{deliver}(x)}_{\top} [GossipSystem_{\mathsf{N},(n,0,0)}(x)]_{\top\mathcal{S}}$。

    我们可以用 $Multicasting_{\mathsf{N},\mathsf{KNOWN}} \rightsquigarrow_{\top\mathcal{S}} \xrightarrow{\overline{deliver}(x)}_{\top} Multicasting_{\mathsf{N},\mathsf{N}}$ 来模拟。
- 对于 $(GossipSystem_{\mathsf{N},(a,b,c)}(x), Multicast_{\mathsf{N},\mathsf{KNOWN}}(x)) \mid |\mathsf{KNOWN}| = a < n \wedge b = 0$:

  $Multicast_{\mathsf{N},\mathsf{KNOWN}}(x)$ 的 $\top\mathcal{S}$-tree 只有一个根节点，且有 $Multicast_{\mathsf{N},\mathsf{KNOWN}} \rightsquigarrow_{\top\mathcal{S}} \xrightarrow{1}_{\top} [Multicasting_{\mathsf{N},\mathsf{KNOWN}}]_{\top\mathcal{S}}$。

  $GossipSystem_{\mathsf{N},(a,b,c)}(x)$ 的 ϵ-tree t 如图所示:

  $GossipSystem_{\mathsf{N},(a,b,c)}(x)$ 的 $\top\mathcal{S}$-tree t' 为: $GossipSystem_{\mathsf{N},(a,b,c)}(x) \xrightarrow{(\top,p')} GossipSystem_{\mathsf{N},(a,b,c)}(x) \xrightarrow{(\top,p')} \cdots$，其中 $p' = (\frac{A^k_{a-1}}{A^k_{n-1}})^a = (\frac{(a-1)!}{(n-1)!})^a = O(\frac{1}{n}) < 1$。$\mathsf{P}^f(t) = 1 - \lim_{level\to\infty}(p')^{level} = 1$，则 $GossipSystem_{\mathsf{N},(a,b,c)}(x)$ 的 ϵ-tree t 为 regular tree，且 $\mathsf{P}^f(t) = \sum\{p_i \mid L_i \xrightarrow{p_i\tau} [GossipSystem_{\mathsf{N},(a,b',c')}(x)]_{\top\mathcal{S}}\}$，其中 $b' > 0 \wedge b' + c' = b + c$，则

$$GossipSystem_{\mathsf{N},(a,b,c)}(x) \rightsquigarrow_{\top\mathcal{S}} \xrightarrow{1}_{\top} [GossipSystem_{\mathsf{N},(a,b',c')}(x)]_{\top\mathcal{S}}$$

- 余下的一组证明是常规的 VPC 证明。 □

3.2.3 Gossip Style Membership 的实现

3.2.3.1 Gossip Style Membership Protocol

Gossip-Style Membership Protocol 使用 Gossip 协议传递每个节点的 Membership List。每个节点周期性的随机选择 k 个节点发送自己的 Membership List，收到 Membership List 的节点根据其他节点的 Membership List 更新自己的 Membership List。

3.2.3.2 对 GossipSystem 的调整

由于 Gossip-Style Membership 无需与外界的输入输出，并且需要提供判定和修改 Membership List 的函数，我们需要对之前的 Gossip System 进行调整。同时，系统内的节点也会有失效的可能，我们可以用 gailp_{fail} 来定义一个节点失效的概率，同时经过一个内部动作 τ，这个节点就会被修复。另外，由于在系统中的所有节点都是消息源，且 Membership List 在系统中不停的更新，因此也没有了感染者与被感染者的角色区分，也需要对名称进行了修改。

在 Gossip System 节点的基础上，将原本对外暴露的 $accept, deliver$ 通道连接 $MembershipNode$，修改后的节点如图 3–3 所示。

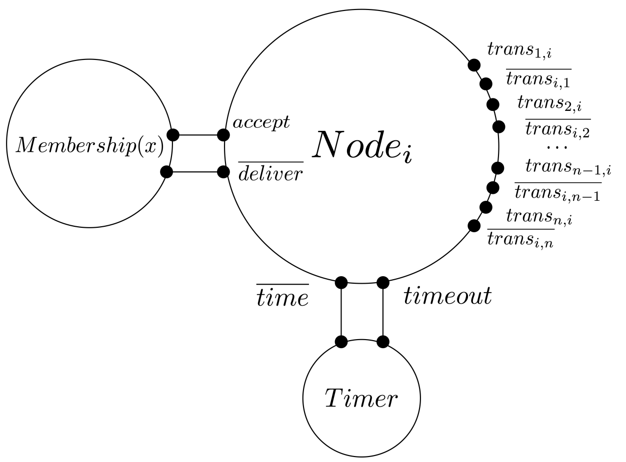

图 3–3 **Gossip-Style Membership Protocol** 节点示意图

$$
\begin{aligned}
FragileNode_i &\overset{def}{=} p_{fail}\tau.BadNode_i \oplus (1-p_{fail})\tau.Node_i \\
BadNode_i &\overset{def}{=} \tau.FragileNode_i \\
Node_i &\overset{def}{=} timeout.accept(x).(\bigoplus_{perm\in \mathsf{PERM}_i} p_{perm}\tau.GossipingNode_{i,perm}(x)) \\
&\quad + \sum_{j\in \mathsf{N}/\{i\}} trans_{j,i}(x).\overline{deliver}(x).Node_i \\
GossipingNode_{i,perm}(x) &\overset{def}{=} \overline{trans_{i,perm_1}}(x).\cdots\overline{trans_{i,perm_b}}(x).\overline{time}.Node_i \\
GossipSystem_{\mathsf{N}} &\overset{def}{=} (Node_1 \mid Node_2 \mid \cdots \mid Node_n) \\
&\quad \backslash\{trans_{i,j} \mid i\in \mathsf{N}\wedge j\in \mathsf{N}\wedge i\neq j\}\cup\{time, timeout, accept, deliver\}
\end{aligned}
$$

其中我们需要定义 Membership 函数，来处理 Membership List 的更新。

Membership List 的内容一般为：因此 $Membership$ 需要有一个本地计

表 3–2 Membership List[1]

Address	HeartBeat	Time(local)
1	10120	66
2	10103	62
3	10098	63

时器 $Timer$、一个心跳计数器 $Counter$，一个本地 $MembershipList(X)$，X 应为一个（$Address, HeartBeat$）的二元组的数组，一个记录本地地址的 $AddrInfo(address)$（地址应在加入网络时由 DNS 分配，此处不考虑它的分配过程）。定义 $x_i[Address]$ 为取 Address 的值的操作子，$x_i[HeartBeat]$ 同理。其中 n

为 MembershipList 的最大容量。

$$Membership \overset{def}{=} addr(address).hbeat(heartbeat).select_1(x_1)\cdots select_n(x_n).$$
$$\overline{accept}(\{Address : address, HeartBeat : heartbeat\}, x_1, \cdots, x_n).Membership$$
$$+\, deliver(x_1, x_2, \cdots, x_n).Hashing(x_1, x_2, \cdots, x_n)$$
$$Hashing(x_1, x_2, \cdots, x_n) \overset{def}{=} (Check(x_1) \mid \cdots \mid Check(x_n) \mid Membership)$$
$$Check(x) \overset{def}{=} addr(address).((address = x[Address])0|\neg(address = x[Address])Find_1(x))$$
$$FindAndUpdate_i(x) \overset{def}{=} select_i(x_i).($$
$$(x_i = \epsilon)(\overline{update_i}(x).0)$$
$$|\neg(x_i = \epsilon)($$
$$(x_i[Address] = x[Address])($$
$$(x_i[HeartBeat] < x[HeartBeat])(\overline{update_i}(x).0)$$
$$|\neg(x_i[HeartBeat] < x[HeartBeat])0)$$
$$|\neg(x_i[Address] = x[Address])FindAndUpdate_{i+1}(x))), i \le n$$
$$FindAndUpdate_i(x) \overset{def}{=} 0, i > n$$
$$MembershipCell_i(addr, hb, t) \overset{def}{=}$$
$$\overline{select_i}(\{Address : addr, HeartBeat : hb\}).MembershipCell_i(addr, hb, t)$$
$$+\, update_i(\{Address : addr', HeartBeat : hb'\}).time(t').MembershipCell_i(addr', hb', t')$$
$$MembershipList \overset{def}{=} (MembershipCell_1(\epsilon) \mid \cdots MembershipCell_n(\epsilon))$$
$$AddrInfo(address) \overset{def}{=} \overline{addr}(address)$$
$$Counter(heartbeat) \overset{def}{=} \overline{hbeat}(\mathrm{s}(heartbeat)).Counter(\mathrm{s}(heartbeat))$$

其中 $\mathrm{s}(x)$ 表示 $\mathrm{s}(x) = x + 1$[此处应有引用]。

3.3 Gossip Style Membership 的仿真模拟

第四章　结论

4.1 总结

4.2 Future Work

附录 A 代码实现

致　谢

感谢那位最先制作出博士学位论文 LaTeX 模板的交大物理系同学！

感谢 William Wang 同学对模板移植做出的巨大贡献！

感谢 @weijianwen 学长一直以来的开发和维护工作！

感谢 @sjtug 以及 @dyweb 对 0.9.5 之后版本的开发和维护工作！

感谢所有为模板贡献过代码的同学们，以及所有测试和使用模板的各位同学！

感谢 LaTeX 和 SJTUThesis，帮我节省了不少时间。

A SAMPLE DOCUMENT FOR LaTeX-BASED SJTU THESIS TEMPLATE

An imperial edict issued in 1896 by Emperor Guangxu, established Nanyang Public School in Shanghai. The normal school, school of foreign studies, middle school and a high school were established. Sheng Xuanhuai, the person responsible for proposing the idea to the emperor, became the first president and is regarded as the founder of the university.

During the 1930s, the university gained a reputation of nurturing top engineers. After the foundation of People's Republic, some faculties were transferred to other universities. A significant amount of its faculty were sent in 1956, by the national government, to Xi'an to help build up Xi'an Jiao Tong University in western China. Afterwards, the school was officially renamed Shanghai Jiao Tong University.

Since the reform and opening up policy in China, SJTU has taken the lead in management reform of institutions for higher education, regaining its vigor and vitality with an unprecedented momentum of growth. SJTU includes five beautiful campuses, Xuhui, Minhang, Luwan Qibao, and Fahua, taking up an area of about 3,225,833 m2. A number of disciplines have been advancing towards the top echelon internationally, and a batch of burgeoning branches of learning have taken an important position domestically.

Today SJTU has 31 schools (departments), 63 undergraduate programs, 250 masters-degree programs, 203 Ph.D. programs, 28 post-doctorate programs, and 11 state key laboratories and national engineering research centers.

SJTU boasts a large number of famous scientists and professors, including 35 academics of the Academy of Sciences and Academy of Engineering, 95 accredited professors and chair professors of the "Cheung Kong Scholars Program" and more than 2,000 professors and associate professors.

Its total enrollment of students amounts to 35,929, of which 1,564 are international students. There are 16,802 undergraduates, and 17,563 masters and Ph.D. candidates. After more than a century of operation, Jiao Tong University has inherited the old tradition of "high starting points, solid foundation, strict requirements and extensive practice." Students from SJTU have won top prizes in various competitions, including ACM International Collegiate Programming Contest, International Mathematical Contest in Modeling and Electronics Design Contests. Famous alumni include Jiang Zemin, Lu Dingyi, Ding Guangen, Wang Daohan, Qian Xuesen, Wu Wenjun, Zou Taofen, Mao Yisheng, Cai Er, Huang Yanpei, Shao Lizi, Wang An and many more. More than 200 of the academics of the Chinese Academy of Sciences and Chinese Academy of Engineering are alumni of Jiao Tong University.