§3 程序逻辑

为了能够表达程序性质, 进而验证程序是否具有这样的性质，我们需要描述程序性质的语言，即程序逻辑。

.

程序逻辑的基础是命题逻辑、一阶逻辑和模态逻辑。命题逻辑和一阶逻辑作为程序的断言是表示程序性质的一种手段。

.

在命题逻辑和一阶逻辑的基础之上, 我们引进模态算子并将其含义解释成与时间先后次序有关的描述, 形成不同的时序逻辑。

§3.1 线性时序逻辑 (LTL)

线性时序逻辑 (Linear Temporal Logic) 关心的是系统运行中的状态以及它们之间的关系。线性时序逻辑可以建立在命题逻辑、谓词逻辑以及其它描述状态的逻辑之上。

例子： 互斥协议性质

§3.1.1 命题线性时序逻辑 (PLTL)

我们先考虑建立在命题逻辑上的线性时序逻辑, 即命题线性时序逻辑，记作 PLTL 。

.

给定一个原子命题集合 AP 。 PLTL 公式的集合由以下语法给出。

$$\phi ::= \quad \bot \mid \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \to \phi \mid \phi \leftrightarrow \phi \mid O\,\phi \mid \Diamond\,\phi \mid \Box\,\phi \mid \phi\,U\,\phi \mid \phi\,R\,\phi$$

其中 p 为 AP 中任意命题。

例子：

$$\Box(\neg(p \wedge q))$$
$$\Diamond q$$
$$p \to \Diamond q$$
$$\Box(p \to \Diamond q)$$

.

PLTL 公式在无限状态序列上解释。设 S 为状态集合，$\pi = \pi_0\pi_1\pi_2\cdots$ 为一个无限状态序列，我们用 π^k 来表示 π_k 开始的状态序列 $\pi_k\pi_{k+1}\cdots$ 。

.

S 中的状态上成立的命题由原子命题集 AP 上的 Kripke 结构 $M = \langle S, \Delta, I, L\rangle$ 决定。 PLTL 公式的语义如下：

$M, \pi^k \models p$	若 $p = \top$, 或 $p \in AP$ 且 $p \in L(\pi_k)$
$M, \pi^k \models \neg\varphi$	若 $M, \pi^k \not\models \varphi$
$M, \pi^k \models \varphi \vee \psi$	若 $M, \pi^k \models \varphi$ 或 $M, \pi^k \models \psi$
$M, \pi^k \models \varphi \wedge \psi$	若 $M, \pi^k \models \varphi$ 且 $M, \pi^k \models \psi$
$M, \pi^k \models \varphi \to \psi$	若 $M, \pi^k \models \varphi$ 则 $M, \pi^k \models \psi$
$M, \pi^k \models \varphi \leftrightarrow \psi$	若 $M, \pi^k \models \varphi \to \psi$ 且 $M, \pi^k \models \psi \to \varphi$
$M, \pi^k \models O\varphi$	若 $M, \pi^{k+1} \models \varphi$
$M, \pi^k \models \Box\psi$	若 $\forall i \geq k, M, \pi^i \models \psi$
$M, \pi^k \models \Diamond\varphi$	若 $\exists i \geq k, M, \pi^i \models \varphi$
$M, \pi^k \models \varphi U \psi$	若 $\exists i \geq k, M, \pi^i \models \psi$ 且 $\forall k \leq j < i, M, \pi^j \models \varphi$
$M, \pi^k \models \varphi R \psi$	若 $\forall i \geq k$, 若 $\forall k \leq j < i, M, \pi^j \not\models \varphi$ 则 $M, \pi^i \models \psi$

例子： 以互斥协议为例。
进程 A 申请进入临界区则一定能够进入临界区可以写成

$$\Box(a = s_1 \to \Diamond a = s_2) \text{ 或 } \Box(p_6 \to \Diamond p_7) \text{ 。}$$

先申请先得到可以写成

$$\Box(a = s_1 \wedge b \neq t_1 \wedge b \neq t_2 \to (a = s_2 \, R \, b \neq t_2)) \text{ 。}$$

.

设 φ 为 PLTL 公式， M 为 Kripke 结构。

$$M \models \varphi$$

当且仅当

对所有 M 的运行 $\pi = \pi_0\pi_1 \cdots$ ，有 $M, \pi \models \varphi$ 。

模态算子的对偶关系与 NNF 范式

只使用逻辑连接符 $\wedge, \vee, \neg$ 且逻辑连接符 $\neg$ 只出现在命题前面的公式称为 NNF 范式。每个 PLTL 公式等价于一个 PLTL 的 NNF 范式。

.

一个 PLTL 的 NNF 范式可应用以下对偶关系构造。

$(\Box p)$	$\equiv$	$\neg(\Diamond\neg p)$
(pRq)	$\equiv$	$\neg(\neg pU\neg q)$
(Op)	$\equiv$	$\neg(O\neg p)$

例子： 证明 $(Op) \equiv \neg(O\neg p)$

例子： 证明 $pRq \equiv (qU(q \wedge p)) \vee \Box q$

模态算子的极小完全集

除了以上对偶的算子之外，我们有以下等式。

$\Box p$	$\equiv$	$(\bot R p)$
$\Diamond p$	$\equiv$	$(\top U p)$

因此 $\{O, U\}$ 构成 PLTL 模态算子的一个完全集，且是极小完全集。

§3.1.2 PLTL 公式的推理

以 $\{O, U, \Box, \Diamond\}$ 为 PLTL 公式的模态算子集。 PLTL 的推理系统包含以下三部分：一部分为 PLTL 公式相关的时序逻辑公理；另一部分为命题逻辑的推理系统；第三部分为时序推理规则。

第一部分：公理

$\Box\neg p \leftrightarrow \neg\Diamond p$
$\Box(p \to q) \to (\Box p \to \Box q)$
$\Box p \to p$
$\Box p \to Op$
$\Box p \to O\Box p$
$\Box(p \to Op) \to (p \to \Box p)$
$O\neg p \leftrightarrow \neg Op$
$O(p \to q) \to (Op \to Oq)$
$(pUq) \leftrightarrow (q \vee (p \wedge O(pUq))$
$(pUq) \to \Diamond q$

习题： 用语义证明 $\Box(p \to Op) \to (p \to \Box p)$ 成立，并说明为什么从右推左不成立。

第二部分：命题逻辑推理系统

如果 p 是重言式，则 p 是公理。
如果 $\vdash p \to q$ 且 $\vdash p$，则 $\vdash q$（ MP 规则)。

第三部分：时序推理规则

如果 $\vdash p$，则 $\vdash \Box p$（泛化规则)。

例子： 证明 $(p \wedge \Box Op) \to \Box p$

例子： 证明 $(Op \to Oq) \to O(p \to q)$

可判定性

PLTL 公式的可满足性是可判定的，其判定复杂性为 PSPACE 完全。

.

只允许时序算子 $\Box$ 和 $\Diamond$ 的 PLTL 公式的子集记为 PLTL(F) 。PLTL(F) 的判定复杂性为 NP 完全。

例子： 信号灯变化：绿 $\to$ 红 $\to$ 黄 $\to$ 绿 $\to$ 红 $\to\cdots$

$$gr \to O(red \to O(ye \to Ogr))$$
$$\Box(gr \to O(red \to O(ye \to Ogr)))$$
$$(gr \to Ored) \vee (red \to Oye) \vee (ye \to Ogr)$$
$$\Box((gr \to Ored) \vee (red \to Oye) \vee (ye \to Ogr))$$
$$\Box((grUred) \vee (redUye) \vee (yeUgr))$$
$$\Box\neg((gr \wedge red) \vee (red \wedge ye) \vee (ye \wedge gr))$$

习题： 用 PLTL 写下信号等变化的规范：信号灯依次序绿红黄变化，每个状态有且只有一个信号，初始信号为黄色，黄色只停留一个状态，红绿色可以连续在多个状态上成立。

§3.1.3 一阶线性时序逻辑

一阶线性时序逻辑，即用一阶逻辑来描述状态性质，不是可判定的，并且不存在完备的一阶线性时序逻辑的推理系统。但其表达能力较强，能够表达更多的性质，可以用在适合推理的程序验证中。

.

给定一个一阶逻辑 $\mathcal{L}^B$，$\mathcal{L}^B$ 上的一阶线性时序逻辑公式的集合由以下语法给出。

$$\phi ::= \quad p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \to \phi \mid \phi \leftrightarrow \phi \mid O\,\phi \mid \Diamond\,\phi \mid \Box\,\phi \mid \phi\,U\,\phi \mid \phi\,R\,\phi$$

其中 p 为 $\mathcal{L}^B$ 的任意公式。

.

一阶线性时序逻辑公式在无限状态序列上解释。设 S 为状态集合，$\pi = \pi_0\pi_1\pi_2\cdots$ 为一个无限状态序列，S 中状态的结构及在其之上成立的公式决定于一阶逻辑 $\mathcal{L}^B$。一阶线性时序逻辑公式的语义如下：

$M, \pi^k \models p$	若 p 是 L_B 的公式且 $M, \pi_k \models p$
$M, \pi^k \models \neg\varphi$	若 $M, \pi^k \not\models \varphi$
$M, \pi^k \models \varphi \vee \psi$	若 $M, \pi^k \models \varphi$ 或 $M, \pi^k \models \psi$
$M, \pi^k \models \varphi \wedge \psi$	若 $M, \pi^k \models \varphi$ 且 $M, \pi^k \models \psi$
$M, \pi^k \models \varphi \to \psi$	若 $M, \pi^k \models \varphi$ 则 $M, \pi^k \models \psi$
$M, \pi^k \models \varphi \leftrightarrow \psi$	若 $M, \pi^k \models \varphi \to \psi$ 且 $M, \pi^k \models \psi \to \varphi$
$M, \pi^k \models O\varphi$	若 $M, \pi^{k+1} \models \varphi$
$M, \pi^k \models \Box\psi$	若 $\exists i \geq k, M, \pi^i \models \psi$
$M, \pi^k \models \Diamond\varphi$	若 $\exists i \geq k, M, \pi^i \models \varphi$
$M, \pi^k \models \varphi U \psi$	若 $\exists i \geq k, M, \pi^i \models \psi$ 且 $\forall k \leq j < i, M, \pi^j \models \varphi$
$M, \pi^k \models \varphi R \psi$	若 $\forall i \geq k$, 若 $\forall k \leq j < i, M, \pi^j \not\models \varphi$ 则 $M, \pi^i \models \psi$

推理规则

我们用 $\varphi \Rightarrow \psi$ 表示 $\Box(\varphi \to \psi)$。给定一个 $B = (F, P)$ 和 B 的解释 I。根据一阶时序逻辑公式的语义，我们有以下推理规则。

- 一般化：

$$\frac{\models \varphi}{\Box\varphi} \qquad \frac{\varphi \Rightarrow \psi}{\varphi \Rightarrow \varphi U \psi} \qquad \frac{\varphi \Rightarrow O\psi}{\varphi \Rightarrow \varphi U \psi}$$

- R 规则：

$$\frac{\begin{array}{l}\zeta \Rightarrow \varphi \\ \varphi \wedge \neg\psi \Rightarrow O\varphi\end{array}}{\zeta \Rightarrow \psi R \varphi}$$

- U 规则：

$$\frac{\begin{array}{l}\varphi \Rightarrow (\psi \vee \zeta)\\ \zeta \Rightarrow (\zeta_0 \wedge w_x^e)\\ (\zeta \wedge e = v) \Rightarrow \zeta_0 U(\psi \vee (\zeta \wedge e \sqsubset v))\end{array}}{\varphi \Rightarrow \zeta_0 U\psi}$$

其中 $\sqsubseteq \in P$ 为二元谓词符号； $w \in QF_B$ 为一元谓词公式 (记其变量为 x) 且 $W = (\{\sigma(x) \mid I(w)(\sigma) = true\}, I_0(\sqsubseteq))$ 为良基集合； $e \in T_B$ 为项。

一阶线性时序逻辑的扩展

以上一阶线性时序逻辑的模态算子都是作用在公式上。更进一步，我们可以允许 O 算子用在项上。这样，我们需要修改 $M, \pi^k \models p$ 的定义。设 p 是 $\mathcal{L}^B$ 的原子公式，则：

$M, \pi^k \models p(O^{i_1}x_1, ..., O^{i_n}x_n)$	若 O 不在 p 中的其它位置出现 且 $M, \pi^k \models p(\pi_{k+i_1}(x_1), ..., \pi_{k+i_n}(x_n))$
$M, \pi^k \models p(Of(O^{i_1}x_1, ..., O^{i_n}x_n))$	若 $M, \pi^k \models p(f(O^{i_1+1}x_1, ..., O^{i_n+1}x_n))$

§3.2 计算树逻辑 CTL

计算树逻辑 (Computation Tree Logic) 是一种分枝时序逻辑。

.

系统的可能的运行可以看成是一个树的结构，即一个状态可能有多个后续状态。一个系统由一颗或多颗树组成。计算树逻辑可以描述计算树的分枝情况和状态的前后关系。

例子： 互斥协议性质

.

我们先考虑一种简单的计算树逻辑，即 CTL 。 CTL 中描述分枝情况和描述状态的前后关系的算子成对出现，即一个描述分枝情况的算子后面必须有一个描述状态的前后关系的算子。

.

计算树逻辑可以建立在命题逻辑、谓词逻辑以及其它描述状态的逻辑之上。在这里，我们只考虑命题逻辑之上的计算树逻辑。

.

给定一个原子命题集合 AP 。 CTL 公式的集合由以下语法给出。

$$\begin{aligned}\phi ::= \quad & \top \mid \bot \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \\ & EX\,\phi \mid EF\,\phi \mid EG\,\phi \mid E(\phi\, U\, \phi) \mid E(\phi\, R\, \phi) \mid \\ & AX\,\phi \mid AF\,\phi \mid AG\,\phi \mid A(\phi\, U\, \phi) \mid A(\phi\, R\, \phi)\end{aligned}$$

其中 p 为 AP 中任意命题。为间单起见，逻辑联接符 $\rightarrow$ 和 $\leftrightarrow$ 没有出现在上面的公式定义中，但我们可以根据其通常的语义自由使用。

.

CTL 公式在一个树状结构或 Kripke 结构上解释。设 $M = \langle S, \Delta, I, L\rangle$ 是原子命题集 AP 上的 Kripke 结构。CTL 公式的语义如下：

$M, s \models p$	若 $p = \top$, 或 $p \in AP$ 且 $p \in L(s)$
$M, s \models \neg\varphi$	若 $M, s \not\models \varphi$
$M, s \models \varphi \vee \psi$	若 $M, s \models \varphi$ 或 $M, s \models \psi$
$M, s \models \varphi \wedge \psi$	若 $M, s \models \varphi$ 且 $M, s \models \psi$
$M, s \models A\varphi$	若对于所有 M 中以 s 为起点的路径 π ： $M, \pi \models \varphi$
$M, s \models E\varphi$	若存在 M 中以 s 为起点的路径 π ： $M, \pi \models \varphi$
$M, \pi \models X\varphi$	若 $M, \pi_1 \models \varphi$
$M, \pi \models G\psi$	若 $\forall i \geq 0, M, \pi_i \models \psi$
$M, \pi \models F\varphi$	若 $\exists i \geq 0, M, \pi_i \models \varphi$
$M, \pi \models \varphi U \psi$	若 $\exists i \geq 0, M, \pi_i \models \psi$ 且 $\forall 0 \leq j < i, M, \pi_j \models \varphi$
$M, \pi \models \varphi R \psi$	若 $\forall i \geq k$, 若 $\forall 0 \leq j < i, M, \pi_j \not\models \varphi$ 则 $M, \pi_i \models \psi$

例子： 以互斥协议为例。
进程 A 申请进入临界区则一定能够进入临界区可以写成

$$AG(a = s_1 \rightarrow AFa = s_2)\ 。$$

先申请先得到可以写成

$$AG(a = s_1 \wedge b \neq t_1 \wedge b \neq t_2 \rightarrow A(a = s_2\, R\, b \neq t_2))\ 。$$

.

设 φ 为 CTL 公式， M 为 Kripke 结构。

$$M \models \varphi$$

当且仅当

对所有 $s \in I$ ， $M, s \models \varphi$ 。

模态算子的对偶关系与 NNF 范式

逻辑连接符 $\neg$ 只出现在命题前面的公式称为 NNF 范式。每个 CTL 公式等价于一个 CTL 的 NNF 范式。一个 CTL 的 NNF 范式可应用以下对偶关系构造。

AXp	$\equiv$	$\neg EX\neg p$
AGp	$\equiv$	$\neg EF\neg p$
AFp	$\equiv$	$\neg EG\neg p$
$A(pRq)$	$\equiv$	$\neg E(\neg pU\neg q)$
$A(pUq)$	$\equiv$	$\neg E(\neg pR\neg q)$

模态算子的最小完全集

除了以上对偶的算子之外，我们有以下等式。

EGp	$\equiv$	$E(\bot Rp)$
$E(pRq)$	$\equiv$	$E(qU(p \wedge q)) \vee EGq$

因此 $\{EX, EG, EU\}$ 构成 CTL 模态算子的一个完全集，且是极小完全集。

习题： 讨论 $A(pUr) \vee A(qUr)$ 和 $A((p \vee q)Ur)$ 的关系及满足这些公式的结构的特征。

§3.2.1 CTL 公式的推理

以 $\{EX, EU, EG, EF, AX, AU, AG, AF, \}$ 为 CTL 公式的模态算子集。CTL 的推理系统包含以下三部分：一部分为 CTL 式相关的时序逻辑公理；另一部分为命题逻辑的推理系统；第三部分为时序推理规则。

第一部分：公理

$EFp \leftrightarrow E(\top Up)$
$AGp \leftrightarrow \neg EF\neg p$
$AFp \leftrightarrow A(\top Up)$
$EGp \leftrightarrow \neg AF\neg p$
$EX(p \vee q) \leftrightarrow EXp \vee EXq$
$AXp \leftrightarrow \neg EX\neg p$
$E(pUq) \leftrightarrow (q \vee (p \wedge EXE(pUq))$
$A(pUq) \leftrightarrow (q \vee (p \wedge AXA(pUq))$
$EX\top \wedge AX\top$
$AG(r \rightarrow (\neg q \wedge EXr)) \rightarrow (r \rightarrow \neg A(pUq))$
$AG(r \rightarrow (\neg q \wedge EXr)) \rightarrow (r \rightarrow \neg AFq)$
$AG(r \rightarrow (\neg q \wedge (p \rightarrow AXr))) \rightarrow (r \rightarrow \neg E(pUq))$
$AG(r \rightarrow (\neg q \wedge AXr)) \rightarrow (r \rightarrow \neg EFq)$
$AG(p \rightarrow q) \rightarrow (EXp \rightarrow Exq)$

第二部分：命题逻辑推理系统

如果 p 是重言式，则 p 是公理。
如果 $\vdash p \rightarrow q$ 且 $\vdash p$，则 $\vdash q$（MP 规则）。

第三部分：时序推理规则

如果 $\vdash p$，则 $\vdash AGp$（泛化规则）。

可判定性

CTL 公式的可满足性是可判定的，其判定复杂性为 EXPTIME 完全。

§3.2.2 CTL 公式与状态集合

由于 CTL 公式在状态上解释，给定一个 Kripke 结构 $M = \langle S, \Delta, I\rangle$。给定的一个公式 φ。S 的子集 $\{s \in S \mid M, s \models \varphi\}$ 记为 $[[\varphi]_M]$。

设 $A \subseteq S$。定义

$$ex(A) = \{s \in S \mid \exists s' \in S, (s, s') \in \Delta \wedge s' \in A\}$$

我们有以下等式：

$[[p]_M]$	$=$	$\{s \mid p \in L(s)\}$
$[[\neg p]_M]$	$=$	$S \setminus [[p]_M]$
$[[p \vee q]_M]$	$=$	$[[p]_M] \cup [[q]_M]$
$[[p \wedge q]_M]$	$=$	$[[p]_M] \cap [[q]_M]$
$[[EXp]_M]$	$=$	$ex([[p]_M])$

对于 EG 和 EU 。我们有以下等式。

$$\begin{aligned} EGp &\equiv p \wedge EXEGp \\ E(pUq) &\equiv q \vee (p \wedge EXE(pUq)) \end{aligned}$$

即

$$\begin{aligned} [[EGp]_M] &= [[p]_M] \cap ex([[EGp]_M]) \\ [[E(pUq)]_M] &= [[q]_M] \cup ([[p]_M] \cap ex([[E(pUq)]_M]) \end{aligned}$$

为方便起见，我们直接将 p, q 看成 S 的子集，模态算子和逻辑联接符看成是 2^S 上的函数。

.

这样， EGp 和 $E(pUq)$ 就分别是 $\tau_1(Z) = p \wedge EX(Z)$ 和 $\tau_2(Z) = q \vee (p \wedge EX(Z))$ 的不动点。

.

且 EGp 是 τ_1 的最大不动点，即 $EGp = \nu Z(p \wedge EXZ)$ ，$E(pUq)$ 是 τ_2 的最小不动点，即 $E(pUq) = \mu Z(q \vee (p \wedge EXZ))$ 。

证明： ...

.

类似地，其它模态算子也可以用不动点表示。

AFp	$=$	$\mu Z(p \vee AXZ)$
AGp	$=$	$\nu Z(p \wedge AXZ)$
EFp	$=$	$\mu Z(p \vee EXZ)$
EGp	$=$	$\nu Z(p \wedge EXZ)$
$A(pUq)$	$=$	$\mu Z(q \vee (p \wedge AXZ))$
$A(pRq)$	$=$	$\nu Z(q \wedge (p \vee AXZ))$
$E(pUq)$	$=$	$\mu Z(q \vee (p \wedge EXZ))$
$E(pRq)$	$=$	$\nu Z(q \wedge (p \vee EXZ))$

例子： 证明 $AFp = \mu Z(p \vee AXZ)$

.

设 φ 为 CTL 公式， M 为 Kripke 结构。 $M \models \varphi$ 当且仅当 $I \subseteq [[\varphi]_M]$ 。

例子： 简化自动售茶机模型

习题： 用简化自动售茶机模型计算 $[[A(q_0 U q_2)]]$ 、$[[EG(q_0 \vee q_2)]]$ 并讨论该模型是否满足这些公式。

§3.2.3 CTL 公式在 Kripke 结构中的推理与交错 Büchi 自动机

由于 CTL 公式在状态上解释，给定一个 Kripke 结构 M 和一个状态 s，我们可以根据 s 或其后续状态的情况来判断 s 是否满足一个公式。

CTL 公式在 Kripke 结构中的推理

我们定义三种推理规则

$$\frac{\top}{A}, \quad \vee\frac{A_1 \quad \cdots \quad A_n}{A}, \quad \wedge\frac{A_1 \quad \cdots \quad A_n}{A}$$

分别表示不需要前提即可得出结论 A，有 n 个前提中的一个前提即可得出结论 A，有 n 个前提中的所有前提则可得出结论 A。我们考虑 CTL 公式的 NNF 范式。

$$\frac{\top}{s \vdash_M p} \quad p \in L(s) \qquad \frac{\top}{s \vdash_M \neg p} \quad p \in AP \setminus L(s)$$

$$\wedge\frac{s \vdash_M p \quad s \vdash_M q}{s \vdash_M p \wedge q} \qquad \vee\frac{s \vdash_M p \quad s \vdash_M q}{s \vdash_M p \vee q}$$

$$\wedge\frac{s_1 \vdash_M p \quad \cdots \quad s_n \vdash_M p}{s \vdash_M AXp} \quad (s_1, ..., s_n) = \{s' \mid (s, s') \in \Delta\}$$

$$\vee\frac{s_1 \vdash_M p \quad \cdots \quad s_n \vdash_M p}{s \vdash_M EXp} \quad (s_1, ..., s_n) = \{s' \mid (s, s') \in \Delta\}$$

$$\vee\frac{s \vdash_M q \quad s \vdash_M p \wedge AXA(pUq)}{s \vdash_M A(pUq)} \qquad \vee\frac{s \vdash_M q \quad s \vdash_M p \wedge EXE(pUq)}{s \vdash_M E(pUq)}$$

$$\wedge\frac{s \vdash_M q \quad s \vdash_M p \vee AXA(pRq)}{s \vdash_M A(pRq)} \qquad \wedge\frac{s \vdash_M q \quad s \vdash_M p \vee EXE(pRq)}{s \vdash_M E(pRq)}$$

交错 Büchi 自动机

定义 $sf(\varphi)$ 为公式的集合如下。$a \in sf(\varphi)$ 当且仅当 a 在 φ 中出现或 $a = AXA(bRc)$ 且 $A(bRc)$ 在 φ 中出现或 $a = EXE(bRc)$ 且 $E(bRc)$ 在 φ 中出现。

给定 Kripke 结构 $M\langle S, \Delta, I\rangle$ 和公式 φ。构造交错 Büchi 自动机 $\mathcal{B}(M, \varphi) = \langle \Sigma, S', \Delta', I', F\rangle$ 如下：

Σ	$=$	$\{a\}$
S'	$=$	$\{s \vdash_M \psi \mid s \in S, \psi \in sf(\varphi)\}$
$\Delta'(t,a)$	$=$	$\{\{t_1,...,t_n\}\}$ 对应 $\wedge$- 规则
$\Delta'(t,a)$	$=$	$\{\{t_1\},...,\{t_n\}\}$ 对应 $\vee$- 规则
$\Delta'(t,a)$	$=$	$\{\{\}\}$ 若 $t=\top$
$\Delta'(t,a)$	$=$	$\{\{t\}\}$ 若 t 是叶结点且 $t \neq \top$
I'	$=$	$\{s \vdash_M \varphi \mid s \in I\}$
F	$=$	$\{s \vdash_M E(pRq) \mid s \in S, E(pRq) \in sf(\varphi)\} \cup \{s \vdash_M A(pRq) \mid s \in S, A(pRq) \in sf(\varphi)\}$

则 $\mathcal{B}(M,\varphi)$ 可以看成是 $M \models \varphi$ 的证明结构。$\mathcal{B}(M,\varphi)$ 为空当且仅当 $M \models \neg\varphi$ 。

对于任意给定 Kripke 结构 $\langle S,\Delta,I\rangle$ 和 NNF 范式的 CTL 公式 φ ，我们能够构造一个相应的交错 Büchi 自动机 $\langle \Sigma, S', \Delta', I', F\rangle$ 且 $|S'| \leq |S| \cdot |sf(\varphi)|$ 。

§3.3 计算树逻辑 CTL*

由于 CTL 中描述分枝情况和描述状态的前后关系的算子成对出现，在一定程度上限制了 CTL 的表达能力。我们可以将其拆开使用。这个逻辑记作 CTL* 。给定一个原子命题集合 AP 。CTL* 公式的集合由以下语法给出。

$$\phi ::= p|\neg\phi|\phi\wedge\phi|\phi\vee\phi|X\,\phi|F\,\phi|G\,\phi|\phi U\,\phi|\phi R\phi|A\phi|E\phi$$

其中 p 为 AP 中任意命题。我们可以将 CTL* 中的公式分为状态公式和路径公式。

若 $p \in AP$,	则 p 是状态公式
若 φ 和 ψ 是状态公式,	则 $\neg\varphi, \varphi\vee\psi, \varphi\wedge\psi$ 是状态公式
若 φ 是路径公式,	则 $E\varphi$ 和 $A\varphi$ 是状态公式
若 φ 是路径公式,	则 φ 是路径公式
若 φ 和 ψ 是路径公式,	则 $\neg\varphi, \varphi\vee\psi, \varphi\wedge\psi, X\varphi,\ F\varphi, G\varphi, \varphi U\psi, \varphi R\psi$ 是路径公式

CTL* 公式在一个树状结构或 Kripke 结构上解释。设 S 为状态集合。设 $M=\langle S,R,I,L\rangle$ 是原子命题集 AP 上的 Kripke 结构。 CTL* 公式的语义如下：

$M, s \models p$	若 $p \in AP$ 且 $p \in L(s)$
$M, s \models \neg\varphi$	若 $M, s \not\models \varphi$
$M, s \models \varphi \vee \psi$	若 $M, s \models \varphi$ 或 $M, s \models \psi$
$M, s \models \varphi \wedge \psi$	若 $M, s \models \varphi$ 且 $M, s \models \psi$
$M, s \models A\varphi$	若对于所有 M 中以 s 为起点的路径 π： $M, \pi \models \varphi$
$M, s \models E\varphi$	若存在 M 中以 s 为起点的路径 π： $M, \pi \models \varphi$
$M, \pi \models p$	若 $p \in AP$ 且 $M, \pi_0 \models p$
$M, \pi \models A\varphi$	若 $M, \pi_0 \models A\varphi$
$M, \pi \models E\varphi$	若 $M, \pi_0 \models E\varphi$
$M, \pi \models \neg\varphi$	若 $M, \pi \not\models \varphi$
$M, \pi \models \varphi \vee \psi$	若 $M, \pi \models \varphi$ 或 $M, \pi \models \psi$
$M, \pi \models \varphi \wedge \psi$	若 $M, \pi \models \varphi$ 且 $M, \pi \models \psi$
$M, \pi \models X\varphi$	若 $M, \pi^1 \models \varphi$
$M, \pi \models G\psi$	若 $\exists i \geq 0, M, \pi^i \models \psi$
$M, \pi \models F\varphi$	若 $\exists i \geq 0, M, \pi^i \models \varphi$
$M, \pi \models \varphi U \psi$	若 $\exists i \geq 0, M, \pi^i \models \psi$ 且 $\forall 0 \leq j < i, M, \pi^j \models \varphi$
$M, \pi \models \varphi R \psi$	若 $\forall i \geq k$, 若 $\forall 0 \leq j < i, M, \pi^j \not\models \varphi$ 则 $M, \pi^i \models \psi$

设 φ 为 CTL* 状态公式， M 为 Kripke 结构。

$$M \models \varphi$$

当且仅当

对所有 $s \in I$， $M, s \models \varphi$ 。

例子： $E(GFp)$ 不同于 $EGEFp$ 。

习题： 讨论 $A((pUr) \vee (qUr))$ 与 $A(pUr) \vee A(qUr)$ 和 $A((p \vee q)Ur)$ 的关系及满足 $A((pUr) \vee (qUr))$ 的结构的特征。

.

PLTL 和 CTL 是 CTL* 的子集。设 PLTL 、CTL 和 CTL* 都在 Kripke 结构上解释。一个 PLTL 公式 φ 等价于一个 CTL* 公式 $A\varphi$ 。这样 PLTL 公式就可以看成是状态公式。

.

在这样的解释下，我们可以比较 PLTL 、 CTL 和 CTL* 的状态公式。

.

PLTL 和 CTL 互不录属且 CTL* 大于 PLTL 和 CTL 的并集。

CTL 公式 $AGEFp$ 不能用 PLTL 表示。
PLTL 公式 $F(p \wedge Xp)$ 不能用 CTL 表示。
CTL* 公式 $E(GFp)$ 不能用 CTL 表示，不能用 PLTL 表示。

§3.4 μ- 演算

由关于 CTL 的讨论我们知道 CTL 公式可以用模态算子 AX,EX, 命题变量和不动点表示。在 CTL 公式的这种表示中，命题变量可以出现在 AX, EX, μ, ν 之后。将这个约束去掉，我们可以得到一个表达能力更强的逻辑。称为 μ- 演算。

.

给定一个原子命题集合 AP ，一个变量集合 V 。一种简单的 μ- 演算的公式的集合可由以下语法给出。

$$\phi ::= p \mid X \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \langle . \rangle \phi \mid [.]\phi \mid \mu X.\phi \mid \nu X.\phi$$

其中 p 为 AP 中任意命题，X 为变量，$\mu X.\phi$ 和 $\nu X.\phi$ 的 ϕ 中不受囿的 X 必须在偶数个 $\neg$ 符号的作用范围之下。

带有动作描述的 μ- 演算

对前述简单 μ- 演算做扩充，增加动作的描述，我们可以得到一种能够描述动作的时序关系的时序逻辑。

.

给定一个原子命题集合 AP ，一个变量集合 V ，和一个动作集合 A 。 μ- 演算公式的集合由以下语法给出。

$$\phi ::= p \mid X \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \langle a \rangle \phi \mid [a]\phi \mid \mu X.\phi \mid \nu X.\phi$$

其中 p 为 AP 中任意命题， X 为变量， a 为动作，并且 $\mu X.\phi$ 和 $\nu X.\phi$ 的 ϕ 中不受囿的 X 必须在偶数个 $\neg$ 符号的作用范围之下。

μ- 演算公式在 LTS 上解释。设 $M = \langle \Sigma, S, \Delta, I, L \rangle$ 为原子命题集 AP 上的 LTS， $e : V \to 2^S$ 为变量到 S 子集的赋值。 μ- 演算公式的语义如下：

$[[p]_M]e$	$=$	$\{s \mid p \in L(s)\}$
$[[X]_M]e$	$=$	$e(X)$
$[[\neg\phi]_M]e$	$=$	$S \setminus [[\phi]_M]e$
$[[\phi_1 \wedge \phi_2]_M]e$	$=$	$[[\phi_1]_M]e \cap [[\phi_2]_M]e$
$[[\phi_1 \vee \phi_2]_M]e$	$=$	$[[\phi_1]_M]e \cup [[\phi_2]_M]e$
$[[\langle a \rangle \phi]_M]e$	$=$	$\{s \mid \exists s'.s \xrightarrow{a} s' \wedge s' \in [[\phi]_M]e\}$
$[[[a]\phi]_M]e$	$=$	$\{s \mid \forall s'.s \xrightarrow{a} s' \Rightarrow s' \in [[\phi]_M]e\}$
$[[\mu X.\phi]_M]e$	$=$	$\cap\{S' \subseteq S \mid [[\phi]_M]e[X/S'] \subseteq S'\}$
$[[\nu X.\phi]_M]e$	$=$	$\cup\{S' \subseteq S \mid S' \subseteq [[\phi]_M]e[X/S']\}$

.

所有变量都是受囿变量的公式称为闭公式。闭公式的语义不受 e 的影响。设 M 为 LTS， φ 为闭公式。 φ 在 M 中的语义可写为 $[[\varphi]_M]$ 。

$$M \models \varphi$$

当且仅当

$$I \subseteq [[\varphi]_M] \text{ 。}$$

例子： 自动售茶机

习题： 以自动售茶机为例，计算 $\mu X.(q_1 \vee \langle 2\rangle X)$ 并讨论系统是否满足这个公式。

模态算子的对偶关系与 NNF 范式

逻辑连接符 $\neg$ 只出现在命题前面的公式称为 NNF 范式。每个 μ- 演算公式等价于一个 μ- 演算的 NNF 范式。一个 μ- 演算的 NNF 范式可应用以下对偶关系构造。

$$\begin{array}{lcl} [a]\phi & \equiv & \neg\langle a\rangle\neg\phi \\ \mu X.\phi & \equiv & \neg\nu X.\neg\phi[X/\neg X] \end{array}$$

μ- 演算公式在 Kripke 结构中的推理

与 CTL 公式类似，给定一个 Kripke 结构 M 和一个状态 s，我们可以根据 s 或其后续状态的情况来判断 s 是否满足一个（ NNF 范式）的 μ- 演算公式。

$$\frac{\top}{s \vdash_M p} \quad p \in L(s) \qquad\qquad \frac{\top}{s \vdash_M \neg p} \quad p \in AP \setminus L(s)$$

$$\wedge\frac{s \vdash_M p \quad s \vdash_M q}{s \vdash_M p \wedge q} \qquad\qquad \vee\frac{s \vdash_M p \quad s \vdash_M q}{s \vdash_M p \vee q}$$

$$\vee\frac{s_1 \vdash_M p \quad \cdots \quad s_n \vdash_M p}{s \vdash_M \langle a\rangle p} \quad \{s_1, ..., s_n\} = \{s' \mid (s, a, s') \in \Delta\}$$

$$\wedge\frac{s_1 \vdash_M p \quad \cdots \quad s_n \vdash_M p}{s \vdash_M [a]p} \quad \{s_1, ..., s_n\} = \{s' \mid (s, a, s') \in \Delta\}$$

$$\frac{s \vdash_M \varphi[\mu X.\varphi/X]}{s \vdash_M \mu X.\varphi} \qquad\qquad \frac{s \vdash_M \varphi[\nu X.\varphi/X]}{s \vdash_M \nu X.\varphi}$$

CTL 公式到 μ- 演算公式的转换

由于 CTL 公式不牵涉到动作的描述，可以假定 CTL 所描述系统中，只用同一个动作，记作 a 。这样，可以用以下规则将 CTL 公式转换到 μ- 演算公式。

$$\begin{array}{lcl} T(p) & = & p \\ T(\neg\varphi) & = & \neg T(\varphi) \\ T(\varphi \wedge \psi) & = & T(\varphi) \wedge T(\psi) \\ T(EX\varphi) & = & \langle a\rangle T(\varphi) \\ T(E(\varphi U\psi) & = & \mu Y.(T(\psi) \vee (T(\varphi) \wedge \langle a\rangle Y)) \\ T(EG\varphi) & = & \nu Y.(T(\varphi) \wedge \langle a\rangle Y) \end{array}$$

§3.5 线性 μ- 演算 (νTL)

μ- 演算亦可在线性结构上定义。在线性结构上，$[.]$ 和 $\langle . \rangle$ 就归结为简单的下一时刻算子，依然记作 O 。给定一个原子命题集合 AP ，一个变量集合 V 。 νTL 公式的集合由以下语法给出。

$$\phi ::= p \mid X \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid O\phi \mid \mu X.\phi \mid \nu X.\phi$$

其中 p 为 AP 中任意命题， X 为变量，并且 $\mu X.\phi$ 和 $\nu X.\phi$ 的 ϕ 中不受囿的 X 必须在偶数个 $\neg$ 符号的作用范围之下。

νTL 公式可在 AP 上的 KS 上解释。设 R 是 KS 上以任意状态为起点的所有运行的集合。设 M 为 AP 上的 KS ， $e : V \to 2^R$ 为变量到 R 子集的赋值。 νTL 公式的语义如下：

$$\begin{array}{lcl}
[[p]_M]e & = & \{\pi \mid p \in L(\pi_0)\} \\
[[X]_M]e & = & e(X) \\
[[\neg\phi]_M]e & = & R \setminus [[\phi]_M]e \\
[[\phi_1 \wedge \phi_2]_M]e & = & [[\phi_1]_M]e \cap [[\phi_2]_M]e \\
[[\phi_1 \vee \phi_2]_M]e & = & [[\phi_1]_M]e \cup [[\phi_2]_M]e \\
[[O\phi]_M]e & = & \{\pi \mid \pi_1 \in [[\phi]_M]e\} \\
[[\mu X.\phi]_M]e & = & \cap\{R' \subseteq R \mid [[\phi]_M]e[X/R'] \subseteq R'\} \\
[[\nu X.\phi]_M]e & = & \cup\{R' \subseteq R \mid R' \subseteq [[\phi]_M]e[X/R']\}
\end{array}$$

设 M 为 KS ， φ 为闭公式。 φ 在 M 中的语义可写为 $[[\varphi]_M]$ 。 $M, \pi \models \varphi$ 当且仅当 $\pi \in [[\varphi]_M]$ 。记 M 的运行集合为 $[[M]]$ 。

$$M \models \varphi$$

当且仅当

$$[[M]] \subseteq [[\varphi]_M] \text{ 。}$$

若将 νTL 公式直接在线性结构中解释，则 νTL 公式有以下语义。 设 $M = (\pi, L)$ 为一线性结构，其中 $\pi = \pi_0\pi_1 \cdots$ 为状态序列， $L : \{\pi_0, \pi_1, ...\} \to 2^{AP}$ 为状态集到命题幂集的映射。设 $e : V \to 2^N$ 为变量到 N 子集的赋值。 νTL 公式的语义如下：

$$\begin{array}{lcl}
[[p]_M]e & = & \{i \in N \mid p \in L(\pi_i)\} \\
[[X]_M]e & = & e(X) \\
[[\neg\phi]_M]e & = & N \setminus [[\phi]_M]e \\
[[\phi_1 \wedge \phi_2]_M]e & = & [[\phi_1]_M]e \cap [[\phi_2]_M]e \\
[[\phi_1 \vee \phi_2]_M]e & = & [[\phi_1]_M]e \cup [[\phi_2]_M]e \\
[[O\phi]_M]e & = & \{i \in N \mid i+1 \in [[\phi]_M]e\} \\
[[\mu X.\phi]_M]e & = & \cap\{S' \subseteq N \mid [[\phi]_M]e[X/S'] \subseteq S'\} \\
[[\nu X.\phi]_M]e & = & \cup\{S' \subseteq N \mid S' \subseteq [[\phi]_M]e[X/S']\}
\end{array}$$

$M \models \varphi$ 当且仅当 $0 \in [[\varphi]_M]$ 。

LTL 公式到 νTL 公式的转换

LTL 公式可以看成是 νTL 的一个子类，可以用以下规则将 LTL 公式转换到 νTL 公式。

$$\begin{array}{lcl} T(p) & = & p \\ T(\neg\varphi) & = & \neg T(\varphi) \\ T(\varphi \wedge \psi) & = & T(\varphi) \wedge T(\psi) \\ T(O\varphi) & = & OT(\varphi) \\ T(\varphi U \psi) & = & \mu Y.(T(\psi) \vee (T(\varphi) \wedge OY)) \end{array}$$

§3.6 交错时序逻辑 (ATL)

交错时序逻辑是一种能够描述动作主体合作完成任务的逻辑。给定一个原子命题集合 AP 。一个动作主体的集合 Σ 。 ATL 公式的集合由以下语法给出。

$$\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \langle\langle A\rangle\rangle O\phi \mid \langle\langle A\rangle\rangle \Box\phi \mid \langle\langle A\rangle\rangle \phi U \phi$$

其中 p 为 AP 中任意命题、 $A \subseteq \Sigma$ 为 Σ 的子集。

ATL 公式在 ATS 上解释。设 $M = \langle \Sigma, S, \Delta, I, L\rangle$ 为 AP 上的 ATS 。给定 $a \in \Sigma$ 。一个 a 的策略 f_a 是 $S^+ \to 2^S$ 的一个函数，满足对所有 $\lambda \in S^*$ 和所有 $s \in S$ ，

$$f(\lambda \cdot s) \in \Delta(s, a)$$

给定 $A \subseteq \Sigma$ ， $F_A = \{f_a \mid a \in A\}$ ，在状态 s 应用策略 F_A 的结果 $out(s, F_A)$ 满足条件：

$$s_0 s_1 s_2 \cdots \in out(s, F_A)$$

当前仅当 $s = s_0$ 且对所有 i, $s_{i+1} \in \cap_{a\in A} f_a(s_0 s_1 \cdots s_i)$ 。

.

ATL 公式的语义如下：

$M, s \models p$	若 $p \in AP$ 且 $p \in L(s)$
$M, s \models \neg\varphi$	若 $M, s \not\models \varphi$
$M, s \models \varphi \vee \psi$	若 $M, s \models \varphi$ 或 $M, s \models \psi$
$M, s \models \langle\langle A\rangle\rangle O\varphi$	若存在 F_A 且对所有 $\lambda \in out(s, F_A)$ ， $M, \lambda[1] \models \varphi$
$M, s \models \langle\langle A\rangle\rangle \Box\varphi$	若存在 F_A 且对所有 $\lambda \in out(s, F_A)$ ，对所有 $i \geq 0$ ， $M, \lambda[i] \models \varphi$
$M, s \models \langle\langle A\rangle\rangle \varphi U \psi$	若存在 F_A 且对所有 $\lambda \in out(s, F_A)$ ， 存在 $i \geq 0$ ， $\lambda[i] \models \psi$ 且对所有 $0 \leq j < i$ ， $M, \lambda[j] \models \varphi$

除了以上基本算子外，还可以定义其它算子，如：

$$\begin{array}{lcl} \Diamond\varphi & = & true U \varphi \\ [[A]]O\varphi & = & \neg\langle\langle A\rangle\rangle O\neg\varphi \\ [[A]]\Box\varphi & = & \neg\langle\langle A\rangle\rangle \Diamond\neg\varphi \end{array}$$

例子： 火车进站控制模型

$$\langle\langle\rangle\rangle\Box(og \wedge \neg gr \rightarrow \langle\langle ctr\rangle\rangle\Box og)$$

$$\langle\langle\rangle\rangle\Box(og \rightarrow \langle\langle ctr, train\rangle\rangle\Diamond ig)$$

§3.7 说明

本章主要介绍程序逻辑。线性时序逻辑部分可参考 [Pel01] 。一阶线性时序逻辑部分可参考 [Pel01, MP83] 。线性 μ- 演算部分可参考 [Kai95, BEM96] 。分枝时序逻辑部分可参考 [CGP99, Pel01] 。模态 μ- 演算部分可参考 [CGP99, BC96] 。交互逻辑部分部分可参考 [AKH97] 。

参考文献

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

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

[Kai95] Roope Kaivola. Axiomatising Linear Time μ-Calculus. CONCUR'95. LNCS 962:423-437.

[BEM96] Julian Bradfield, Javier Esparza and Angelika Mader. An Effective Tableau System for the Linear Time μ-Calculus. ICALP'96. LNCS 1099:98-109.

[MP83] Zohar Manna and Amir Pnueli. How to cool a temporal proof system for your pet language. The 10th ACM Symposium on Principles of Programming Languages 141-154. 1983.

[AKH97] Rajeev Alur, Thomas A. Henzinger and Orna Kupferman. Alternating-Time Temporal Logic. COMPOS 1997: 23-60.

[BC96] Girish Bhat and Rance Cleaveland. Efficent Local Model-Checking for Fragments of the Modal mu-Calculus. TACAS 1996: 107-126.