华东师范大学
软硬件协同设计技术与应用教育部工程研究中心
URL—http://faculty.ecnu.edu.cn/chenyixiang
yxchen@sei.ecnu.edu.cn

时态逻辑系统

陈仪香,吴恒洋

MoE Engineering Center for Software/Hardware Co-Design Technology and Application
Software Engineering Institute
East China Normal University(ECNU)
Shanghai, China

2018级研究生软件工程理论课程，2018年4月

时态逻辑系统

- 表达/刻画逻辑中的时态性：一个公式不是静态地取真值，而是动态地取真值。
- 一个公式可能在某些状态是真的，而在其它状态是假的。
- 真值的静态性变成动态性。
- 公式随着系统的状态演化而改变真值。

模型检测

时态逻辑系统可用于模型检测。

- 模型通常是迁移系统/有限自动机,它描述了状态迁移过程，反映状态的演化，而公式是时态逻辑公式ϕ。
- 模型检测的目的是表明模型$\mathcal{M}$满足公式ϕ，即$\mathcal{M} \models \phi$。
- 通常实现模型检测，需要做下面三件事情:
  - 建立模型$\mathcal{M}$,
  - 编写公式ϕ,
  - 运行模型检测器，输入$\mathcal{M}$和ϕ,
- 模型检测器将输出Yes若$\mathcal{M} \models \phi$成立,否则输出No。

时态逻辑分类

- 线性时态逻辑系统LTL：时间是按照线性进行迁移的
- 计算树逻辑系统CTL：时间是按照树进行迁移的.

线性时态逻辑系统LTL

- 引入连接词表示时间：X, F, G, U, W, R
  - X—Next 下一个状态，
  - F—某个Future 状态，
  - G—所有将来的状态(Globally)，
  - U—Until 直到
  - W—Weak-Until,弱直到
  - R—Release, 解释,释放
- 引入原子公式Atoms：$p, q, e, \cdots, p_1, p_2, \cdots$
  如: 打印机Q_5是忙的，进程3259在悬挂，记录$R1$的内容是整数值6，数据的长度是99,等
- 计算路,也叫状态序列，简称路

Back
Close

LTL的语法

定义 LTL的公式

公式ϕ定义为

$$\begin{aligned}\phi \ ::= \ & \top \mid \bot \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \rightarrow \phi \\ & (X\phi) \mid (F\phi) \mid (G\phi) \mid (\phi U \phi) \mid (\phi W \phi) \mid (\phi R \phi)\end{aligned}$$

例子: $(F(p \rightarrow Gr) \vee ((\neg q)Up))$,画出Parse tree; 非法公式: Ur, pGq.

LTL的语法

1. 在任何状态下，若有一个请求出现,那么这个请求将会被接受.
   G (请求出现$\longrightarrow$ F接受)
2. 某个进程往往在每个计算路上被无限次地激活.
   GF激活
3. 一部上升的电梯在第二层时不会改变上升方向直到第5层楼,若电梯内有人要到第5层楼.
   G (2层$\wedge$向上$\wedge$ 有人要到5层$\longrightarrow$(向上方向U5层楼))
4. 已经到达了开始状态，但准备工作还没有做好是不可能的.
   $G\neg$(开始了$\wedge\neg$ 准备)。

LTL的语义

迁移系统(Transition System)：通过状态(静态结构)和迁移(动态结构)来为系统提供模型.

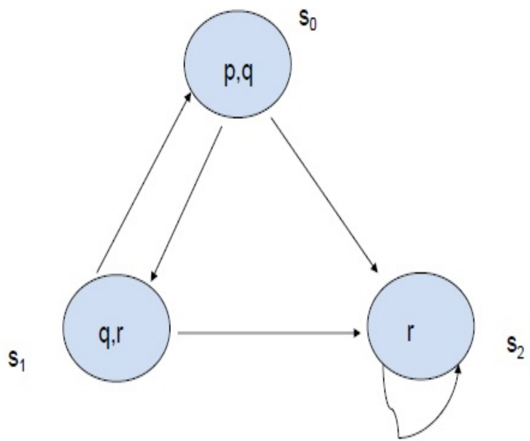
定义　迁移系统

迁移系统$\mathcal{M} = (S, \to, L)$是由下面三部分组成:

- S是状态集
- $\to$是S上的二元关系,称为迁移关系,使得$\forall s \in S$,都有$s' \in S$且$s \to s'$, 即$\to$ 是S上的连续关系
- 标号函数$L : S \to \mathcal{P}(Atoms)$

注：(1) 迁移系统是一种特殊的Kripke模型。
(2) 迁移系统可直接称为模型.
(3) 例子：

例子



Back
Close

LTL的语义

定义　路

模型$\mathcal{M}=(S,\rightarrow,L)$的路是指$S$中的无限状态序列$s_1,s_2,\cdots,s_n,\cdots$使得$\forall i\geq 1, s_i\rightarrow s_{i+1}$.

通常将路写成：$s_1\rightarrow s_2\rightarrow\cdots$，并用$\pi$表示一条路.

注：(1) π^i表示从状态s_i开始的路. 请写出上例中的一些路.

(2)计算路的展开(unwinding)。

LTL的语义

定义　路满足公式
给定模型$\mathcal{M} = (S, \rightarrow, L)$以及路$\pi = s_1 \rightarrow s_2 \rightarrow \cdots$. 定义$\pi$满足公式$\phi$，记作$\pi \models \phi$, 归纳如下:

1. $\pi \models \top$
2. $\pi \not\models \bot$
3. $\pi \models p$ 当且仅当$p \in L(s_1)$
4. $\pi \models \neg\phi$ 若$\pi \not\models \phi$
5. $\pi \models \phi \wedge \psi$ 若$\pi \models \phi$且$\pi \models \psi$.
6. $\pi \models \phi \vee \psi$ 若$\pi \models \phi$或$\pi \models \psi$.
7. $\pi \models \phi \rightarrow \psi$ 若$\pi \models \phi$则$\pi \models \psi$
8. $\pi \models X\phi$ 若$\pi^2 \models \phi$
9. $\pi \models G\phi$ 若$\forall i \geq 1, \pi^i \models \phi$
10. $\pi \models F\phi$ 若$\exists i \geq 1$使得$\pi^i \models \phi$

11. $\pi \models \phi U \psi$ 若$\exists i \geq 1$使得$\pi^i \models \psi$且对于所有的$j = 1, 2, \cdots, i-1$都有$\pi^j \models \phi$.

12. $\pi \models \phi W \psi$ 若或者$\exists i \geq 1$使得$\pi^i \models \psi$且对于所有的$j = 1, 2, \cdots, i-1$都有$\pi^j \models \phi$或者对于所有的$k \geq 1$都有$\pi^k \models \phi$.

13. $\pi \models \phi R \psi$ 若或者$\exists i \geq 1$使得$\pi^i \models \phi$且对于所有的$j = 1, 2, \cdots, i$都有$\pi^j \models \psi$或者对于所有的$k \geq 1$都有$\pi^k \models \psi$.

图示: $\pi \models \phi$.

- 原子命题a:

$$\begin{array}{ccccccccccc} \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \cdots \\ a & & \text{任意} & & \text{任意} & & \cdots & & & & \end{array}$$

- Xa:

$$\begin{array}{ccccccccccc} \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \cdots \\ \text{任意} & & a & & \text{任意} & & \text{任意} & & \cdots & & \end{array}$$

- aUb:

$$\begin{array}{ccccccccccc} \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \cdots \\ a, \neg b & & a, \neg b & & b & & \text{任意} & & \cdots & & \end{array}$$

($\neg b$ 可以不标.)

- aRb:

$$\begin{array}{ccccccccccc} \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \cdots \\ b & & b & & a, b & & \text{任意} & & \cdots & & \end{array}$$

  或者

$$\begin{array}{ccccccccccc} \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \bigcirc & \to & \cdots \\ b & & b & & b & & \cdots & & & & \end{array}$$

- aWb: 请大家画出

定义　状态满足公式

设$\mathcal{M}=(S,\to,L)$是一个模型,$s\in S$, ϕ是一个LTL公式，若对$\mathcal{M}$的从s出发的每条路π都有$\pi\models\phi$，则称状态s满足ϕ，记作$\mathcal{M},s\models\phi$，或$s\models\phi$.

例子: 接前面的例子, 考察迁移系统中，状态满足哪些逻辑公式.

- $s_0 \models (p \wedge q)$;
- $s_0 \models \neg r$;
- $s_0 \models \top$;
- $s_0 \models Xr$;
- $s_0 \not\models X(q \wedge r)$;
- $s_0 \models G\neg(p \wedge r)$;
- $s_2 \models Gr$;
- $s \models F(\neg q \wedge r) \rightarrow FGr$;
- $s_0 \not\models GFp$; 路$s_0 \rightarrow s_1 \rightarrow s_0 \rightarrow s_1 \rightarrow \cdots$满足该公式, 但路$s_0 \rightarrow s_2 \rightarrow s_2 \rightarrow s_2 \rightarrow \cdots$不满足.
- $s_0 \models GFp \rightarrow GFr$, 但$s_0 \not\models GFr \rightarrow GFp$.

语义等价

定义　语义等价$\phi \equiv \psi$

设ϕ, ψ是LTL公式,若对于所有的模型$\mathcal{M}$以及$\mathcal{M}$中的所有的路π都有$\pi \models \phi$ 当且仅当$\pi \models \psi$, 则称ϕ 与ψ 是语义等价的,记作$\phi \equiv \psi$.

定理　语义等价等价刻画 设ϕ, ψ是LTL公式,它们是语义等价的, 当且仅当若对于所有的模型$\mathcal{M}$以及$\mathcal{M}$中的所有的状态s都有$s \models \phi$ 当且仅当$s \models \psi$.

语义等价

定理 下面各条成立

de Margan律	$\phi \wedge \psi \equiv \neg(\neg\phi \vee \neg\psi)$
	$\phi \vee \psi \equiv \neg(\neg\phi \wedge \neg\psi)$
幂等律	$\neg\neg\phi \equiv \phi$
对偶性	$G\phi \equiv \neg F\neg\phi$
	$F\phi \equiv \neg G\neg\phi$
	$\phi U\psi \equiv \neg(\neg\phi R\neg\psi)$
	$\phi R\psi \equiv \neg(\neg\phi U\neg\psi)$
自对偶性	$X\phi \equiv \neg X\neg\phi$
分配性	$F(\phi \vee \psi) \equiv F\phi \vee F\psi$
	$G(\phi \wedge \psi) \equiv G\phi \wedge G\psi$

思考: $F(\phi \wedge \psi) \equiv F\phi \wedge F\psi$ (?); $G(\phi \vee \psi) \equiv G\phi \vee G\psi$ (?)

连接词的充分性

定理 连接词相互定义

$$
\begin{aligned}
&F\phi \equiv \top U\phi \\
&G\phi \equiv \bot R\phi \\
&\phi W\psi \equiv \phi U\psi \vee G\phi \\
&\phi W\psi \equiv \psi R(\phi \vee \psi) \\
&\phi R\psi \equiv \psi W(\phi \wedge \psi) \\
&\phi U\psi \equiv \phi W\psi \wedge F\psi
\end{aligned}
$$

连接词的充分性：
$\{U, X\}, \{R, X\}, \{W, X\}$

综上, LTL可以简便的表示如下:

$$\phi ::= \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid X\phi \mid \phi U\phi$$

与$\Box$, $\Diamond$的关系

定义:

$$\Diamond\,\phi \stackrel{def}{=} \top U \phi \tag{1}$$

与

$$\Box\phi \stackrel{def}{=} \neg\Diamond\neg\phi \tag{2}$$

进一步:

$$\Diamond\phi = F\phi, \quad \Box\phi = G\phi.$$

$\Diamond$:最终,将来; $\Box$: 总是, 从现在起永远.

模型检测:互斥

当并发进程共享资源(如磁盘上的某个文件, 或数据库登陆), 通常要求两个进程不能同时获取进入. 进程不能同时编辑相同的文件.

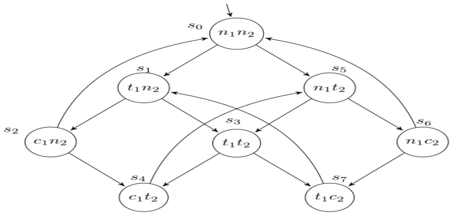
需要给定某个临界区, 在任意时刻只安排一个进程在临界区.

问题：如何设计协议以确定在任意时刻，哪个进程被允许进入临界区.

1. n:表示在非临界状态

2. t: 试图进入临界状态

3. c: 已在临界区的状态



A model for mutual exclusion.

exclusion.png

1.安全性: 任何时候最多一个进程在临界区

2.活性: 任何进程只要要求进入临界区,最终会进入.

3.非阻塞性: 任何进程总能要求进入临界区.

1. 安全性:

$$G\neg(c_1 \wedge c_2)$$

2.活性:

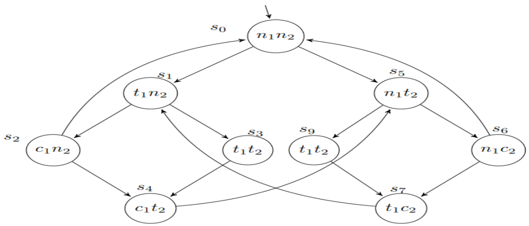
$$G(t_1 \rightarrow Fc_1)$$

3.非阻塞性: 不能被LTL表示. 这是因为需要表达:对于满足n_1的每个状态, 要有后继状态满足t_1,然而路径上的存在量词不能被LTL表示.

1.安全性:被初始状态满足(每个状态都满足).

2. 活性: 不被初始状态满足, 这是因为在路$s_0 \to s_1 \to s_3 \to s_7 \to s_1 \to s_3 \to s_7 \cdots$上$c_1$总是错的.

问题: 能否设计满足活性的互斥模型



Another model for mutual exclusion.

exclusion2.png

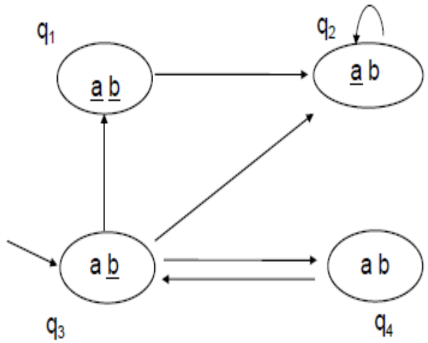
作业

1. 画出下面LTL公式的Parse 树
   - $Fp \wedge Gq \rightarrow pWr$
   - $F(p \rightarrow Gr) \vee \neg qUp$
   - $pW(qWr)$
   - $GFp \rightarrow F(q \vee s)$
2. 证明: $\phi U\psi \equiv \psi R(\phi \vee \psi) \wedge F\psi$
3. 依照下图的系统，考虑下面每个LTL公式ϕ
   - Ga
   - aUb
   - $aUX(a \wedge \neg b)$
   - $X\neg b \wedge G(\neg a \vee \neg b)$
   - $X(a \wedge b) \wedge F(\neg a \wedge \neg b)$

   (a) 找到一条从q_3出发的路，满足公式ϕ

(b) 确定是否有$\mathcal{M}, q_3 \models \phi$.

(c) 若将$\underline{a}$和$\underline{b}$解释为a与b的非，并表示通信协议中的发射信息，而a, b为接受信息，解释这些公式的具体含义.



计算树逻辑CTL

- 计算树逻辑，也叫分支时间逻辑，Computation Tree Logic，Branching-Time Logic。
- 它的时间模型向一棵树的结构，其未来是不确定的，未来会有不同的路，而且任何一条路都是一条实际的路。
- LTL的时态连接词U, F, G, X+量词A和E，其中A表示所有的路，而E表示存在一条路。

CTL的语法

CTL的公式定义为:

$$\begin{aligned}\phi \; ::= \; & \bot \mid \top \mid p \mid \neg\phi \mid \phi\wedge\phi \mid \phi\vee\phi \mid \phi\rightarrow\phi \mid \\ & AX\phi \mid EX\phi \mid AF\phi \mid EF\phi \mid AG\phi \mid EG\phi \mid \\ & A[\phi U\phi] \mid E[\phi U\phi].\end{aligned}$$

其中p是原子命题公式。
例子: $A(AX\neg pUE(EX(p\wedge q)U\neg p)$
Parse树:

Back
Close

例子

1. 存在一可到达满足q的状态

$$EFq$$

2. 从所有满足p的状态出发，有一直保持p直到满足q的状态出现

$$AG(p \rightarrow E[pUq])$$

3. 只要满足p的状态出现，就有系统可能永远保持q

$$AG(p \rightarrow EGq)$$

4. 有一可达的状态使得从此状态出发的所有可达状态都满足q

$$EFAGq$$

5. 进程总可以请求进入它的界区

$$AG(r \to EXt)$$

6. 对于任何状态，若一个请求出现则这个请求最终会被接受

$$AG(\underline{\text{请求}} \to AF\ \underline{\text{接受}})$$

7. 一部在2楼处于上升电梯，当有乘客在想到5楼时，电梯不会改变上升方向直到5楼

$$AG(\underline{\text{2楼}} \wedge \underline{\text{上升}} \wedge \underline{\text{按下5楼按钮}} \to A[\underline{\text{上升}}U\underline{\text{5楼}}])$$

8. 从任何状态出发总能到达<u>Restart</u>状态

$$AG(EF\underline{\text{Restart}})$$

CTL的语义

定义

给定模型$\mathcal{M} = (S, \to, L), s \in S, \phi$是CTL公式。以$\phi$的结构归纳定义$\mathcal{M}, s \models \phi$如下：

1. $\mathcal{M}, s \models \top$
2. $\mathcal{M}, s \not\models \bot$
3. $\mathcal{M}, s \models p$ 当且仅当$p \in L(s)$
4. $\mathcal{M}, s \models \neg\phi$ 若$\mathcal{M}, s \not\models \phi$
5. $\mathcal{M}, s \models \phi \wedge \psi$ 若$\mathcal{M}, s \models \phi$且$\mathcal{M}, s \models \psi$.
6. $\mathcal{M}, s \models \phi \vee \psi$ 若$\mathcal{M}, s \models \phi$或$\mathcal{M}, s \models \psi$.
7. $\mathcal{M}, s \models \phi \to \psi$ 若$\mathcal{M}, s \models \phi$则$\mathcal{M}, s \models \psi$
8. $\mathcal{M}, s \models AX\phi$ 若对于所有的s_1, 只要$s \to s_1$就有$\mathcal{M}, s_1 \models \phi$
9. $\mathcal{M}, s \models EX\phi$ 若存在某个s_1使得$s \to s_1$且$\mathcal{M}, s_1 \models \phi$
10. $\mathcal{M}, s \models AG\phi$ 若对于从s出发的所有路$s_1 \to s_2 \to \cdots$,其中s_1就是s，以及此路上的所有s_i 都有$\mathcal{M}, s_i \models \phi$

Back
Close

11. $\mathcal{M}, s \models EG\phi$ 存在一条从s出发的路$s_1 \rightarrow s_2 \rightarrow \cdots$,其中$s_1$就是$s$，以及此路上的所有$s_i$ 都有$\mathcal{M}, s_i \models \phi$

12. $\mathcal{M}, s \models AF\phi$ 若对于从s出发的所有路$s_1 \rightarrow s_2 \rightarrow \cdots$,其中$s_1$就是$s$，以及路上有$s_i$使得$\mathcal{M}, s_i \models \phi$

13. $\mathcal{M}, s \models EF\phi$ 若存在一条从s出发的路$s_1 \rightarrow s_2 \rightarrow \cdots$,其中$s_1$就是$s$，以及此路上有$s_i$使得$\mathcal{M}, s_i \models \phi$

14. $\mathcal{M}, s \models A[\phi U \psi]$ 若对于从s出发的所有路$s_1 \rightarrow s_2 \rightarrow \cdots$,其中$s_1$就是$s$,存在$i \geq 1$使得$\mathcal{M}, s_i \models \psi$且对于所有的$j = 1, 2, \cdots, i-1$都有$\mathcal{M}, s_i \models \phi$.

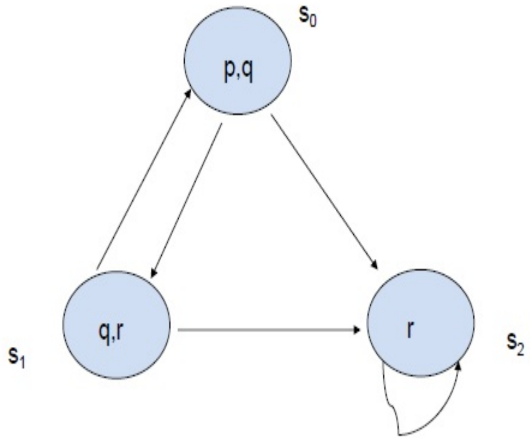
15. $\mathcal{M}, s \models E[\phi U \psi]$ 若存在从s出发的路$s_1 \rightarrow s_2 \rightarrow \cdots$,其中$s_1$就是$s$,存在$i \geq 1$使得$\mathcal{M}, s_i \models \psi$且对于所有的$j = 1, 2, \cdots, i-1$都有$\mathcal{M}, s_i \models \phi$.

定义　模型满足性

设$\mathcal{M} = \{S, \rightarrow, L\}$是模型，$\phi$是CTL公式。若对于任一$s \in S$都有$\mathcal{M}, s \models \phi$，则称模型$\mathcal{M}$满足CTL公式$\phi$, 记作$\mathcal{M} \models \phi$。

Back
Close

例子



语义等价

定义　语义等价

CTL公式ϕ,ψ称为语义等价，记作$\phi \equiv \psi$，若对于任何模型$\mathcal{M}$都有$\mathcal{M} \models \phi$当且仅当$\mathcal{M} \models \psi$。

定理　下面各条成立：

1. $AF\phi \equiv \neg EG\neg\phi$
2. $EF\phi \equiv \neg AG\neg\phi$
3. $AX\phi \equiv \neg EX\neg\phi$
4. $AF\phi \equiv A[\top U \phi]$
5. $EF\phi \equiv E[\top U \phi]$

CTL连接词充分性

定理

CTL时态连接词集是充分的当且仅当它包含EU以及$\{AX, EX\}$中一个元素以及$\{EG, AF, AU\}$中一个元素。

若选用$\{EX, EU, AF\}$为时态连接词充分集，则有下面各条成立：

1. $AX\phi \equiv \neg EX\neg\phi$
2. $EF\phi \equiv E[\top U\phi]$
3. $EG\phi \equiv \neg AF\neg\phi$
4. $AG\phi \equiv \neg EF\neg\phi$
5. $A[\phi_1 U\phi_2] \equiv \neg(E[\neg\phi_2 U(\neg\phi_1 \wedge \neg\phi_2)] \vee EG\neg\phi_2)$

可写成:

$$\phi ::= \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid EX\phi \mid EG\phi \mid E[\phi U\phi].$$

下面的语义等价也是重要的, 可用于CTL模型检测算法.

$$
\begin{aligned}
AG\phi &\equiv \phi \wedge AX\,AG\phi \\
EG\phi &\equiv \phi \wedge EX\,EG\phi \\
AF\phi &\equiv \phi \vee AX\,AF\phi \\
EF\phi &\equiv \phi \vee EX\,EF\phi \\
A[\phi U \psi] &\equiv \psi \vee (\phi \wedge AX\,A[\phi U \psi]) \\
E[\phi U \psi] &\equiv \psi \vee (\phi \wedge EX\,E[\phi U \psi])
\end{aligned}
$$

CTL和LTL表达能力

- 存在LTL公式, 在CTL中没有等价的形式, 例如: FGq;
- 存在CTL公式, 在LTL中没有等价的形式, 例如: $AFAGq$;

CTL*

状态公式:

$$\phi ::= \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid A[\alpha] \mid E[\alpha]$$

路径公式

$$\alpha ::= \phi \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha U \alpha \mid G\alpha \mid F\alpha \mid X\alpha$$

注:LTL和CTL是CTL*子集.

Back
Close

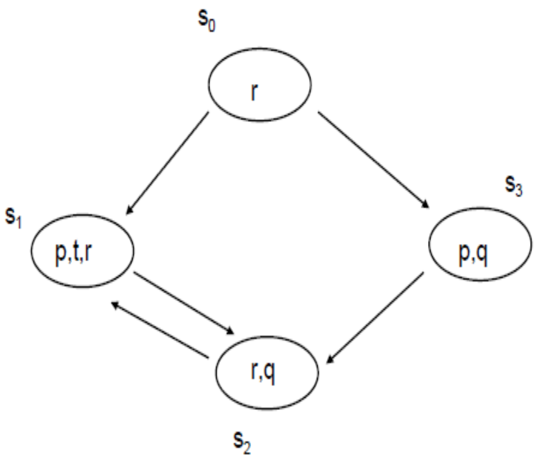
作业

1. 画出下面CTL公式的Parse树
   - $EFEGp \to AFr$
   - $A[pUA[qUr]]$
   - $E[A[pUq]Ur]$
   - $AG(p \to A[pU(\neg p \wedge A[\neg Uq])])$
2. 依照下图的系统,
   (a) 从s_0开始，将这个系统展开成一个无穷树，并画出所有长度为4的计算路.
   (b) 确定是否有$\mathcal{M}, s_0 \models \phi$以及$\mathcal{M}, s_2 \models \phi$成立，其中$\phi$是LTL或CTL公式:
      i. $\neg p \to r$
      ii. Ft
      iii. $\neg EGr$
      iv. $E(tUq)$

v. EFq

vi. EGr

vii. $G(r \vee q)$



3. 设$\mathcal{M} = (S, \rightarrow, L)$是任何CTL模型，用符号$[\![\phi]\!]$表示集合$\{s \mid s \in S, \mathcal{M}, s \models \phi\}$.证明：

   (a) $[\![\top]\!] = S$

(b) $[\![\bot]\!] = \emptyset$

(c) $[\![\neg\phi]\!] = S - [\![\phi]\!]$

(d) $[\![\phi \wedge \psi]\!] = [\![\phi]\!] \cap [\![\psi]\!]$

(e) $[\![\phi \vee \psi]\!] = [\![\phi]\!] \cup [\![\psi]\!]$

(f) $[\![\phi \to \psi]\!] = (S - [\![\phi]\!]) \cup [\![\psi]\!]$

(g) $[\![AX\phi]\!] = S - [\![EX\neg\phi]\!]$

(h) $[\![A(\phi U \psi)]\!] = [\![\neg(E(\neg\phi U(\neg\phi \wedge \neg\psi)) \vee EG\neg\psi)]\!]$

面向CTL的模型检测

时态逻辑系统可用于模型检测。

模型通常是迁移系统/有限自动机,它描述了状态迁移过程，反映状态的演化, 而公式是时态逻辑公式ϕ。

模型检测的目的是表明模型$\mathcal{M}$满足公式ϕ, 即$\mathcal{M} \models \phi$。通常实现模型检测，需要做下面三件事情:

- 建立模型$\mathcal{M}$,
- 编写公式ϕ,
- 运行模型检测器，输入$\mathcal{M}$和ϕ,
- 模型检测器将输出Yes若$\mathcal{M} \models \phi$成立,否则输出No。

模型检测的基本问题是: 给定一个模型$M = (S, \rightarrow, L)$, $s_0 \in S$ 以及CTL公式ϕ，是否有$M, s_0 \models \phi$ 成立?

模型检测算法：$SAT_M(\phi) = \{s \mid M, s \models \phi\}$, 模型$M$中状态满足$\phi$的状态集.

基本问题就变成了s_0是否属于集合$SAT_M(\phi)$?

不动点

定义 (单调函数) 设S是一状态集，2^S是S的幂集，$F: 2^S \to 2^S$是一函数，

(a) 称F是单调函数，若$\forall X, Y \in 2^S$，当$X \subseteq Y$时有$F(X) \subseteq F(Y)$,

(b) S的子集X称为F的不动点，若$F(X) = X$.

符号$F^i(X)$表示函数F作用在集合X上i次, 即

$$F^i(X) = \begin{cases} F(X), & i = 1 \\ F(F^{i-1}(X)), & i > 1 \end{cases}$$

定理 (Knaster-Tarski不动点定理)

设S是含有$n+1$个元素的集合，$F:2^S\to 2^S$是单调函数，则$F^{n+1}(\emptyset)$是F的最小不动点，而$F^{n+1}(S)$是F的最大不动点。

证明:

因为$\emptyset\subseteq F(\emptyset)$, 所以由$F$ 的单调性有$F(\emptyset)\subseteq F(F(\emptyset))$,即,$F^1(\emptyset)\subseteq F^2(\emptyset)$.进一步可以使用数学归纳法证明: 对所有的$i\geq 1$

$$F^1(\emptyset)\subseteq F^2(\emptyset)\subseteq F^3(\emptyset)\subseteq\cdots F^i(\emptyset).$$

特别的,当$i=n+1$时,上面的表达式中有某个k,使得$F^k(\emptyset)$是F的不动点. 否则, $F^1(\emptyset)$至少含有一个元素. 由于$F^1(\emptyset)\subseteq F^2(\emptyset)$,所以$F^2(\emptyset)$至少含有两个元素.继续这个过程,最终将得到$F^{n+2}(\emptyset)$将含有至少$n+2$个元素. 这与$S$只含有$n+1$个元素矛盾.因此$F(F^k(\emptyset))=F^k(\emptyset)$,其中$0\leq k\leq n+1$,这蕴含了$F^{n+1}(\emptyset)$是$F$的不动点.

下面证明$F^{n+1}(\emptyset)$是最小不动点. 假设X是F的任一不动点,需要证明$F^{n+1}(\emptyset)\subseteq X$. 由于$\emptyset\subseteq X$, 所以由$F$的单调性以及$X$是$F$的不动点得到$F(\emptyset)\subseteq F(X)=X$. 通过归纳得到$F^i(\emptyset)\subseteq X$ $(i\geq 0)$. 特别地,$F^{n+1}(\emptyset)\subseteq X$.

最大不动点的证明类似,此处忽略.

例子

设$Y \in 2^S$, $s_0 \in S$，定义$F(Y) = Y \cup \{s_0\}$，则F是2^S上的单调函数, 且对于任一$Y \subseteq S$都有$F(Y)$是F的不动点. 最小不动点是$\{s_0\}$, 最大不动点是S.

定义　符号: 设$Y \subseteq S$，定义

$pre_\exists(Y) = \{s \in S \mid$ 存在$s' \in S$使得$s \to s'$且$s' \in Y\}$,
即若s有一个后继在Y中，则s在$pre_\exists(Y)$中。

$pre_\forall(Y) = \{s \in S \mid$对于所有$s' \in S$，若$s \to s'$则$s' \in Y\}$,
即若s的所有后继都在Y中，则s在$pre_\forall(Y)$中.

事实:

- $pre_\forall(Y) = S - pre_\exists(S - Y)$.
- 函数$pre_\exists(Y)$ 和$pre_\forall(Y)$ 是单调的.

回忆:

$$\begin{aligned} AF\phi &\equiv \phi \vee AXAF\phi \\ E[\phi U\psi] &\equiv \psi \vee (\phi \wedge EXE[\phi U\psi]). \end{aligned}$$

记:

$$\begin{aligned} SAT(AF\phi) &= SAT_{AF}(\phi) \\ SAT(E[\phi U\psi]) &= SAT_{EU}(\phi,\psi) \\ SAT(AX\phi) &= SAT_{AX}(\phi) \\ SAT(EX\phi) &= SAT_{EX}(\phi) \end{aligned}$$

则:

$$\begin{aligned} SAT_{EX}(\phi) &= pre_{\exists}(SAT(\phi)) \\ SAT_{AX}(\phi) &= pre_{\forall}(SAT(\phi)) \end{aligned}$$

进一步,

$$\begin{aligned} SAT_{AF}(\phi) &= SAT(\phi) \cup pre_{\forall}(SAT_{AF}(\phi)) && (*1) \\ SAT_{EU}(\phi,\psi) &= \psi \cup (\phi \cap pre_{\exists}(SAT_{EU}(\phi,\psi))) && (*2) \end{aligned}$$

定理 (SAT_{AF})

设ϕ是一CTL公式, $M=(S,\rightarrow,L)$是一模型, 其中S是含有$n+1$个状态的集合。定义函数$F:2^S\rightarrow 2^S$为

$$F(X)=SAT(\phi)\cup pre_{\forall}(X),\forall X\in 2^S$$

则$SAT_{AF}(\phi)=F^{n+1}(\emptyset)$, 即$SAT_{AF}(\phi)$是$F$的最小不动点。

证明:

(1) F 是单调的.

(2)由前面$(*1)$知, $SAT_{AF}(\phi)$ 是F的不动点.

(3)$SAT_{AF}(\phi)=F^{n+1}(\emptyset)$,分两步完成:

首先证明:对于任意的n, $F^n(\emptyset)\subseteq SAT_{AF}(\phi)$, 可用数学归纳法.

(i)$n=0$时, $F^1(\emptyset)=SAT(\phi)\cup pre_{\forall}(\emptyset)=SAT(\phi)\subseteq SAT_{AF}(\phi)$, 成立.

(ii)假设$n=k$时,结论成立, 即, $F^k(\emptyset)\subseteq SAT_{AF}(\phi)$.
则当$n=k+1$时有

$$\begin{array}{rcl} F^{k+1}(\emptyset) & = & F(F^k(\emptyset)) \\ & = & SAT(\phi)\cup pre_{\forall}(F^k(\emptyset)) \\ & \subseteq & SAT(\phi)\cup pre_{\forall}(SAT_{AF}(\phi)) \\ & = & SAT_{AF}(\phi) \end{array}$$

因此, 对于任意的n, $F^n(\emptyset)\subseteq SAT_{AF}(\phi)$成立,特别地,

$$F^{n+1}(\emptyset)\subseteq SAT_{AF}(\phi)$$

其次, 假设存在s_0满足: $s_0\in SAT_{AF}(\phi)$,但$s_0\notin F^{n+1}(\emptyset)$. 则由

$$F^{n+1}(\emptyset)=SAT(\phi)\cup pre_{\forall}(F^n(\emptyset))$$

知$s_0\notin SAT(\phi)$,且存在s_1使得$s_0\to s_1$且$s_1\notin F^n(\emptyset)$.
进一步, 由$s_1\notin F^n(\emptyset)$
知$s_1\notin SAT(\phi)$,且存在s_2使得$s_1\to s_2$且$s_2\notin F^{n-1}(\emptyset)$.

继续这个过程得到:

$$s_0 \to s_1 \to s_2 \cdots \to s_n \to s_{n+1}$$

其中$s_i \notin SAT(\phi)(0 \le i \le n+1)$.
由于S含有$n+1$个元素, 所以存在i, j使得$s_i = s_j$.这样得到一条路:

$$s_0 \to s_1 \to s_2 \cdots \to s_i \to s_i \to s_i \cdots$$

这条路上所有的状态都不满足ϕ,这与$s_0 \models AF\phi$矛盾! 因此,

$$SAT_{AF}(\phi) \subseteq F^{n+1}(\emptyset).$$

结果

$$SAT_{AF}(\phi) = F^{n+1}(\emptyset).$$

定理 (SAT_{EU})

设ϕ,ψ是CTL公式, $M=(S,\to,L)$是一模型，其中S是含有$n+1$个状态的集合.定义函数$G:2^S\to 2^S$为

$$G(X)=SAT(\psi)\cup(SAT(\phi)\cap pre_{\exists}(X)),\forall X\in 2^S$$

则$SAT_{EU}(\phi,\psi)=G^{n+1}(\emptyset)$, 即$SAT_{EU}(\phi,\psi)$是$G$的最小不动点。

证明:

(1) G 是$2^S\to 2^S$上的单调函数.

(2)据$(*2)$, $SAT_{EU}(\phi,\psi)$是G的不动点.

(3)下面我们证明$G^{n+1}(\emptyset)=SAT_{EU}(\phi,\psi)$.

首先利用数学归纳法证明: 对于任意的k, $G^k(\emptyset)\subseteq SAT_{EU}(\phi,\psi)$(留作练习).这样,

$$G^{n+1}(\emptyset)\subseteq SAT_{EU}(\phi,\psi).$$

另一方面, 设$s\in SAT_{EU}(\phi,\psi)$.则存在一条路$s=s_1\to s_2\cdots\to s_k\to s_{k+1}\cdots$ 使得$s_i\models\phi$ $(i\le k)$, $s_{k+1}\models\psi$. 此时$s\in G^{k+1}(\emptyset)$. 这是因为, 当$k=0$时,$s=s_1\in SAT(\psi)=G^1(\emptyset)$;当$k=1$时,$s=s_1\to s_2\in SAT(\psi)$, $s_1\in G^2(\emptyset)$. 一般情形可用数学归纳法证明.这说明

$$SAT_{EU}(\phi,\psi)\subseteq G^{n+1}(\emptyset)$$

Back
Close

所以有$G^{n+1}(\emptyset) = SAT_{EU}(\phi, \psi)$.

思考: $SAT_{EG}(\phi)$的情况:是什么函数的不动点?最小最大?

定理 (SAT_{EG})

设ϕ是CTL公式, $M=(S,\to,L)$是一模型，其中S是含有$n+1$个状态的集合.定义函数$F:2^S\to 2^S$为

$$F(X)=SAT(\phi)\cap pre_{\exists}(X),\forall X\in 2^S$$

则$SAT_{EG}(\phi)=F^{n+1}(S)$, 即$SAT_{EG}(\phi)$是$F$的最大不动点.

标号算法

标号算法(Labelling Algorithm)的基本思想是:若状态s满足ϕ,则在s处进行标注公式ϕ，具体是以ϕ的结构进行标号。依据连接词的充分性，只需考虑六个连接词:

$$\perp, \neg, \wedge, AF, EU, EX$$

定义 **语标号算法** 设ψ是ϕ的子公式且满足ψ的所有直接子公式的状态都已标注了，现在来标注ψ, 标号算法如下: ψ是

1. $\perp$: 没有状态标注$\perp$
2. p: 若$p \in L(s)$则状态s标注p
3. $\psi_1 \wedge \psi_2$: 若s已标注了ψ_1 和ψ_2则s标注$\psi_1 \wedge \psi_2$
4. $\neg\psi_1$: 若s没有标注ψ_1则s标注$\neg\psi_1$
5. $AF\psi_1$:
   - 若状态s标注了ψ_1则s标注$AF\psi_1$
   - 若状态s的所有后继都标注了$AF\psi_1$，则s标注$AF\psi_1$
6. $E[\psi_1 U \psi_2]$:

- 若状态s标注了ψ_2，则状态s标注$E[\psi_1 U \psi_2]$
- 若状态s标注了ψ_1且s有一个后继标注了$E[\psi_1 U \psi_2]$，则状态s标注$E[\psi_1 U \psi_2]$

7. $EX\psi_1$: 若状态s的一个后继标注了ψ_1，则s标注$EX\psi_1$。

通过标注算法可将一个模型中状态进行标注，实现对这个模型的检测.

CTL检测算法伪代码

本段定义伪代码输出集合$SAT(\phi) = \{s \in S \mid s \models \phi\}$.

主函数$SAT(\phi)$:　　　/* 确定满足ϕ的状态集合/

Begin

CASE

ϕ is $\top$: return S

ϕ is $\bot$: return $\emptyset$

ϕ is atomic formula p: return $\{s \in S \mid p \in L(s)\}$

ϕ is $\neg\phi$: return $S - SAT(\phi)$

ϕ is $\phi_1 \wedge \phi_2$: return $SAT(\phi_1) \cap SAT(\phi_2)$

ϕ is $\phi_1 \vee \phi_2$: return $SAT(\phi_1) \cup SAT(\phi_2)$

ϕ is $\phi_1 \rightarrow \phi_2$: return $SAT(\neg\phi_1 \vee \phi_2)$

ϕ is $AX\phi_1$: return $SAT(\neg EX\neg\phi_1)$

ϕ is $EX\phi$: return $SAT_{EX}(\phi_1)$

ϕ is $A[\phi_1 U \phi_2]$: return $SAT(\neg(E[\neg\phi_2 U(\neg\phi_1 \wedge \neg\phi_2)] \vee EG\neg\phi_2))$

ϕ is $E[\phi_1 U \phi_2]$: return $SAT_{EU}(\phi_1, \phi_2)$

ϕ is $EF\phi_1$: return $SAT(E[TU\phi_1])$

ϕ is $AF\phi_1$: return $SAT_{AF}\phi_1$

ϕ is $AG\phi_1$: return $SAT(\neg EF\neg\phi_1)$

End CASE

End

由这个伪代码可以看到，我们只需考虑三个基本时态连接词EX, EU, AF的计算方法。

Back
Close

连接词EX, AF, EU的伪代码

建立$SAT_{EX}(\phi), SAT_{AF}(\phi), SAT_{EU}(\phi,\psi)$伪代码。
设ϕ是CTL公式，$M=(S,\rightarrow,L)$是一个模型，

Function $SAT_{EX}(\phi)$
/* 确定满足$EX\phi$的状态集合/
Local var: X, Y
Begin
$X := SAT(\phi)$;
$Y := pre_{\exists}(X)$;
Return Y
End

SAT_{EX}伪代码的正确性:由于$SAT_{EX}(\phi) = pre_{\exists}(SAT(\phi))$是成立的，因而$SAT_{EX}(\phi)$是正确的.

Function $SAT_{AF}(\phi)$

/* 确定满足$AF\phi$的状态集合/

Local var: X, Y

Begin

$X := S;$

$Y := SAT(\phi);$

Repeat until $X = Y$

Begin

$X := Y$

$Y := Y \cup pre_{\forall}(Y)$

End

Return Y

End

$SAT_{AF}(\phi)$伪代码的正确性:

$$\begin{aligned}
Y_0 &= SAT(\phi) = F(\emptyset)\\
Y_1 &= Y_0 \cup pre_{\forall}(Y_0) = F(Y_0) = F^2(\emptyset)\\
Y_2 &= Y_1 \cup pre_{\forall}(Y_1)\\
&= Y_0 \cup pre_{\forall}(Y_0) \cup pre_{\forall}(Y_0 \cup pre_{\forall}(Y_0))\\
&= Y_0 \cup pre_{\forall}(Y_1)\\
&= F(Y_1) = F^3(\emptyset)\\
&\vdots\\
Y_n &= Y_0 \cup pre_{\forall}(Y_{n-1}) = F(Y_{n-1}) = F^n(\emptyset)
\end{aligned}$$

判断条件$X = Y$,即为

$$Y_k = F(Y_k)$$

由不动点定理:

$$F^{n+1}(\emptyset) = F(F^{n+1}(\emptyset))$$

且

$$SAT_{AF}(\phi) = F^{n+1}(\emptyset).$$

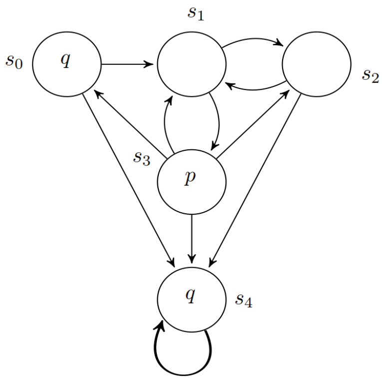
知上述算法是正确的且终止.

```
Function SAT_EU(φ, ψ)
    /* 确定满足E[φUψ]的状态集合/
 Local var: X, Y, W
  Begin
    W := SAT(φ) /* 标注了φ的状态集
    X := S
    Y := SAT(ψ) /* 标注了ψ的状态集
    Repeat Until X = Y
    Begin
      X := Y
      Y := Y ∪ (W ∩ pre_∃(Y))
    End
    Return Y
  End
```

$SAT_{EU}(\phi, \psi)$伪代码的正确性:同上面类似分析.

例子:计算$SAT(EFp)$ 与$SAT(EGq)$.

Back
Close



A system for which we compute invariants.

作业

(1)证明:设ϕ是CTL公式, $M=(S,\rightarrow,L)$是一模型，其中S是含有$n+1$个状态的集合.定义函数$F:2^S\rightarrow 2^S$为

$$F(X)=SAT(\phi)\cap pre_{\exists}(X),\forall X\in 2^S$$

则$SAT_{EG}(\phi)=F^{n+1}(S)$, 即$SAT_{EG}(\phi)$是$F$的最大不动点.

(2)给出计算$SAT_{EG}(\phi)$的伪代码,并分析其正确性和终止.