中国科学院研究生院课程讲稿

§2 程序与系统模型

程序的正确从广义上说就是包含该程序的系统的运行过程满足给定的性质。要说明程序的正确，我们需要描述系统的这种运行过程，即系统的运行模型，简称为系统模型。本章介绍系统模型。

§2.1 隐式迁移模型

隐式迁移模型通常是是一阶逻辑的扩充。 给定常元和函数符号集合 F ，谓词符号集合 P 。记 $B=(F,P)$ 上一阶逻辑公式集合为 WFF_B ， B 上的项的集合为 T_B 。记 WFF_B 中不带量词的公式集合为 QFF_B 。 系统的隐式迁移模型通常是在这样的一阶逻辑上增加符号进行赋值操作和流程控制，其中 WFF_B 中的公式可作为流程控制的条件， T_B 中的项可作为赋值语句中值的表示。

§2.1.1 结构化循环语句模型

结构化循环语句模型是一阶逻辑的扩充。 重要模型要素有赋值、顺序复合、条件语句和循环语句。 给定 $B=(F,P)$ 。我们增加以下集合的符号：

$$\{:=, ;, if, then, else, fi, while, do, od, \epsilon\}$$

Definition 2.1 给定 $B=(F,P)$ 和变量集合 V 。一个 (B,V) 上的结构化循环语句模型是一个字符串，其集合 S 定义如下：

$$\begin{array}{lll} S & ::= & \epsilon \mid T;\epsilon \\ \hline T & ::= & x:=t \mid \\ & & T;T \mid \\ & & \mathit{if}\ e\ \mathit{then}\ T\ \mathit{else}\ T\ \mathit{fi} \mid \\ & & \mathit{while}\ e\ \mathit{do}\ T\ \mathit{od} \end{array}$$

其中 $x\in V$ 为变元， $t\in T_B$ 为 B 上的项且 $Var(t)\subseteq V$ ， $e\in QFF_B$ 为一个不带量词的公式且 $Var(e)\subseteq V$ 。

给定 $B=(F,P)$ 和变量集合 V 。 (B,V) 上的结构化循环语句模型的集合记为 $\mathcal{L}_{\bigcirc}^{(B,V)}$ 。为书写方便，在不引起歧义的情况下，模型结尾的 ϵ 可以省略不写。

系统状态

模型的系统状态有两个部分，即模型运行的剩余部分和变量状态。 变量状态空间为 V 中变量取值的组合 $\Sigma=\{\sigma|\sigma(x)\in D, x\in V\}$ 。 系统的状态空间为模型和 V 中变量取值的组合 $\mathcal{L}_{\bigcirc}^{(B,V)}\times\Sigma$ 。

状态迁移关系

一个模型的运行取决于 B 的解释。 给定 B 的一个解释 I 。模型系统状态的迁移关系 $\rightarrow$ 为 $(\mathcal{L}_{\bigcirc}^{(B,V)}\times\Sigma)^2$ 的一个子集。 $(S_0,\sigma_0)\rightarrow(S_1,\sigma_1)$ 当且仅当以下一项成立。

S_0	条件	S_1	σ_1
$x := t; S$		S	$\sigma_0[x/I(t)(\sigma_0)]$
if (e) then S else S' fi; S''	$\sigma_0 \models_I e$	$S; S''$	σ_0
if (e) then S else S' fi; S''	$\sigma_0 \not\models_I e$	$S'; S''$	σ_0
while e do S od; S'	$\sigma_0 \models_I e$	S; while (e) do S od; S'	σ_0
while e do S od; S'	$\sigma_0 \not\models_I e$	S'	σ_0
ϵ		ϵ	σ_0

以上的最后一项说明模型运行终止后状态保持不变。

运行

用 $[(S_i,\sigma_i)]_{i\geq 0}$ 表示无穷系列

$$(S_0,\sigma_0)(S_1,\sigma_)(S_2,\sigma_2)......$$

模型 T 的一个运行是状态集 $\mathcal{L}_{\bigcirc}^{(B,V)} \times \Sigma$ 上的一个无穷序列 $[(S_i,\sigma_i)]_{i\geq 0}$ 满足 $S_0 = T$ 且对任意 $i \geq 0$,

$$(S_i,\sigma_i) \rightarrow (S_{i+1},\sigma_{i+1})$$

模型 T 在初始变量状态为 σ_0 时运行可到达的状态的集合为 $\{(T',\sigma)|(T,\sigma_0)\xrightarrow{*}(T',\sigma)\}$ 。

终止

模型 T 在初始变量状态为 σ 时的运行终止，当且仅当存在 σ' 使得 $(T,\sigma)\xrightarrow{*}(\epsilon,\sigma')$ 。

模型性质

模型的基本性质包括模型运行的终止性以及模型的终止状态和初始状态的关系。设 φ 和 ψ 是谓词。一些基本的性质描述如下：

（1）给定初始时的变量状态满足的条件 φ ，给定 ψ 。要求如果终止，则模型的终止时的变量状态满足 ψ 。这个要求称为模型对于 φ 和 ψ 的部分正确性。 φ 称为模型的前断言， ψ 称为模型的后断言。

（2）给定初始时的变量状态满足的条件 φ ，给定 ψ 。要求模型能够终止并且模型的终止时的变量状态满足 ψ 。这个要求称为模型对于 φ 和 ψ 的完全正确性。

（3）完全正确性包括了模型的终止性，即给定初始时的变量状态满足的条件 φ ，要求模型能够终止。

模型性质的形式定义

设谓词 φ 为模型 T 的初始状态满足的条件。 T 对于谓词 φ 的终止性定义如下：

$$\text{终止性:} \quad \varphi(\sigma) \rightarrow ((T,\sigma)\xrightarrow{*}(\epsilon,\sigma'))$$

设谓词 φ 为模型 T 的初始状态满足的条件，谓词 ψ 为对模型 T 的终止状态的要求。 T 对于谓词 φ 和谓词 ψ 的部分正确性和完全正确性定义如下：

$$\begin{array}{ll} \text{部分正确性:} & \varphi(\sigma) \rightarrow ((T,\sigma)\xrightarrow{*}(\epsilon,\sigma') \rightarrow \psi(\sigma')) \\ \text{完全正确性:} & \varphi(\sigma) \rightarrow ((T,\sigma)\xrightarrow{*}(\epsilon,\sigma') \wedge \psi(\sigma')) \end{array}$$

相应地可以定义以公式为前后断言的模型正确性。 对于公式 φ 和 ψ ，可以把公式通过 I 解释成为谓词，则可以有类似的性质描述。 T 对于公式 φ 的终止性即 T 对于谓词 $I(\varphi)$ 的终止性即

$$\text{终止性:}\quad I(\varphi)(\sigma) \rightarrow ((T,\sigma)\xrightarrow{*}(\epsilon,\sigma'))$$

T 对于公式 φ 和公式 ψ 的部分正确性和完全正确性定义如下：

$$\begin{array}{ll}\text{部分正确性:} & I(\varphi)(\sigma) \rightarrow ((T,\sigma)\xrightarrow{*}(\epsilon,\sigma') \rightarrow I(\psi)(\sigma')) \\ \text{完全正确性:} & I(\varphi)(\sigma) \rightarrow ((T,\sigma)\xrightarrow{*}(\epsilon,\sigma') \wedge I(\psi)(\sigma'))\end{array}$$

T 对于公式 φ 和公式 ψ 的部分正确性和完全正确性分别记作 $\models_I \{\varphi\}T\{\psi\}$ 和 $\models_I [\varphi]T[\psi]$ 。

§2.1.2　流程图模型

流程图模型 是一阶逻辑的扩充。 流程图模型主要有两种语句。一种是赋值语句。一种是条件语句。这些语句称为指令。 在一阶逻辑的基础上，我们增加以下两个集合的符号：

$$\begin{array}{ll}\text{辅助符号集 AX} & \{:=, :, goto, if, else\} \\ \text{标号符号集 LB} & \{beg, end, l_1, l_2, ...,\}\end{array}$$

给定 $B=(F,P)$ 和变量集合 V 。定义 (B,V) 上的流程图模型的两种指令如下：

l_1: $(x_1,...,x_n) := (t_1,...,t_n)$ goto l_2
l_1: if (e) goto l_2 else goto l_3

其中 $l_1,l_2,l_3 \in LB$ 为标号， $l_1 \neq end$, $t_1,...,t_n \in T_B$ 且 $\bigcup_{i=1}^{n} Var(t_i) \subseteq V$, $e \in QFF_B$ 且 $Var(e) \subseteq V$, $x_1,...,x_n \in V$ 且对 $1 \leq i < j \leq n$, $x_i \neq x_j$ 。 冒号左边的标号称为被定义标号。冒号右边的句子称为标号定义。

Definition 2.2 一个 (B,V) 上的结流程图模型 T 为满足以下条件的指令集合。

标号 beg 必须有定义
标号 end 不被定义
出现在标号定义中的除 end 外的标号必须有定义
每个标号最多被定义一次

给定 $B=(F,P)$ 和变量集合 V 。 (B,V) 上流程图模型的集合记为 $\mathcal{L}_{\hookleftarrow}^{(B,V)}$ 。记 $\mathcal{L}_{\hookleftarrow}^{B}$ 为任给变量集合 V 的 $\mathcal{L}_{\hookleftarrow}^{(B,V)}$ 的并集。

系统状态

流程图模型的系统状态有两个部分：即标号和变量状态。标号可以理解为模型或系统的控制状态。 变量状态空间为 V 中变量取值的组合 $\Sigma = \{\sigma|\sigma(x) \in D, x \in V\}$ 。 系统的状态空间为标号和 V 中变量取值的组合 $LB \times \Sigma$ 。

状态迁移关系

给定 B 的一个解释 I 。给定指令 t 。 $(l_i,\sigma_i)\xrightarrow{t}(l_{i+1},\sigma_{i+1})$ 当且仅当以下一种情况成立。

t 为 l_i: $(v_1,...,v_n) := (t_1,...,t_n)$ goto l_{i+1} , 且 $\sigma_{i+1} = \sigma_i[v_1/I(t_1)(\sigma_i)]\cdots[v_n/I(t_n)(\sigma_i)]$
t 为 l_i: (if (e) goto l' else goto l'' , $\sigma_{i+1} = \sigma_i$ 且 $\sigma_i \models_I e$ 则 $l_{i+1} = l'$ ，否则 $l_{i+1} = l''$

给定流程图模型 T 和 B 的一个解释 I 。我们定义流程图模型系统状态的转换关系 $\xrightarrow{T}$ 如下。 $(l_i,\sigma_i)\xrightarrow{T}(l_{i+1},\sigma_{i+1})$ 当且仅当以下一种情况成立。

存在指令 $t \in T$ 使得 $(l_i, \sigma_i) \xrightarrow{t} (l_{i+1}, \sigma_{i+1})$ 或
不存在定义标号 l_i 的指令且 $(l_i, \sigma_i) = (l_{i+1}, \sigma_{i+1})$ 。

在不引起混淆的时候，我们将 $\xrightarrow{T}$ 中的 T 省略写成 $\rightarrow$ 。

运行

用 $[(l_i, \sigma_i)]_{i \geq 0}$ 表示无穷系列

$$(l_0, \sigma_0)(l_1, \sigma_1)(l_2, \sigma_2)......$$

流程图模型 T 的一个运行是状态集 $LB \times \Sigma$ 上的一个无穷序列 $[(l_i, \sigma_i)]_{i \geq 0}$ 满足 $l_0 = beg$ 且对任意 $i \geq 0$, $(l_i, \sigma_i) \rightarrow (l_{i+1}, \sigma_{i+1})$ 。

流程图模型 T 在初始变量状态为 σ_0 时运行可到达的状态的集合为 $\{(l, \sigma) | (beg, \sigma_0) \xrightarrow{*} (l, \sigma)\}$ 。

终止

模型在初始变量状态为 σ 时的运行终止，当且仅当存在 σ' 使得 $(beg, \sigma) \xrightarrow{*} (end, \sigma')$ 。

模型性质

设谓词 φ 为模型 T 的初始状态满足的条件。 T 对于 φ 的终止性定义如下：

$$\text{终止性：} \quad \varphi(\sigma) \rightarrow ((beg, \sigma) \xrightarrow{*} (end, \sigma'))$$

设谓词 φ 为模型 T 的初始状态满足的条件，谓词 ψ 为对模型 T 的终止状态的要求。 T 对于 φ 和 ψ 的部分正确性和完全正确性定义如下：

$$\text{部分正确性：} \quad \varphi(\sigma) \rightarrow ((beg, \sigma) \xrightarrow{*} (end, \sigma') \rightarrow \psi(\sigma'))$$
$$\text{完全正确性：} \quad \varphi(\sigma) \rightarrow ((beg, \sigma) \xrightarrow{*} (end, \sigma') \wedge \psi(\sigma'))$$

相应地可以定义以公式为前后断言的模型的正确性。

§2.1.3 卫式迁移模型

卫式迁移模型是一阶逻辑的扩充。 卫式迁移模型所描述的迁移关系主要有两个要素：卫式和赋值。除此之外，系统还包含初始条件的描述。 在一阶逻辑的基础上，我们增加符号

$$\{\longrightarrow, :=\}$$

给定 $B = (F, P)$ 和变量集合 V 。定义 (B, V) 上的迁移如下：

$$p \longrightarrow (v_1, v_2, ..., v_n) := (e_1, e_2, ..., e_n)$$

其中 $p \in QFF_B$ 为一个不带量词的公式且 $Var(p) \subseteq V$ ， $v_1, v_2, ..., v_n \in V$ 为变元， $e_1, e_2, ..., e_n \in T_B$ 为 B 上的项且 $\bigcup_{i=1}^{n} Var(e_i) \subseteq V$ 。一个迁移表示系统的一个原子动作。

Definition 2.3 一个 (B, V) 上的卫式迁移模型是一个二元组

$$(T, \Theta)$$

其中 T 为 (B, V) 上迁移的有限集合， $\Theta \in QFF_B$ 为初始条件，是一个不带量词的公式且 $Var(\Theta) \subseteq V$ 。

B 的解释确定了迁移系统描述中常元符号、函数符号和谓词符号的含义。 变量的赋值 σ 代表了系统运行中变量在某一时刻的状态。因此也称 σ 为状态。状态的集合为 Σ 。 以下假设 B 的解释 $I = (D, I_0)$ 为已经给定。

状态空间

系统的状态空间为 V 中变量取值的组合 $\Sigma = \{\sigma | \sigma(x) \in D, x \in V\}$ 。

状态迁移关系

迁移 $\varphi \longrightarrow (v_1, v_2, ..., v_n) := (e_1, e_2, ..., e_n)$ 在状态 σ 可执行，当且仅当 $\sigma \models_I \varphi$ 。 设 t 是 $\varphi \longrightarrow (v_1, v_2, ..., v_n) := (e_1, e_2, ..., e_n)$ 为一个迁移。用 $\sigma \xrightarrow{t} \sigma'$ 表示 t 在状态 σ 可执行且 $\sigma' = \sigma[v_1/I(e_1)(\sigma)] \cdots [v_n/I(e_n)(\sigma)]$ 。

用 $\sigma \rightarrow \sigma'$ 表示

在状态 σ	存在可执行迁移 t 且 $\sigma \xrightarrow{t} \sigma'$
在状态 σ	不存在可执行迁移且 $\sigma' = \sigma$

运行

用 $[\sigma_i]_{i \geq 0}$ 表示无穷系列

$$\sigma_0 \sigma_1 \sigma_2 \sigma_3 \ldots\ldots$$

卫式迁移模型 (T, Θ) 上的一个运行是状态集 Σ 上的一个无穷序列 $[\sigma_i]_{i \geq 0}$ 满足 $\sigma_0 \models_I \Theta$ 且对任意 $i \geq 0$,

$$\sigma_i \rightarrow \sigma_{i+1}$$

可达状态

用 $\sigma \rightarrow \sigma'$ 表示存在 $t \in T$ 使得 $\sigma \xrightarrow{t} \sigma'$ 。状态集合 A 的可达状态集合 $rh(A)$ 为 $\{\sigma' | \sigma \xrightarrow{*} \sigma', \sigma \in A\}$ 。系统的可达状态集合 $rh(\Theta)$ 为 $\{\sigma' | \sigma \xrightarrow{*} \sigma', \sigma \models_I \Theta\}$ 。

安全性质

给定一个状态公式 φ 。系统 M 满足安全性质 φ ，记作 $M \models \Box\varphi$ ，当且仅当系统的每个运行中的每个状态都满足 φ 。即 $\forall \sigma \in rh(\Theta). \sigma \models_I \varphi$ 。

响应性质

给定一个状态公式 φ 。系统 M 满足简单响应性质 φ ，记作 $M \models \Diamond\varphi$ ，当且仅当系统的每个运行中都有状态满足 φ 。

§2.1.4 谓词迁移模型

谓词迁移模型是一阶逻辑的扩充，表示为一阶逻辑公式的二元组。

给定 $B = (F, P)$ 和变量集合 $V = \{v_1, ..., v_n\}$ 。 定义 $V' = \{v' | v \in V\}$ 。

Definition 2.4 一个 (B, V) 上的谓词迁移模型是一个二元组

$$(\rho, \Theta)$$

其中 $\rho \in QFF_B$ 为迁移关系，一个不带量词的公式且 $Var(\rho) \subseteq V \cup V'$ ， $\Theta \in QFF_B$ 为初始条件，是一个不带量词的公式且 $Var(\Theta) \subseteq V$ 。

B 的解释确定了迁移系统描述中常元符号、函数符号和谓词符号的含义。 变量的赋值 σ 代表了系统运行中变量在某一时刻的状态。因此也称 σ 为状态。状态的集合为 Σ 。 以下假设 B 的解释 $I = (D, I_0)$ 为已经给定。

状态空间

系统的状态空间为 V 中变量取值的组合 $\Sigma = \{\sigma|\sigma(x) \in D, x \in V\}$ 。

状态迁移关系

迁移 ρ 在状态 σ 可执行，当且仅当存在 $a_1, ..., a_n$ 使得 $\sigma[v_1'/a_1]\cdots[v_n'/a_n] \models_I \rho$ 。 用 $\sigma \rightarrow \sigma'$ 表示 存在 $a_1, ..., a_n$ 满足 $\sigma[v_1'/a_1]\cdots[v_n'/a_n] \models_I \rho$ 且 $\sigma' = \sigma[v_1/a_1]\cdots[v_n/a_n]$ ，或 ρ 在状态 σ 不可执行且 $\sigma' = \sigma$ 。

运行

用 $[\sigma_i]_{i\geq 0}$ 表示无穷系列

$$\sigma_0\sigma_1\sigma_2\sigma_3......$$

谓词迁移模型 (ρ, Θ) 上的一个运行是状态集 Σ 上的一个无穷序列 $[\sigma_i]_{i\geq 0}$ 满足 $\sigma_0 \models_I \Theta$ 且对任意 $i \geq 0$,

$$\sigma_i \rightarrow \sigma_{i+1}$$

可达状态

用 $\sigma \rightarrow \sigma'$ 表示存在 $t \in T$ 使得 $\sigma \xrightarrow{t} \sigma'$ 。状态集合 A 的可达状态集合 $rh(A)$ 为 $\{\sigma'|\sigma \xrightarrow{*} \sigma', \sigma \in A\}$ 。系统的可达状态集合 $rh(\Theta)$ 为 $\{\sigma'|\sigma \xrightarrow{*} \sigma', \sigma \models_I \Theta\}$ 。

安全性质和响应性质

类似于条件迁移模型，我们可以讨论安全性质和响应性质。

§2.1.5 不同模型的关系

以上介绍了结构化循环语句模型、流程图模型、卫式迁移模型和逻辑迁移模型。有些模型可以相互转换。

给定结构化循环语句模型 S 。我们可以构造一个相应的流程图模型。 我们用 S 构造带标号的程序。这个构造记为 f ，定义如下

S	$f(S)$
ϵ	end
$x := t$	$x := t$
$T_1; T_2$	$f(T_1); \quad l_{T_2} : f(T_2)$
if e then T_1 else T_2 fi	if e then $l_{T_1} : f(T_1)$ else $l_{T_2} : f(T_2)$ fi
while e do T od	while e do $l_T : f(T)$ od

我们用带标号的程序 S 构造流程图模型。这个构造记为 g ，定义如下（其中 $l(S)$ 为 S 的标号部分）：

S	$g(S)$	
end	$\{\}$	
l_T	$\{\}$	
l_T : x:=t; T_1	$\{l_T : x := t$ goto $l(T_1);\}$	$\cup\ g(T_1)$
l_T : if e then T_1 else T_2 fi; T_3	$\{l_T :$ if (e) goto $l(T_1)$ else goto $l(T_2);\}$	$\cup\ g(T_1; l(T_3)) \cup g(T_2; l(T_3)) \cup g(T_3)$
l_T : while e do T_1 od; T_2	$\{l_T :$ if (e) goto $l(T_1)$ else goto $l(T_2);\}$	$\cup\ g(T_1; l_T) \cup g(T_2)$

给定结构化循环语句模型 S 。则 $g(beg : f(S))$ 为相应的流程图模型。

§2.2 显式状态迁移模型

在隐式迁移模型的描述中，系统运行的状态空间以及系统的运行由解释 I 所决定。明确的状态空间以及状态之间迁移关系的描述有助于简化系统性质的检验。

§2.2.1 Kripke 模型 (KS)

以条件迁移模型或谓词迁移模型为出发点，状态空间即是 Σ 。除了状态空间之外，描述系统运行的基本要素还有状态之间的迁移关系，以及系统的初始状态。

Definition 2.5 一个 *Kripke* 模型是一个三元组

$$\langle S, \Delta, I\rangle$$

其中 S 为状态集合， $\Delta \subseteq S \times S$ 为 S 上的完全迁移关系， $I \subseteq S$ 为初始状态集。

路径和运行

用 $s \rightarrow s'$ 表示存在从 s 到 s' 的迁移，即 $(s, s') \in \Delta$ 。用 $[s_i]_{i=0}^{n}$ 表示有穷序列

$$s_0 s_1 s_2 \ldots s_n$$

Kripke 模型上的有穷路径是 S 上的有穷序列 $[s_i]_{i=0}^{n}$ 满足对任意 $0 \leq i \leq n-1$, $s_i \rightarrow s_{i+1}$ 。
Kripke 模型上的无穷路径是 S 上的无穷序列 $[s_i]_{i\geq 0}$ 满足对任意 $i \geq 0$, $s_i \rightarrow s_{i+1}$ 。
Kripke 模型上的运行是 Kripke 模型上的无穷路径 $[s_i]_{i\geq 0}$ 满足 $s_0 \in I$ 。

可达状态

用 $\xrightarrow{*}$ 表示 $\rightarrow$ 的传递自反闭包。状态集合 A 的可达状态集合 $rh(A)$ 为 $\{\sigma' | s \xrightarrow{*} s', s \in A\}$ 。系统的可达状态集合为 $rh(I)$ 。

安全性质

系统的安全性质可用状态集合表示。状态集合 A 安全，当且仅当系统的每个运行中的每个状态都在 A 中。即 $rh(I) \subseteq A$ 。

响应性质

系统的响应性质可用状态集合表示。状态集合 A 是响应的，当且仅当系统的每个运行中有状态在 A 中。

可达性质

安全性质的对偶性质是可达性质。状态集合 A 可达，当且仅当系统的某个运行中的某个状态在 A 中。即 $rh(I) \cap A \neq \emptyset$ 。

§2.2.2 标号 KS

由于 KS 对于系统的描述过于简化， KS 作为软件模型，用状态集表示性质显得很复杂。如果表示性质的集合很大，则其检查就很复杂。为了表达和计算的简单，可用命题表示性质。为了能够用命题表示性质，我们必须将结构中的状态和什么公式在哪些状态上成立联系起来。为了定义上的简单，我们只考虑在每个状态上某些简单命题是否成立。

Definition 2.6 给定一个原子命题集合 AP 。一个 AP 上的标号 KS 是一个四元组

$$\langle S, \Delta, I, L\rangle$$

其中 $\langle S, \Delta, I\rangle$ 为 *Kripke* 模型， $L : S \rightarrow 2^{AP}$ 为状态集合到 AP 的幂集的映射。

在这样的一个结构中， $L(s)$ 表示在 s 上所有 s 成立的原子命题的集合。原子命题 p 在 s 上成立当且仅当 $p \in L(s)$ ，其它命题逻辑公式的成立与否遵从命题逻辑的解释。

记原子命题集合 AP 上所有命题公式的集合为 $\mathcal{L}_{AP}$ 。我们可以用命题逻辑的公式表示性质。给定一个 AP 上的标号 Kripke 模型 $\langle S, \Delta, I, L\rangle$ ，一个状态 $s \in S$ 满足性质 $\varphi \in \mathcal{L}_{AP}$ 记作 $M, s \models \varphi$ ，其定义如下： $M, s \models \varphi$ 成立当且仅当：

$C, s \models p$	若 $p \in AP$ 且 $p \in L(s)$ 。
$M, s \models \neg\psi$	若 $M, s \not\models \psi$ 。
$M, s \models \psi_0 \vee \psi_1$	若 $M, s \models \psi_0$ 或 $M, s \models \psi_1$ 。
$M, s \models \psi_0 \wedge \psi_1$	若 $M, s \models \psi_0$ 且 $M, s \models \psi_1$ 。
$M, s \models \psi_0 \rightarrow \psi_1$	若 $M, s \models \psi_0$ 则 $M, s \models \psi_1$ 。
$M, s \models \psi_0 \leftrightarrow \psi_1$	若 $M, s \models \psi_0$ 当且仅当 $M, s \models \psi_1$ 。

设 $m = \{x_1, ..., x_k\} \subseteq AP$ 且 $AP \setminus m = \{y_1, ..., y_l\}$ 。定义 $m \models \varphi$ 当且仅当 $(\varphi^{1,\dots,1}_{x_1,\dots,x_k})^{0,\dots,0}_{y_1,\dots,y_l}$ 的值为 1 。则 $M, s \models \varphi$ 当且仅当 $L(s) \models \varphi$ 。

语言

M 的语言记为 $\mathcal{L}(M) \subseteq (2^{AP})^\omega$ ，定义如下： $u = u_0 u_1 \cdots \in \mathcal{L}(M)$ 当且仅当 M 有一个运行 $r = r_0 r_1 \cdots$ 且对所有 i ， $u_i = L(r_i)$ 。

公式与状态集合

一个公式对应于一个状态集合。即 φ 对应于状态集合 $\{s | M, s \models \varphi\}$ 。用 $[[\varphi]]$ 表示状态集合 $\{s | M, s \models \varphi\}$ 。

安全性质

系统的安全性质可用公式表示。系统满足安全性质 φ，当且仅当系统的每个运行中的每个状态都满足 φ 。即 $\forall s \in rh(I).(M, s \models \varphi)$ ，即 $\forall s \in rh(I).(L(s) \models \varphi)$ ，即 $rh(I) \subseteq [[\varphi]]$ 。

响应性质

系统的响应性质可用公式表示。系统满足响应性质 φ，当且仅当系统的每个运行中都有状态都满足 φ 。

§2.2.3 扩展标号 KS

标号 KS 只考虑在每个状态上某些简单命题是否成立，我们可以对其进行扩展，在每个状态上标上某个公式是否成立。

Definition 2.7 *给定一个原子命题集合 AP 。一个 AP 上的扩展标号 KS 是一个四元组*

$$\langle S, \Delta, I, L\rangle$$

其中 $\langle S, \Delta, I\rangle$ 为 Kripke 模型， $L : S \rightarrow \mathcal{L}_{AP}$ 为状态集合到命题逻辑公式集合的映射。

给定一个 AP 上的扩展标号 Kripke 模型 $\langle S, \Delta, I, L\rangle$ ，一个状态 $s \in S$ 满足性质 $\varphi \in \mathcal{L}_{AP}$ 记作 $M, s \models \varphi$ 当且仅当 $L(s) \rightarrow \varphi$ 。

语言

M 的语言记为 $\mathcal{L}(M) \subseteq (2^{AP})^\omega$ ，定义如下： $u = u_0u_1 \cdots \in \mathcal{L}(M)$ 当且仅当 M 有一个运行 $r = r_0r_1 \cdots$ 且对所有 i ， $u_i \models L(r_i)$ 。

安全性质

系统满足安全性质 $\varphi \in \mathcal{L}_{AP}$ ，当且仅当系统的每个运行中的每个状态都满足 φ ，即

$$\forall s \in rh(I).(L(s) \to \varphi) \text{ 。}$$

可达性质

系统满足可达性质 $\varphi \in \mathcal{L}_{AP}$ ，当且仅当存在可以满足 φ 的可达状态，即

$$\exists s \in rh(I).(L(s) \wedge \varphi \text{ 是可满足的}) \text{ 。}$$

§2.2.4 公平标号 Kripke 模型

对于 Kripke 模型而言，只要满足状态迁移关系的无穷状态序列就是系统的一个运行。有时候为了更精确地描述系统的运行，我们需要对运行做一定限制以排除一些满足状态转换关系但不合理的无穷状态序列。 若要求能够排除这样的无穷状态序列，我们需要在结构中加入新的成分，即设置公平运行的条件，定义公平标号 Kripke 模型。

Definition 2.8 一个公平标号 *Kripke* 模型是一个五元组

$$\langle S, \Delta, I, L, F \rangle$$

其中 $\langle S, \Delta, I, L \rangle$ 是一个 AP 上的标号 *Kripke* 模型， $F \subseteq 2^S$ 为公平性约束的集合。

运行

公平标号 Kripke 模型 $\langle S, \Delta, I, L, F \rangle$ 上的一个运行即 Kripke 模型 $\langle S, \Delta, I \rangle$ 上的一个运行。 定义 $inf(\pi)$ 为无限多次出现在 π 中的状态的集合。一个公平标号 Kripke 模型上的运行 $\pi = [s_i]_{i \geq 0}$ 是公平的，当且仅当对所有 $f \in F$ ，

$$inf(\pi) \cap f \neq \emptyset$$

语言

M 的语言记为 $\mathcal{L}(M) \subseteq (2^{AP})^\omega$ ，定义如下： $u = u_0u_1 \cdots \in \mathcal{L}(M)$ 当且仅当 M 有一个公平运行 $r = r_0r_1 \cdots$ 且对所有 i ， $u_i \models L(s_i)$ 。

语言非空性质

给定 $M = \langle S, \Delta, I, L, F \rangle$ 。 $\mathcal{L}(M) \neq \emptyset$ 当且仅当 M 有一个公平运行 当且仅当有向图 (S, Δ) 存在由 I 可达的强连通分量 C 且对所有 $f \in F$ ， $C \cap f \neq \emptyset$ 。

§2.2.5 不同模型的关系

以上介绍了 Kripke 模型、标号 Kripke 模型、扩展标号 Kripke 模型和公平标号 Kripke 模型。部分模型之间可以相互转换。部分模型可以由隐式迁移系统生成。

标号 KS 与扩展标号 KS

给定 AP 上的扩展标号 KS 模型 $M = \langle S, \Delta, I, L\rangle$ 。我们可以构造一个相应的标号 KS 模型。 定义 $M' = \langle S', \Delta', I', L'\rangle$ 如下

$$\boxed{\begin{array}{ll} S' = & \{(s,m)|s \in S, m \models L(s)\} \\ \Delta' = & \{((s,m),(s',m'))|\{(s,s') \in \Delta\} \\ I' = & \{(s,m)|\{s \in I\} \\ L': & L((s,m)) = m \end{array}}$$

则 $\mathcal{L}(M') = \mathcal{L}(M)$ 。

谓词迁移模型与标号 KS

KS 可由有穷状态的谓词迁移模型或卫式迁移模型生成。若 KS 由谓词迁移模型生成， KS 的状态空间的大小由该模型的变量及其取值范围决定。给定 (B,V) 上的谓词迁移模型 (ρ,Θ) 。给定 B 上的解释 $\mathcal{I}$ 。设 $V = \{v_1,...,v_n\}$ 且令 $|v|$ 为 v 可能取值的个数。

状态空间的大小为 $\lvert v_1\rvert \cdot \lvert v_2\rvert \cdots \lvert v_n\rvert$ 。 若用 n 元组表示则每个状态为 $\langle a_1,...,a_n\rangle$ 其中 $a_1,...,a_n$ 分别为 $v_1,...,v_n$ 的值。
迁移集合由 ρ 产生。 设 $\sigma,\sigma' \in \Sigma$ 。若 $\sigma \to \sigma'$ ，则 $(\langle\sigma(v_1),\sigma(v_2),...,\sigma(v_n)\rangle, \langle\sigma'(v_1),\sigma'(v_2),...,\sigma'(v_n)\rangle) \in \Delta$ 。
初始状态集合 I 根据 Θ 定义产生。若 $\sigma \models_{\mathcal{I}} \Theta$ 为真，则 $\langle\sigma(v_1),\sigma(v_2),...,\sigma(v_n)\rangle \in I$ 。

对每个变量 v ，设其值域为 $\{a_1,...,a_{|v|}\}$ 。我们可以构造 $|v|$ 个命题 $\{p_1,...,p_{|v|}\}$ ，每个命题代表 $v = a_1, v = a_2, ..., v = a_{|v|}$ 中对应的一项。 设 $p_{v,a}$ 代表 $v = a$ ，定义 L 如下： $p_{x_i,a} \in L(\langle x_1,...,x_n\rangle)$ 当且仅当 $x_i = a$ 。这样，我们有一个原子命题集合 AP 和其上的标号 Kripke 模型 $M_{(\rho,\Theta)} = \langle S, \Delta, I, L\rangle$ 。

谓词迁移模型与标号 KS 的等价

给定 (B,V) 上的谓词迁移模型 (ρ,Θ) 和 B 上的解释 $\mathcal{I} = (D,\mathcal{I}_0)$ 其中 D 为有穷集合。给定 AP 上的标号 KS 模型 $M = \langle S,\Delta,I,L\rangle$ 。给定一个 AP 子集到谓词集合 $\{v = d|v \in V, d \in D\}$ 子集的一一对应关系 ζ 。

谓词迁移模型与标号 KS 是 ζ 等价的当且仅当对任意 (ρ,Θ) 的一个运行 $\sigma_0\sigma_1\cdots$ 存在 M 的一个运行 $s_0s_1\cdots$ 满足 $\sigma_i \models (v = d)$ 当且仅当 $\zeta^{-1}(v = d) \in L(s_i)$ ，且对任意 M 的一个运行 $s_0s_1\cdots$ 存在 (ρ,Θ) 的一个运行 $\sigma_0\sigma_1\cdots$ 满足 $p \in L(s_i)$ 当且仅当 $\sigma_i \models \zeta(p)$ 。

构造正确性

给定 (B,V) 上的谓词迁移模型 (ρ,Θ) 和 B 上的解释 $\mathcal{I} = (D,\mathcal{I}_0)$ 其中 D 为有穷集合。给定 $\zeta = \{(p_{x,a},(x = a))|x \in V, a \in D\}$ 。则谓词迁移模型 (ρ,Θ) 与标号 KS 模型 $M_{(\rho,\Theta)}$ 是 ζ 等价的。

标号 KS 上的模拟与互模拟关系

设 $M_1 = \langle S_1,\Delta_1,I_1,L_1\rangle$ 和 $M_2 = \langle S_2,\Delta_2,I_2,L_2\rangle$ 为原子命题集合 AP 上的标号 KS 。

$\sigma \subseteq S_1 \times S_2$ 是 M_1 到 M_2 的模拟关系当且仅当 σ 是有向图 $\langle S_1,\Delta_1\rangle$ 到 $\langle S_2,\Delta_2\rangle$ 的模拟关系且 σ 满足 （ 1 ）对所有 $(x,y) \in S_1 \times S_2$ ，若 $(x,y) \in \sigma$ 则 $L_1(x) = L_2(y)$ ； （ 2 ）对所有 $x \in I_1$ ，存在 $y \in I_2$ 满足 $(x,y) \in \sigma$ 。 若存在一个这样的 σ ，则称 M_2 模拟 M_1 。

$\sigma \subseteq S_1 \times S_2$ 是 M_1 和 M_2 上的互模拟关系当且仅当 σ 是有向图 $\langle S_1, \Delta_1\rangle$ 和 $\langle S_2, \Delta_2\rangle$ 上的互模拟关系且 σ 满足 （1）对所有 $(x,y) \in S_1 \times S_2$，若 $(x,y) \in \sigma$ 则 $L_1(x) = L_2(y)$；（2）对所有 $x \in I_1$，存在 $y \in I_2$ 满足 $(x,y) \in \sigma$，且对所有 $y \in I_2$，存在 $x \in I_1$ 满足 $(x,y) \in \sigma$。若存在一个这样的 σ，则称 M_2 与 M_1 互模拟。

§2.3 标号迁移系统与自动机

Kripke 模型描述的主要是状态和状态之间的转换关系，不关心是什么动作使得状态发生了变化。在某些时候动作也是系统描述中关键的一部分。

§2.3.1 标号迁移系统 (LTS)

Definition 2.9 一个标号迁移系统是一个四元组

$$\langle \Sigma, S, \Delta, I\rangle$$

其中 Σ 为标号集合， S 为状态集合， $\Delta \subseteq S \times \Sigma \times S$ 是一个有标号的迁移关系， I 为初始状态集。

运行

用 $s \xrightarrow{a} s'$ 表示存在从 s 到 s' 的 a 迁移，即 $(s, a, s') \in \Delta$。LTS 上的一个运行是状态集 S 上的一个无穷序列 $[s_i]_{i\geq 0}$ 满足 $s_0 \in I$ 且对任意 $i \geq 0,$ 存在 $a \in \Sigma$，使得 $s_i \xrightarrow{a} s_{i+1}$。

字符串

标号集 Σ 上的一个字符串是 Σ 上的一个无穷序列。 $[a_i]_{i\geq 1} \in \Sigma^\omega$ 是标号迁移系统 $M = \langle \Sigma, S, \Delta, I\rangle$ 上的一个字符串当且仅当存在 LTS 上的一个运行 $\pi = [s_i]_{i\geq 0}$ 使得对任意 $i \geq 0,$ $s_i \xrightarrow{a_{i+1}} s_{i+1}$。 π 称为字符串 ω 上的一个运行。

语言

标号迁移系统 M 上的字符串的集合是 M 的语言，记为 $\mathcal{L}(M)$。

双标号迁移系统

与 KS 类似，我们可以在 LTS 的状态上记载一些信息。

Definition 2.10 给定一个原子命题集合 AP。一个 AP 上的标号迁移结构是一个五元组

$$\langle \Sigma, S, \Delta, I, L\rangle$$

其中 $\langle \Sigma, S, \Delta, I\rangle$ 是一个标号迁移系统， $L : S \to 2^{AP}$ 为状态集到 AP 的幂集的映射。

§2.3.2 ω 自动机

对于 LTS 而言，只要满足状态迁移关系的无穷状态序列就是系统的一个运行。有时候为了更精确地描述系统的运行，我们需要对运行做一定限制以排除一些满足状态转换关系但不合理的无穷状态序列。 若要求能够排除这样的无穷状态序列，我们需要在结构中加入新的成分。我们在 LTS 的基础上，为其运行设置可接受条件，定义 ω 自动机。

Büchi 自动机

为简单起见，我们先定义只有一个简单接受条件的 Büchi 自动机。

Definition 2.11 一个 *Büchi* 自动机是一个五元组

$$\langle \Sigma, S, \Delta, I, F \rangle$$

其中 $\langle \Sigma, S, \Delta, I \rangle$ 是一个标号迁移系统， $F \subseteq S$ 为自动机的接受状态集。

可接受运行

Büchi 自动机 $\mathcal{B} = \langle \Sigma, S, \Delta, I, F \rangle$ 上的运行就是标号迁移系统 $\langle \Sigma, S, \Delta, I \rangle$ 上的运行。 $\mathcal{B}$ 上的运行 $\pi = [s_i]_{i \geq 0}$ 是可接受的当且仅当

$$inf(\pi) \cap F \neq \emptyset$$

语言

$\mathcal{B}$ 的标号集 Σ 上的一个字符串 $\omega = [a_i]_{i \geq 1}$ 是可接受的当且仅当存在 ω 上的可接受运行。$\mathcal{B}$ 上可接受字符串的集合为 $\mathcal{B}$ 的语言，记作 $\mathcal{L}(\mathcal{B})$ 。

扩展 Büchi 自动机

为了能够表示多个接受条件，我们扩展 Büchi 自动机的定义，将接受状态集合改为接受状态集的集合。

Definition 2.12 一个扩展 *Büchi* 自动机是一个五元组

$$\langle \Sigma, S, \Delta, I, F \rangle$$

其中 $\langle \Sigma, S, \Delta, I \rangle$ 是一个标号迁移系统， $F \subseteq 2^S$ 为自动机的接受状态集的集合。

可接受运行

扩展 Büchi 自动机 $\mathcal{B} = \langle \Sigma, S, \Delta, I, F \rangle$ 上的运行就是标号迁移系统 $\langle \Sigma, S, \Delta, I \rangle$ 上的运行。$\mathcal{B}$ 上的运行 $\pi = [s_i]_{i \geq 0}$ 是可接受的，当且仅当对所有 $f \in F$ ，

$$inf(\pi) \cap f \neq \emptyset$$

自动机的等价

一个自动机等价于另一个自动机当且仅当其所定义的语言是相同的。两类自动机的表达能力相同，当且仅当任意一类自动机的一个实例都有一个等价的另一类自动机的一个实例。 扩展 Büchi 自动机的表达能力和 Büchi 自动机是一样的。

自动机的运算

自动机的运算包括并、交和补。 A' 是 A 和 B 的并自动机，记作 $A = A \cup B$ ，当前仅当 $\mathcal{L}(A') = \mathcal{L}(A) \cup \mathcal{L}(B)$ 。 A' 是 A 和 B 的交自动机，记作 $A = A \cap B$ ，当前仅当 $\mathcal{L}(A') = \mathcal{L}(A) \cap \mathcal{L}(B)$ 。 A' 是 A 的补自动机，记作 $A' = \neg A$ ，当前仅当 $\mathcal{L}(A') = \Sigma^\omega \setminus \mathcal{L}(A)$ 。 以下定义扩展 Büchi 自动机的并和交。

给定两个扩展 Büchi 自动机 $\mathcal{B}_1 = \langle \Sigma, S_1, \Delta_1, I_1, F_1 \rangle, \mathcal{B}_2 = \langle \Sigma, S_2, \Delta_2, I_2, F_2 \rangle$ 且 $S_1 \cap S_2 = \emptyset$ 。设

$$F = \{f \cup S_2 | f \in F_1\} \cup \{f \cup S_1 | f \in F_2\}$$

定义 $\mathcal{B}_1 \cup \mathcal{B}_2 = \langle \Sigma, S_1 \cup S_2, \Delta_1 \cup \Delta_2, I_1 \cup I_2, F \rangle$ ，则 $\mathcal{L}(\mathcal{B}_1 \cup \mathcal{B}_2) = \mathcal{L}(\mathcal{B}_1) \cup \mathcal{L}(\mathcal{B}_2)$ 。

给定两个扩展 Büchi 自动机 $\mathcal{B}_1 = \langle \Sigma, S_1, \Delta_1, I_1, F_1 \rangle, \mathcal{B}_2 = \langle \Sigma, S_2, \Delta_2, I_2, F_2 \rangle$ 且 $S_1 \cap S_2 = \emptyset$ 。设

$$\Delta = \{((q_1, q_2), a, (q_1', q_2')) | (q_1, a, q_1') \in \Delta_1, (q_2, a, q_2') \in \Delta_2\}$$
$$F = \{f \times S_2 | f \in F_1\} \cup \{S_1 \times f | f \in F_2\}$$

定义 $\mathcal{B}_1 \cap \mathcal{B}_2 = \langle \Sigma, S_1 \times S_2, \Delta, I_1 \times I_2, F \rangle$ ，则 $\mathcal{L}(\mathcal{B}_1 \cap \mathcal{B}_2) = \mathcal{L}(\mathcal{B}_1) \cap \mathcal{L}(\mathcal{B}_2)$ 。

扩展 Büchi 自动机是 Büchi 自动机的一种扩展，在扩展 Büchi 自动机基础上再对其接受条件做进一步扩展或重新定义，我们可以得到 Streett 自动机、 Rabin 自动机或 Muller 自动机。

Streett 自动机

Definition 2.13 一个 *Streett* 自动机是一个五元组

$$\langle \Sigma, S, \Delta, I, F \rangle$$

其中 $\langle \Sigma, S, \Delta, I \rangle$ 是一个标号迁移系统， $F \subseteq 2^S \times 2^S$ 为自动机的接受状态集的集合。

可接受运行

设 $\mathcal{B} = \langle \Sigma, S, \Delta, I, F \rangle$ 是 Streett 自动机。 $\mathcal{B}$ 上的运行就是标号迁移系统 $\langle \Sigma, S, \Delta, I \rangle$ 上的运行。 $\mathcal{B}$ 上的运行 $\pi = [s_i]_{i \geq 0}$ 是可接受的，当且仅当对所有 $(f, g) \in F$ ，

$$inf(\pi) \cap f \neq \emptyset \rightarrow inf(\pi) \cap g \neq \emptyset$$

Rabin 自动机

Rabin 自动机的结构和 Streett 自动机完全一样，只是接受条件正好和 Streett 自动机的接受条件相反。给定一个 Rabin 自动机 $\mathcal{B} = \langle \Sigma, S, \Delta, I, F \rangle$ 。 $\mathcal{B}$ 上的运行 $\pi = [s_i]_{i \geq 0}$ 是可接受的，当且仅当存在 $(f, g) \in F$ ，

$$inf(\pi) \cap f \neq \emptyset \wedge inf(\pi) \cap g = \emptyset$$

Muller 自动机

Definition 2.14 一个 *Muller* 自动机是一个五元组

$$\langle \Sigma, S, \Delta, I, F \rangle$$

其中 $\langle \Sigma, S, \Delta, I \rangle$ 为迁移系统， $F \subseteq 2^S$ 为自动机的接受状态集的集合。

可接受运行

设 $\mathcal{B} = \langle \Sigma, S, \Delta, I, F \rangle$ 是 Muller 自动机。 $\mathcal{B}$ 上的运行就是标号迁移系统 $\langle \Sigma, S, \Delta, I \rangle$ 上的运行。 $\mathcal{B}$ 上的运行 $\pi = [s_i]_{i \geq 0}$ 是可接受的，当且仅当

$$inf(\pi) \in F$$

基于迁移的扩展 Büchi 自动机

以上自动机的接受条件都是状态集合。我们也可以将接受条件定义在迁移上。以下定义基于迁移的扩展 Büchi 自动机。

Definition 2.15 基于迁移的扩展 *Büchi* 自动机是一个五元组

$$\langle S, \Sigma, \Delta, I, F \rangle$$

其中 $\langle \Sigma, S, \Delta, I \rangle$ 是一个标号迁移系统， $F \subseteq 2^\Delta$ 是接受迁移集合。

可接受运行

用 $[(s_i,a_{i+1},s_{i+1})]_{i\geq 0}$ 表示

$$(s_0,a_1,s_1)(s_1,a_2,s_2)(s_2,a_3,s_3)\cdots$$

定义 $inf([(s_i,a_{i+1},s_{i+1})]_{i\geq 0})$ 为无限多次出现在 $[(s_i,a_{i+1},s_{i+1})]_{i\geq 0}$ 中的迁移的集合。

基于迁移的 Büchi 自动机 $\mathcal{B}=\langle\Sigma,S,\Delta,I,F\rangle$ 上的运行就是标号迁移系统 $\langle\Sigma,S,\Delta,I\rangle$ 上的运行。 $\mathcal{B}$ 上的运行 $\pi=[s_i]_{i\geq 0}$ 是可接受的，当且仅当存在 Σ 上的字符串 $[a_i]_{i\geq 1}$ 使得对所有 $f\in F$ ，

$$inf([(s_i,a_{i+1},s_{i+1})]_{i\geq 0})\cap f\neq\emptyset$$

可接受字符串

一个字符串 $[a_i]_{i\geq 1}$ 是可接受的当且仅当存在 $[a_i]_{i\geq 1}$ 上的运行 $[s_i]_{i\geq 0}$ ，使得对所有 $f\in F$ ，

$$inf([(s_i,a_{i+1},s_{i+1})]_{i\geq 0})\cap f\neq\emptyset$$

表达能力

从语言的角度，Büchi 自动机、扩展 Büchi 自动机、Streett 自动机、Rabin 自动机、Muller 自动机与基于迁移的 Büchi 扩展自动机的表达能力相同，且都是补封闭的。

语言非空与语言包含

$\mathcal{L}(A)\subseteq\mathcal{L}(B)$ 当且仅当 $\mathcal{L}(A)\cap(\Sigma^\omega\setminus\mathcal{L}(B))=\emptyset$ 当且仅当 $\mathcal{L}(A\cap\neg B)=\emptyset$ 。

§2.3.3 标号交错迁移系统与交错自动机

标号迁移系统的每个动作其后续状态可能有多个，但每次运行只能从其中选择一个状态。对这类迁移系统的一种扩展是让其可以选择多个后续状态。这样扩展的迁移系统的一次运行可以是树状结构。

标号交错迁移系统 (ATS)

Definition 2.16 *一个标号交错迁移系统是一个四元组 $\langle\Sigma,S,\Delta,I\rangle$ ，其中 Σ 为动作集合， S 为状态集合， $\Delta\subseteq S\times\Sigma\times 2^S$ 为标号的迁移关系， I 为系统初始状态的集合。*

字符串上的运行

给定一个 Σ 上的无限长的字符串 $\omega=a_0a_1a_2...$ 。标号交错迁移系统在 ω 上的一个运行是状态集 S 上的一颗树 r 。设 $r(0)$ 为 r 的根节点的单点集、 $r(i)$ 为 r 的第 i 层节点的集合、 $child(x)$ 为节点 x 的子节点的集合。用 $\Delta(x,a)$ 表示集合 $\{y|(x,a,y)\in\Delta\}$ 。 r 满足以下条件。

$r(0)\subseteq I$
若 $x\in r(i)$ 为第 i 层的一个节点，则 $child(x)\in\Delta(x,a_i)$

若 $\Delta(x,a_i)$ 包含空集，则 x 可以没有子节点。

运行

状态集 S 上的一颗树 r 是标号交错迁移系统 $\langle\Sigma,S,\Delta,I\rangle$ 的一个运行当且仅当存在一个 Σ 上的无限长的字符串 $\omega=a_0a_1a_2...$ 使得 r 是标号交错迁移系统在 ω 上的一个运行。

§2.3.4 双标号交错迁移系统

与 KS 类似，我们可以在 ATS 的状态上记载一些信息。

Definition 2.17 给定一个原子命题集合 AP 。一个 AP 上的双标号交错迁移系统是一个五元组

$$\langle \Sigma, S, \Delta, I, L \rangle$$

其中 $\langle \Sigma, S, \Delta, I \rangle$ 是一个标号交错迁移系统， $L : S \to 2^{AP}$ 为状态集到 AP 的幂集的映射。

交错 Büchi 自动机

Definition 2.18 一个交错 *Büchi* 自动机是一个五元组

$$\langle \Sigma, S, \Delta, I, F \rangle$$

其中 $\langle \Sigma, S, \Delta, I \rangle$ 是一个标号交错迁移系统， $F \subseteq S$ 为自动机的接受状态集。

可接受运行

交错 Büchi 自动机 $\mathcal{B} = \langle \Sigma, S, \Delta, I, F \rangle$ 是上的运行就是标号交错迁移系统 $\langle \Sigma, S, \Delta, I \rangle$ 上的运行。 $\mathcal{B}$ 的一个运行树 r 是可接受的，当且仅当 r 的所有无穷路径 ρ 满足

$$inf(\rho) \cap F \neq \emptyset$$

表达能力

字母表 Σ 上的一个字符串 ω 是可接受的当且仅当存在 ω 上的可接受运行树。交错 Büchi 自动机与 Büchi 自动机的表达能力相同。

§2.3.5 确定型自动机与非确定型自动机

ω- 自动机由迁移系统和接受条件组成。不同的接受条件定义了不同的自动机。我们也可以对迁移系统分类，得到确定型自动机和非确定型自动机。确定型自动机的初始状态只有一个且其迁移关系受以下限制：

$$(s, a, s'), (s, a, s'') \in \Delta \to s' = s''$$

给定一个字符串。对确定型自动机来讲，若有与之匹配的运行，则这样的运行是唯一的。

表达能力

确定型 Muller 自动机是补封闭的，其表达能力与非确定型 Muller 自动机相同。确定型 Büchi 自动机不是补封闭的。

§2.3.6 不同模型的关系

以上介绍了不同类型的自动机。部分自动机模型等价于公平标号 Kripke 模型。以上的非确定型自动机都可以互相转换并保持语言等价。

公平标号 Kripke 模型与扩展 Büchi 自动机

给定 AP 上的公平标号 Kripke 模型 $\mathcal{M} = \langle S, \Delta, I, L, F \rangle$ 。定义

Σ	$=$	2^{AP}
Δ'	$=$	$\{(s, a, s) \mid (s, s') \in \Delta, a = L(s)\}$

定义扩展 Büchi 自动机 $\mathcal{B} = \langle \Sigma, S, \Delta', I, F \rangle$ ，则 $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{M})$ 。

扩展 Büchi 自动机与 Büchi 自动机

给定扩展 Büchi 自动机 $\mathcal{B} = \langle \Sigma, S, \Delta, I, \{f_1, ..., f_n\} \rangle$ 。定义

$n(s,i)$	$=$	if $(i = n)$ then 0; else if $(s \in f_{i+1})$ then $i+1$; else i.
S'	$=$	$S \times \{0, ..., n\}$
Δ'	$=$	$\{(s,i), a, (s', n(s',i)) \| (s,a,s') \in \Delta, i \in \{0,...,n\}\}$
I'	$=$	$I \times \{0\}$
F'	$=$	$S \times \{n\}$

定义 Büchi 自动机 $\mathcal{B}' = \langle \Sigma, S', \Delta', I', F' \rangle$ ，则 $\mathcal{L}(\mathcal{B}') = \mathcal{L}(\mathcal{B})$ 。

基于迁移的扩展 Büchi 自动机与 Büchi 自动机

给定基于迁移的扩展 Büchi 自动机 $\mathcal{B} = \langle \Sigma, S, \Delta, I, F \rangle$ ，定义

$n(s,a,s',i)$	$=$	if $(i = n)$ then 0; else if $((s,a,s') \in f_{i+1})$ then $i+1$; else i.
S'	$=$	$S \times \{0, ..., n\}$
Δ'	$=$	$\{(s,i), a, (s', n(s,a,s',i)) \| (s,a,s') \in \Delta, i \in \{0,...,n\}\}$
I'	$=$	$I \times \{0\}$
F'	$=$	$S \times \{n\}$

定义 Büchi 自动机 $\mathcal{B}' = \langle \Sigma, S', \Delta', I', F' \rangle$ ，则 $\mathcal{L}(\mathcal{B}') = \mathcal{L}(\mathcal{B})$ 。 由 Büchi 自动机，也可构造语言与之相同的基于迁移的扩展 Büchi 自动机。因此基于迁移的扩展 Büchi 自动机的表达能力和 Büchi 自动机是一样的。

§2.4 时间迁移系统和混成系统

对一些系统，我们除了关系系统的控制状态外，可能还关心在一个状态持续的时间里或两个状态之间的一些量变化。

§2.4.1 时间迁移系统 (TTS)

为了能够描述迁移动作之间时间长短的限制，我们需要在模型中加入时钟的概念。 时间的限制由时间变量上的公式表示。设 X 为时钟变量的集合， Q 为时间常量的集合。时钟公式的集合 $\Phi(X)$ 由以下语法给出。

$$\phi ::= x \leq c | c \leq x | \neg \phi | \phi \wedge \phi$$

其中 $x \in X$ 为时钟变量， $c \in Q$ 为时间常量。一个时钟赋值 v 是一个 X 到 R 的函数。 用 $v + t$ 表示对所有 $x \in X$ 满足 $v'(x) = v(x) + t$ 的时钟赋值 v' ； 用 $t \cdot v$ 表示对所有 $x \in X$ 满足 $v'(x) = t \cdot v(x)$ 的时钟赋值 v' ； 用 $[Y \to t]v$ 表示对所有 $x \in X \setminus Y$ 满足 $v'(x) = v(x)$ 和对所有 $x \in Y$ 满足 $v'(x) = t$ 的时钟赋值 v' 。

Definition 2.19 一个时间迁移系统是一个五元组

$$\langle \Sigma, S, X, \Delta, I \rangle$$

其中 Σ 为字母表， S 为状态集合， X 为时钟变量集合， $\Delta \subseteq S \times \Sigma \times 2^X \times \Phi(X) \times S$ 为迁移关系， $I \in S$ 为系统初始状态的集合。

系统状态

记 $V = X \to R$ 为时钟赋值的集合。一个系统状态为 $S \times V$ 的一个元素。

时间字符串上的运行

一个时间字符串为 $\Sigma \times R$ 上的无穷序列。为方便起见，也写成 $(\sigma, \tau) = ([\sigma_i]_{i\geq 1}, [\tau_i]_{i\geq 1}) \in \Sigma^\omega \times R^\omega$ 其中其中 τ 满足对所有 i, $\tau_{i+1} > \tau_i$ 且对任意 $t \in R$ 存在 i, $\tau_i > t$ 。有时候因建模需要， $\tau_{i+1} > \tau_i$ 可以弱化成 $\tau_{i+1} \geq \tau_i$ 。

定义 $\tau_0 = 0$ 。时间迁移系统 $\langle \Sigma, S, X, \Delta, I\rangle$ 在时间字符串 $(\sigma, \tau) = [(\sigma_i, \tau_i)]_{i\geq 1} \in (\Sigma \times R)^\omega$ 上的一个运行是状态集 $S \times V$ 上的一个无穷序列 $[(s_i, v_i)]_{i\geq 0}$ 满足

$s_0 \in I$,
对所有 $x \in X$,$v_0(x) = 0$,
对任意 $i \geq 0$ ，存在 $\lambda \subseteq X$, $\varphi \in \Phi(X)$, $(s_i, \sigma_{i+1}, \lambda, \varphi, s_{i+1}) \in \Delta$ 使得
$(v_i + \tau_{i+1} - \tau_i)$ 满足 φ 且 $v_{i+1} = [\lambda \to 0](v_i + \tau_{i+1} - \tau_i)$ 。

运行

$S \times V$ 上的一个无穷序列 r 是时间迁移系统 $\langle \Sigma, S, X, \Delta, I\rangle$ 一个运行当且仅当存在一个时间字符串 (σ, τ) 使得 r 是时间迁移系统在 (σ, τ) 上的一个运行。时间迁移系统的状态可达性问题的复杂度是 PSPACE 完全的。

§2.4.2 时间 Büchi 自动机 (TA)

Definition 2.20 一个时间 *Büchi* 自动机是一个六元组

$$\langle \Sigma, S, X, \Delta, I, F\rangle$$

其中 $\langle \Sigma, S, X, \Delta, I\rangle$ 构成一个时间迁移系统， $F \subseteq S$ 为自动机的接受状态集。

可接受运行

时间 Büchi 自动机 $\mathcal{B} = \langle \Sigma, S, X, \Delta, I, F\rangle$ 。一个时间字符串 (σ, τ) 上的 B 的运行 $r = (s, v)$ 是可接受的，当且仅当 $inf(r) \cap F \neq \emptyset$ ，其中 $inf(r)$ 代表无限多次出现在运行 r 中的状态的集合。

可接受时间字符串

若一个时间字符串 (σ, τ) 上存在 B 的可接受运行，则称 (σ, τ) 是可接受。$\mathcal{B}$ 上可接受时间字符串的集合为 $\mathcal{B}$ 的语言，记作 $\mathcal{L}(\mathcal{B})$ 。 时间自动机是并和交封闭的，但不是补封闭的。时间自动机的语言非空问题的复杂度是 PSPACE 完全的。

§2.4.3 混成迁移系统

混成迁移系统是时间迁移系统的一种扩充。设 $X = \{x_1, ..., x_n\}$ 为实数变量的集合。记 $\Phi(X)$ 为 X 上谓词的集合。记 $\uplus(X) : X \to (R^n \to R)$ 为偏函数的集合。记 $\dot{X} = \{\dot{x}_1, ..., \dot{x}_n\}$ 。

Definition 2.21 一个混成迁移系统是一个六元组

$$\langle \Sigma, S, X, \Delta, I, flow\rangle$$

其中 Σ 为字母表， S 为状态集合， X 为实数变量集合， $\Delta \subseteq S \times \Sigma \times \uplus(X) \times \Phi(X) \times S$ 为迁移关系， $I \in S \times (2^R)^n$ 为系统初始状态的集合， $flow : S \to \Phi(X \cup \dot{X})$ 为每个状态标上一个系统变量变化的约束条件。

系统状态和状态的迁移

记 $V = X \to R$ 为变量赋值的集合。一个系统状态为 $S \times V$ 的一个元素。为方便起见，用 $v \in V$ 表示 $(v(x_1), ..., v(x_n)) \in R^n$ 。

对 $\delta \in R_{\geq 0}$ ，用 $(s, v) \xrightarrow{\sigma} (s, v')$ 表示 v' 满足以下条件：存在一个可微函数 $f : [0, \delta] \to R^n$ 和其一阶微分 $\dot{f} : (0, \delta) \to R^n$ 满足 $f(0) = v$, $f(\delta) = v'$ 且

$$\forall \zeta \in (0, \delta). flow(s)[X/f(\zeta)][\dot{X}/\dot{f}(\zeta)] = true.$$

用 $(s, v) \xrightarrow{\sigma} (s', v')$ 表示 (s', v') 满足以下条件：存在 $\lambda \in \uplus(X)$, $\varphi \in \Phi(X)$, $(s, \sigma, \lambda, \varphi, s') \in \Delta$ 且 $\{z_1, ..., z_k\} = dom(\lambda)$ 使得 v 满足 φ 且 $v' = v[z_1/\lambda(z_1)(v)] \cdots [z_k/\lambda(z_k)(v)]$ 。

时间字符串上的运行

一个时间字符串为 $\Sigma \times R$ 上的无穷序列。混成迁移系统 $\langle \Sigma, S, X, \Delta, I, flow \rangle$ 在时间字符串 $(\sigma, \tau) = [(\sigma_i, \tau_i)]_{i \geq 1} \in (\Sigma \times R)^\omega$ 上的一个运行是状态集 $S \times V$ 上的一个无穷序列 $[(s_i, v_i)]_{i \geq 0}$ 满足 $(s_0, v_0) \in I$ 且对任意 $i \geq 0$ ，存在 v_i' 使得 $(s_i, v_i) \xrightarrow{\tau_{i+1} - \tau_i} (s_i, v_i')$ 且 $(s_i, v_i') \xrightarrow{\delta_{i+1}} (s_{i+1}, v_{i+1})$ 。

运行

$S \times V$ 上的一个无穷序列 r 是混成迁移系统 $\langle \Sigma, S, X, \Delta, I, flow \rangle$ 一个运行当且仅当存在一个时间字符串 (σ, τ) 使得 r 是混成迁移系统在 (σ, τ) 上的一个运行。

§2.5 Petri 网

Petri 网的特点是其可以描述真并发。其基本要素有库所、迁移、连接库所和迁移的有向边。由库所进入迁移的边可以看成是输入条件，由迁移进入库所的边可以看成是迁移的输出。

Definition 2.22 *一个 Petri 网是一个四元组*

$$\langle P, T, F, M_0 \rangle$$

其中 P 为库所集合， T 为迁移集合， $F \subseteq (P \times T) \cup (T \times P)$ 为边的集合， $M_0 : P \to N$ 为初始状态，其中 N 为自然数集合。

系统状态

一个系统状态为 $P \to N$ 的一个函数，是对所有库所的一个赋值。

迁移

定义 ${}^\circ p(t) = \{p \in P | (p, t) \in F\}$ 和 $p^\circ(t) = \{p \in P | (t, p) \in F\}$ 。迁移 $t \in T$ 在状态 M 是可执行的，当且仅当输入条件满足，即 $\forall p \in^\circ p(t), M(p) \geq 1$ 。 用 $M \xrightarrow{t} M'$ 表示 $t \in T$ 在状态 M 是可执行且 $\forall p \in P$ ， $M'(p) = M(p) - \alpha_0(p, t) + \alpha_1(p, t)$ 其中

$\alpha_0(p, t) = 1$ 当且仅当 $p \in^\circ p(t)$
$\alpha_1(p, t) = 1$ 当且仅当 $p \in p^\circ(t)$

可达状态

给定 Petri 网 $Q = \langle P, T, F, M_0 \rangle$ 。用 $M \to M'$ 表示存在 $t \in T$ 使得 $M \xrightarrow{t} M'$ 成立。则 Q 的可达状态集合可表示为 $\{M | M_0 \xrightarrow{*} M\}$ 。 Petri 网的状态可达性问题的复杂度是可判定的且是 EXPSPACE 难的。

安全 Petri 网

Petri 网 $\langle P, T, F, M_0\rangle$ 是 k 界的，当且仅当对其所有可达状态 M 有 $\forall p \in P.M(p) \leq k$ 。 1 界 Petri 网称为安全 Petri 网。安全 Petri 网的状态可达性问题的复杂度是 PSPACE 完备的。

§2.6 通信系统

通信系统模型用来表达多个进程通过通道交换信息进行控制和计算的过程。每个进程内部有状态和状态迁移。状态迁移的原因可以是内部事件或输入输出。

§2.6.1 通道

Definition 2.23 给定一个类型 $\langle S, N\rangle$ 其中 S 为字母表的， N 为自然数。一个类型为 $\langle S, N\rangle$ 的通道 m ，记作 $m \in \langle S, N\rangle$ ，是一个值域为 $\bigcup_{i=0}^{N}\{\langle x_1, ..., x_i\rangle | x_i \in S\}$ 的变量。

通道状态记载通道的给定赋值，记为 σ 。若 $m \in \langle S, N\rangle$ ，则 $\sigma(m) \in \bigcup_{i=0}^{N}\{\langle x_1, ..., x_i\rangle | x_i \in S\}$ 。通道状态空间为通道状态的集合，记为 Σ 。

事件

若 $m \in \langle S, N\rangle$ ，则与 m 相关的事件集合 $\alpha(m)$ 定义为 $\alpha(m) = \{m?s | s \in S\} \cup \{m!s | s \in S\}$ 。设 $C = \{m_1, ..., m_n\}$ 为通道集合。则与 C 相关的事件集合为 $\alpha(C) = \alpha(m_1) \cup \cdots \cup \alpha(m_n)$ 。与通道无关的内部事件记为 $\{\epsilon\}$ 。

可执行事件

对于 $x = \langle x_1, ..., x_n\rangle$ ，定义 $|x| = n$, $x \vdash s = \langle x_1, ..., x_n, s\rangle$, $HEAD(x) = x_1$, $TAIL(x) = \langle x_2, ..., x_n\rangle$ 。给定一个通道状态 σ 和一个通道 $m \in \langle S, N\rangle$ 。若以下一项成立则 $a \in \alpha(m) \cup \{\epsilon\}$ 可执行。

$a = \epsilon$
$a = m!s$ 且 $\lvert\sigma(m)\rvert < N$ 且 $s \in S$
$a = m?s$ 且 $\lvert\sigma(m)\rvert > 0$ 且 $s = HEAD(\sigma(m))$

给定 $\sigma \in \Sigma$,$m \in \langle S, N\rangle$,$a \in \alpha(m)$ 。用 $\sigma \xrightarrow{a} \sigma'$ 表示 a 在 σ 可执行且

若 $a = \epsilon$,	则 $\sigma' = \sigma$
若 $a = m!s$,	则 $\sigma' = \sigma[m/\sigma(m) \vdash s]$
若 $a = m?s$,	则 $\sigma' = \sigma[m/TAIL(\sigma(m))]$

§2.6.2 通信单元

Definition 2.24 一个通信单元是一个四元组

$$\langle Q, C, \Delta, q_0\rangle$$

其中 Q 为状态集合， C 为通道集合， $\Delta \subseteq Q \times \{\alpha(C) \cup \{\epsilon\}\} \times Q$ 为标号的迁移关系， $q_0 \in Q$ 为初始状态。

系统状态

给定一个通信单元 $\langle Q, C, \Delta, q_0\rangle$ ，系统状态由两部分组成 (s,σ) 其中 $s \in Q$ 为控制状态，$\sigma \in \Sigma$ 为通道状态。

运行

迁移 (q,a,q') 在 (s,σ) 可执行，当且仅当 $q = s$ 且 a 在 σ 可执行。 设 $C = \{m_1, ..., m_k\}$ 。一个运行是 $Q \times \Sigma$ 上的一个无穷序列 $[(z_i, \sigma_i)]_{i \geq 0}$ 满足 $z_0 = q_0$, $\sigma_0(m_1) = \cdots = \sigma_0(m_k) = \langle\rangle$ 且对任意 $i \geq 0$,

存在 $(q,a,q') \in \Delta$
(q,a,q') 在 (z_i,σ_i) 可执行
$z_{i+1} = q'$ 且 $\sigma_i \xrightarrow{a} \sigma_{i+1}$

§2.6.3 通信系统

Definition 2.25 一个通信系统 $\{P_1, ..., P_n\}$ 是由通信单元 $P_1 = \langle Q_1, C_1, \Delta_1, q_{10}\rangle$, ..., $P_n = \langle Q_n, C_n, \Delta_n, q_{n0}\rangle$ 组成的集合，其中 $Q_1, ..., Q_n$ 为两两不相交的集合。

系统状态

定义 $S_1 \otimes \cdots \otimes S_n = \{\{s_1, ..., s_n\} | s_1 \in S_1, ..., s_n \in S_n\}$ 。通信系统的系统状态的集合为

$$(Q_1 \otimes \cdots \otimes Q_n) \times \Sigma$$

运行

一个迁移 $(q,a,q') \in \Delta_1 \cup \cdots \cup \Delta_n$ 在状态 (Q,σ) 可执行当且仅当 $q \in Q$ 且 a 在 σ 可执行。设 $C_1 \cup \cdots \cup C_n = \{m_1, ..., m_k\}$ 。一个运行是 $(Q_1 \otimes \cdots \otimes Q_n) \times \Sigma$ 上的一个无穷序列 $[(z_i, \sigma_i)]_{i \geq 0}$ 满足 $z_0 = \{q_0, ..., q_n\}$, $\sigma_0(m_1) = \cdots = \sigma_0(m_k) = \langle\rangle$ 且对任意 $i \geq 0$,

存在 $(q,a,q') \in \Delta_1 \cup \cdots \cup \Delta_n$
(q,a,q') 在 (z_i,σ_i) 可执行
$z_{i+1} = (z_i \setminus \{q\}) \cup \{q'\}$ 且 $\sigma_i \xrightarrow{a} \sigma_{i+1}$

并发算子

给定两个通信单元 $P_1 = \langle Q_1, C_1, \Delta_1, q_{10}\rangle$, $P_2 = \langle Q_2, C_2, \Delta_2, q_{20}\rangle$ 且 $Q_1 \cap Q_2 = \emptyset$ 。 P_1 和 P_2 的并发定义如下

$$P_1 || P_2 = \langle Q_1 \otimes Q_2, C_1 \cup C_2, \Delta, \{q_{10}, q_{20}\}\rangle$$

其中

$$\Delta = \{(\{q_1, q_2\}, a, \{q_1', q_2\}) | (q_1, a, q_1') \in \Delta_1\} \cup \{(\{q_1, q_2\}, a, \{q_1, q_2'\}) | (q_2, a, q_2') \in \Delta_2\}$$

并发算子满足结合率和交换率。

§2.7 说明

本章主要内容为程序及系统运行的模型。有关卫式迁移系统部分可参考 [Pel01] 。 Kripke 模型和自动机可参考 [CGP99] 。交错自动机部分可参考 [Var97, AHK97] 。时间自动机部分可参考 [AD94] 。混成自动机部分可参考 [Hen96] 。 Petri 网的介绍可参考 [NW96] 。 通信系统部分可参考 [Hol90] 。

参考文献

[AD94] Rajeev Alur and David L. Dill. A Theory of Timed Automata. Theor. Comput. Sci. 126(2): 183-235 (1994).

[AHK97] Rajeev Alur, Thomas A. Henzinger and Orna Kupferman. Alternating-Time Temporal Logic. COMPOS 1997: 23-60.

[CGP99] E. Clark, O. Grumberg and D. Peled. Model Checking. MIT press, 1999.

[Hen96] Thomas A. Henzinger. The Theory of Hybrid Automata. LICS 1996: 278-292.

[Hol90] Gerard J. Holzmann. Design and Validation of Computer Protocols. Prentice Hall, 1990.

[NW96] Mogens Nielsen and Glynn Winskel. Petri Nets and Bisimulation. Theor. Comput. Sci. 153(1&2): 211-244 (1996).

[Pel01] Doron A. Peled. Software Reliability Methods. Springer-Verlag. 2001.

[Var97] Moshe Y. Vardi. Alternating Automata: Unifying Truth and Validity Checking for Temporal Logics. CADE 1997: 191-206.