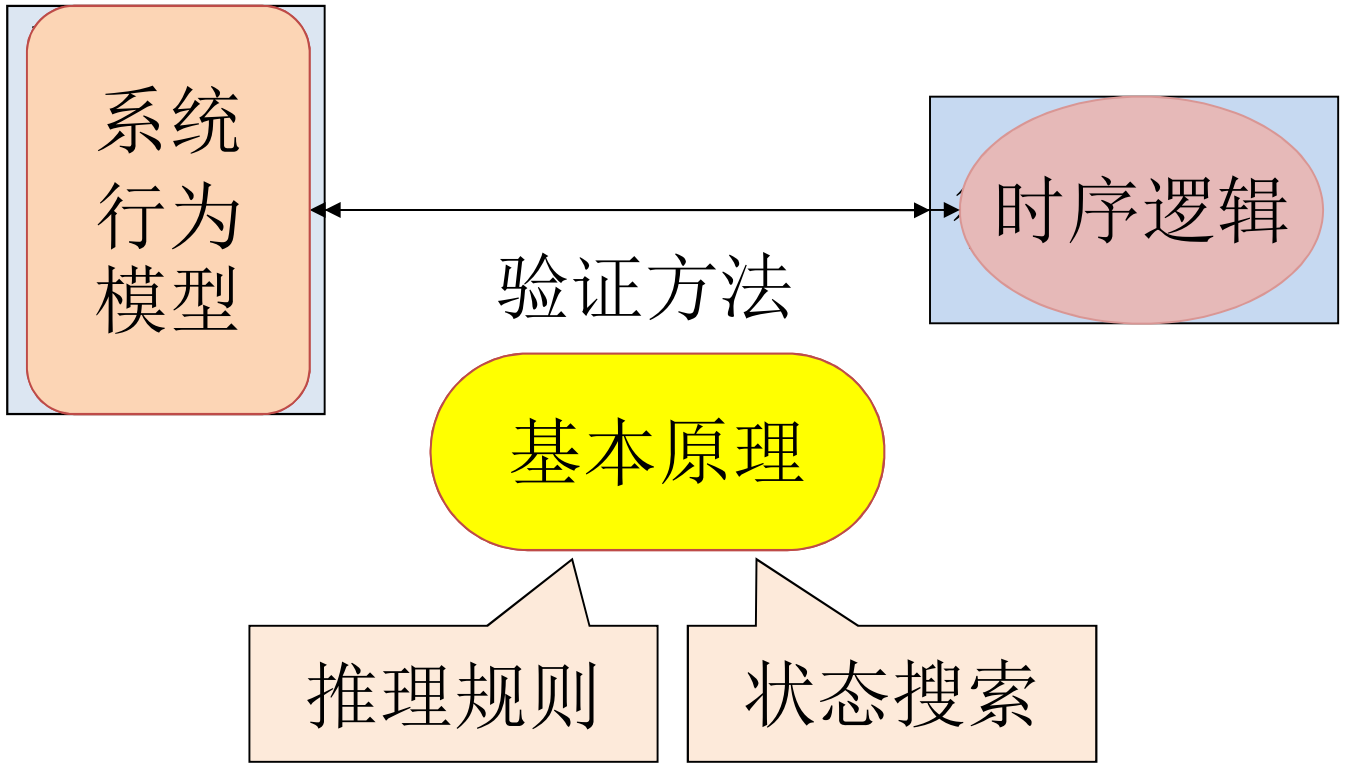
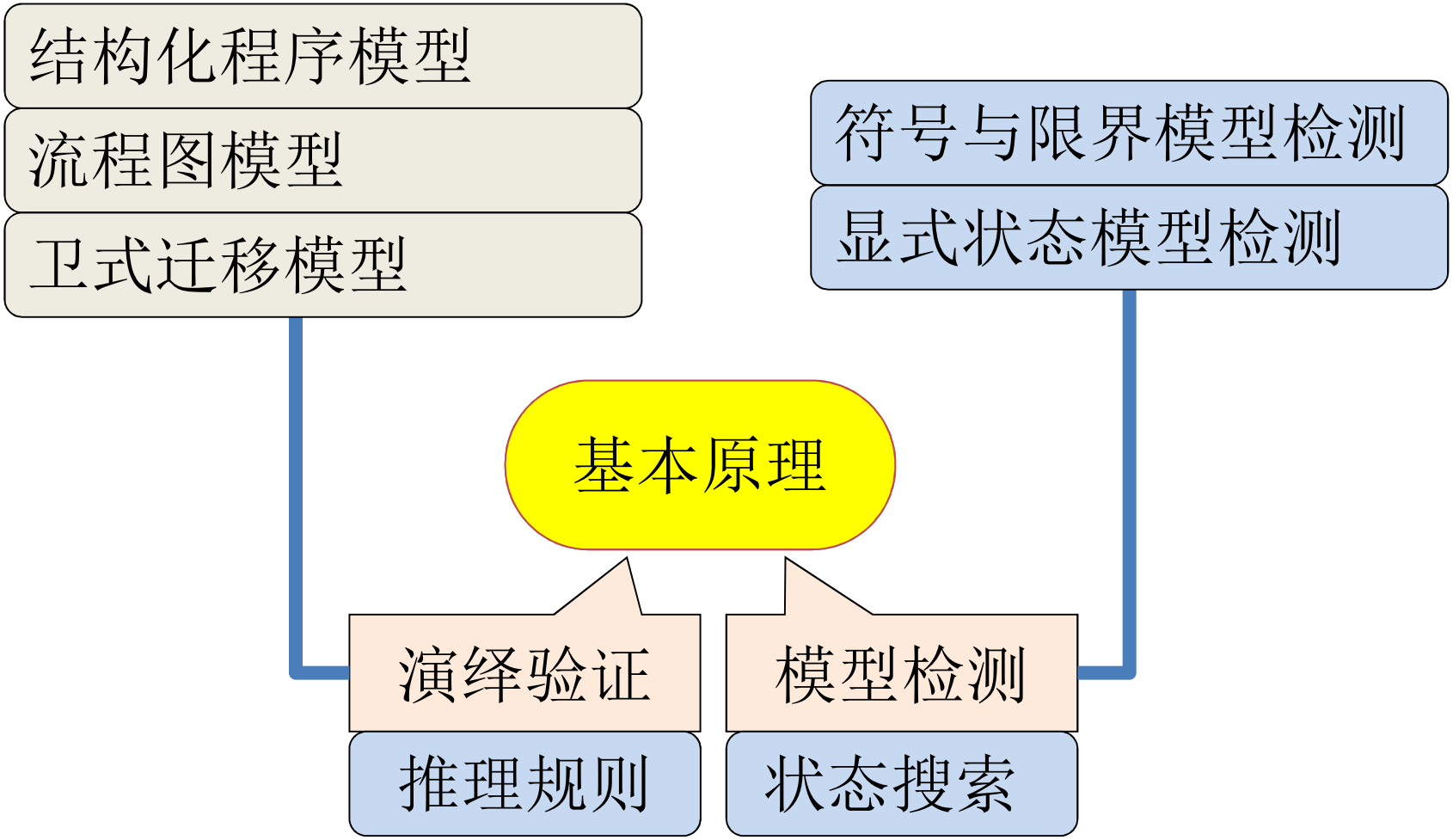
模型检测方法

中国科学院软件研究所
计算机科学国家重点实验室
张文辉
http://lcs.ios.ac.cn/~zwh/

课程内容



课程内容


结构化程序模型
流程图模型
卫式迁移模型
符号与限界模型检测
显式状态模型检测
基本原理
演绎验证
模型检测
推理规则
状态搜索

模型

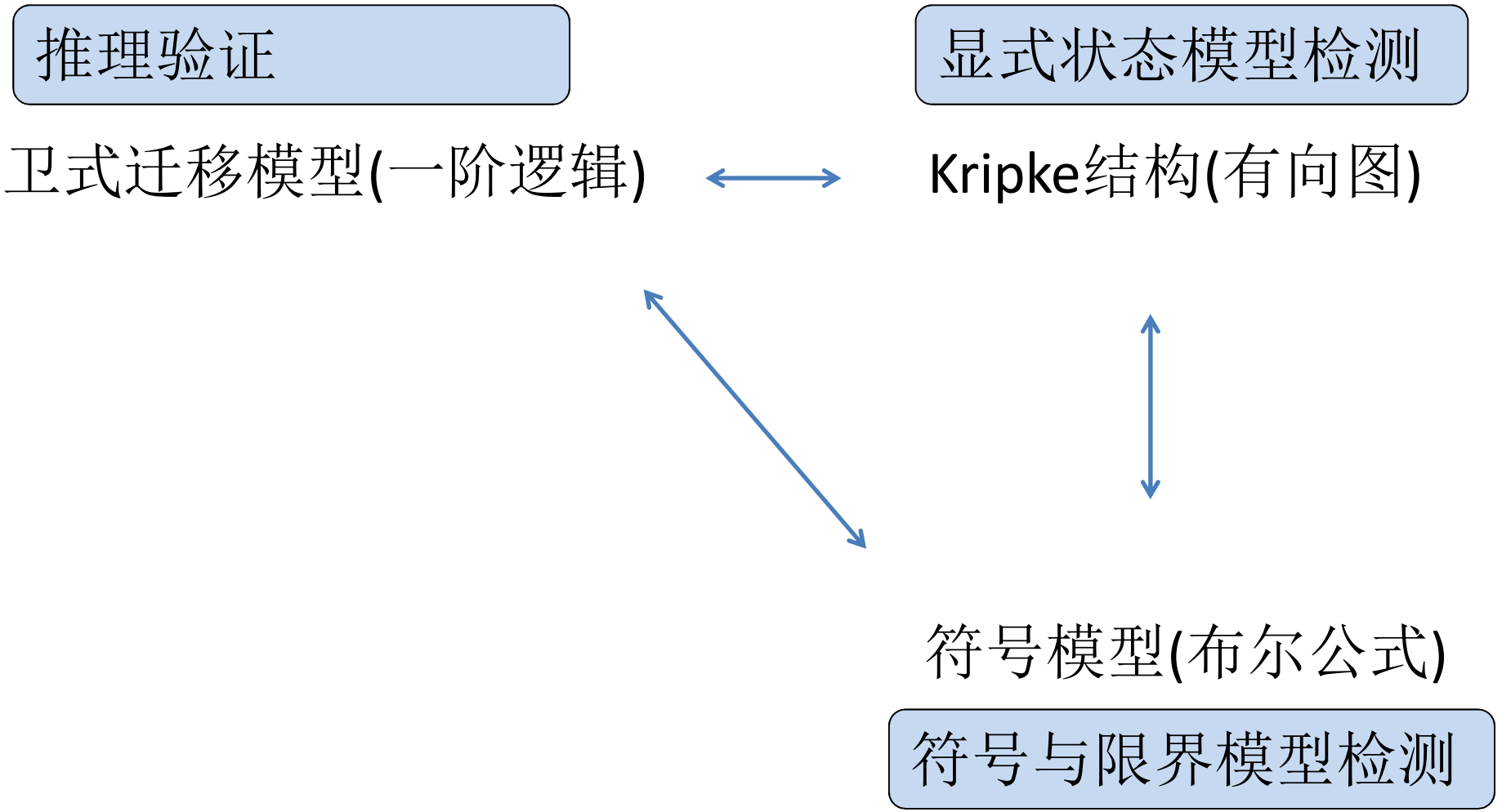
逻辑:

卫式迁移模型	$M=(T,\Theta,\Phi)$
流程图模型	T
结构化程序模型	S

状态:

Kripke结构	$M=(S,R,I,L,\Phi)$
自动机模型	$A=(\Sigma,S,\Delta,I,F)$

模型检测方法


推理验证
显式状态模型检测
卫式迁移模型(一阶逻辑)
Kripke结构(有向图)
符号模型(布尔公式)
符号与限界模型检测

内容

- 符号模型
- CTL模型检测方法
- LTL模型检测方法

(I) 符号模型

用BF(V)表示变量集合V上的布尔公式。

给定AP。
AP上的符号模型 M=(V,ρ,Θ,N)

- V = {v1,…,vn} 布尔变量集合
- ρ: {v1,…,vn,v1’,…,vn’} 上的布尔公式
- Θ: {v1,…,vn} 上的布尔公式
- N: AP→BF(V), 即N(p)为{v1,…,vn}上的布尔公式(代表p)

符号模型与标号Kripke模型的对应

给定AP.	给定AP.
AP上的标号Kripke模型	AP上的符号模型
K=<S,R,I,L>	M=(V,ρ,Θ,N)
S = {s0,…,sm}	V = {v1,…,vn} 布尔变量集合
$R \subseteq S \times S$	ρ: {v1,…,vn,v1’,…,vn’} 上的BF
$I \subseteq S$	Θ: {v1,…,vn} 上的BF
L: $S \rightarrow 2^S$	N: AP→BF(V)

标号Kripke模型到符号模型

给定标号Kripke模型

K=<S,R,I,L>

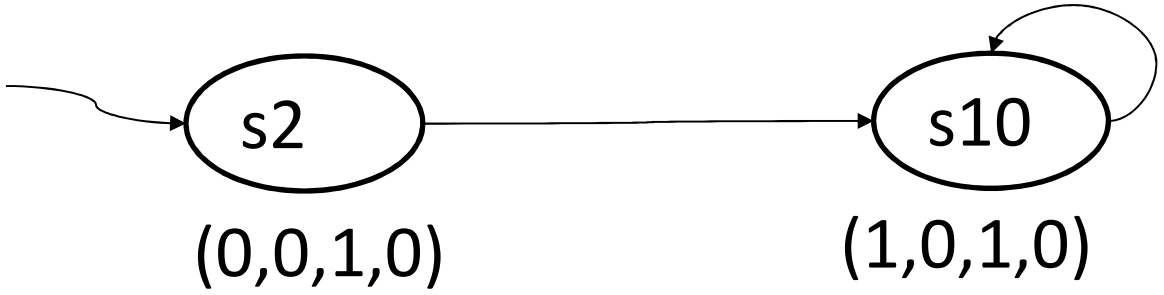
V={v1,…,vn}可以编码2^n个状态。

状态 --- 用n位2进制编码 --- 布尔公式
状态集
状态迁移 --- 用2n位2进制编码 --- 布尔公式
状态迁移关系

标号函数(状态→标号集) --- (标号→状态集)

符号模型

S = {s0,…,s15} ~ V={v1,v2,v3,v4}

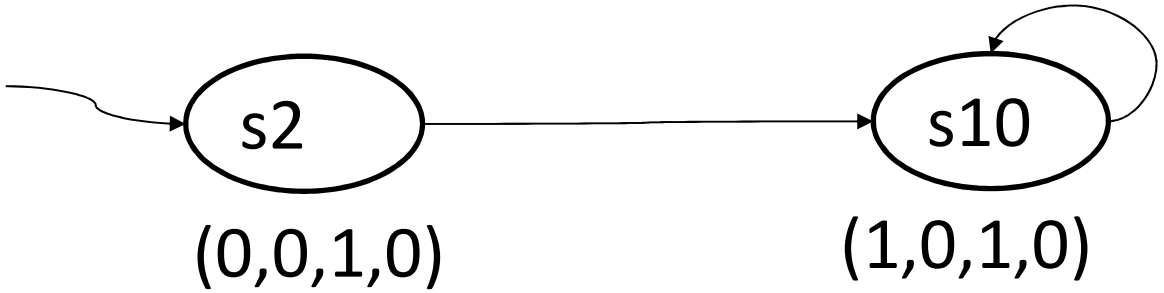


$(\neg v1 \wedge \neg v2 \wedge v3 \wedge \neg v4)$ $(v1 \wedge \neg v2 \wedge v3 \wedge \neg v4)$

$(\neg v2 \wedge v3 \wedge \neg v4)$

符号模型

$R \sim \rho$



(s2,s10):

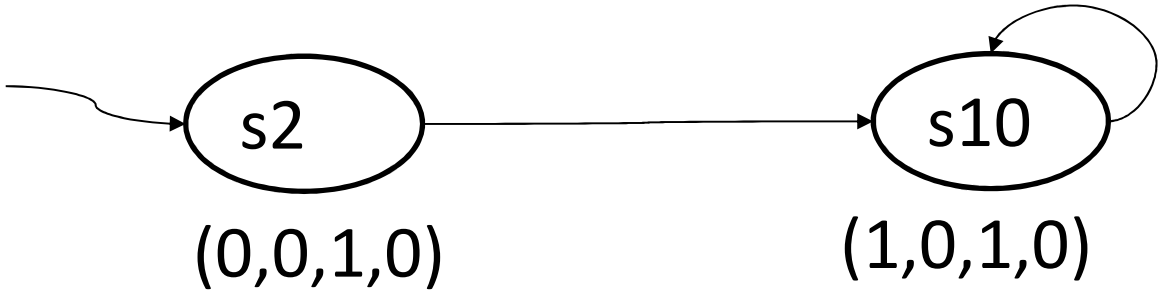
$(\neg v1 \wedge \neg v2 \wedge v3 \wedge \neg v4) \wedge (v1' \wedge \neg v2' \wedge v3' \wedge \neg v4')$

{(s2,s10),(s10,s10)}:

$(\neg v2 \wedge v3 \wedge \neg v4) \wedge (v1' \wedge \neg v2' \wedge v3' \wedge \neg v4')$

符号模型

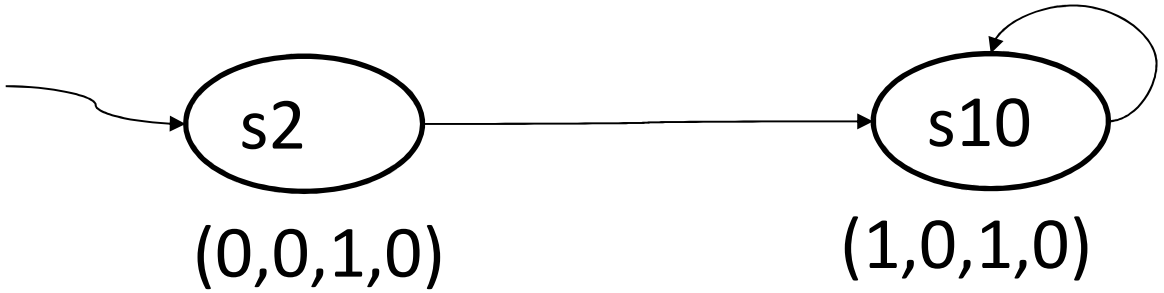
$I \sim \Theta$



$(\neg v1 \wedge \neg v2 \wedge v3 \wedge \neg v4)$

符号模型

L ~ N



AP={p,q}:

[[p]] = { s | p∈L(s) } ~ N(p)

[[q]] = { s | q∈L(s) } ~ N(q)

给定符号模型

(V,ρ,Θ,N)

$V=\{v1,\ldots vn\}$

标号Kripke结构

符号模型M=(V,ρ,Θ,N)与Kripke结构K=(S,R,I,L)的对应关系

S = { (a1,…,an) | a1,…,an ∈{0,1} }

R = { ((a1,…,an) , (a1’,…,an’)) | ρ(a1,…,an,a1’,…,an’) = 1 }

I = { (a1,…,an) | Θ(a1,…,an) = 1 }

L((a1,…,an)) = { p | N(p)(a1,...,an)=1, p∈AP }

M |= φ iff K|= φ.

符号模型的特殊情况

- AP={v1,...,vn};
- 符号模型M=(V,ρ,Θ,N)且N(vi) = vi
- 简化为M=(V,ρ,Θ)

标号Kripke结构

符号模型M=(V,ρ,Θ)与Kripke结构K=(S,R,I,L)的对应关系

$S = \{ (a1,\ldots,an) \mid a1,\ldots,an \in \{0,1\} \}$

$R = \{ ((a1,\ldots,an) , (a1',\ldots,an')) \mid \rho(a1,\ldots,an,a1',\ldots,an') = 1 \}$

$I = \{ (a1,\ldots,an) \mid \Theta(a1,\ldots,an) = 1 \}$

$L((a1,\ldots,an)) = \{ vi \mid i \in \{1,\ldots,n\} , ai = 1 \}$

$M \models \varphi$ iff $K \models \varphi$.

公式→状态集合

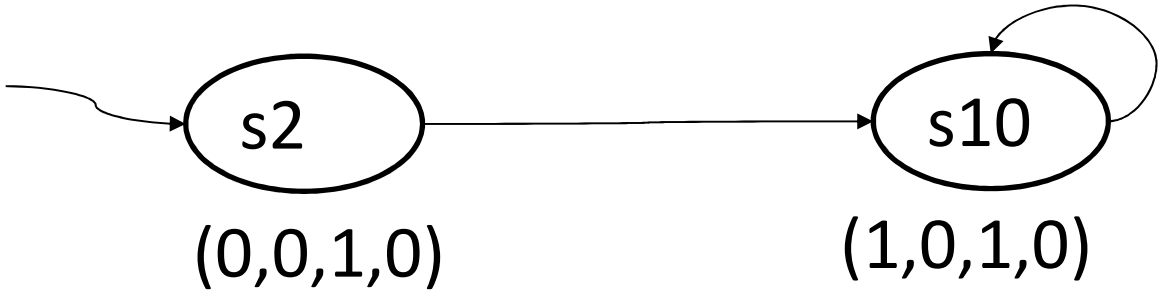
- α: (v1,…,vn)上的布尔公式
- s= (a1,…,an)
- 定义α(s)为用a1,…,an分别替换α中的v1,…,vn
- 定义s $\in$ α 当且仅当α(s) =1.
- 定义[[α]] = { s | α(s) =1 }

前驱集合公式的计算

- α:　　(v1,…,vn)上的布尔公式
- α':　　相应的(v1',…,vn')上的布尔公式
- $ex(\alpha) = \exists v1',\ldots,vn'.(\rho \wedge \alpha')$
- $EX([[\alpha]]) = \{ s \mid ex(\alpha)(s) = 1 \}$

符号模型

$\rho = (\neg v2 \wedge v3 \wedge \neg v4) \wedge (v1' \wedge \neg v2' \wedge v3' \wedge \neg v4')$



$EX(\{s10\}) = \{s2, s10\}$

$ex(v1 \wedge \neg v2 \wedge v3 \wedge \neg v4) = (\neg v2 \wedge v3 \wedge \neg v4)$

高效的符号模型表示及计算

AP上的符号模型 M=(V,ρ,Θ,N)

布尔公式的有序二叉决策图(OBDD)表示

集合(公式)表示的唯一性

集合(公式)的交并补(且或非)计算

前驱集合的计算(量词消去)

问题：

状态爆炸问题

(OBDD的大小与布尔变量个数成指数关系)

模型检测

- 问题：$M \models \varphi$

- 方法类型
  - 显式状态模型检测
  - 符号模型检测
  - 限界模型检测/限界正确性检查

- 性质类型
  - CTL
  - LTL

(II) CTL模型检测

- 显式状态模型检测
- 符号模型检测
- 限界模型检测/限界正确性检查
- 公平约束下的CTL模型检测

(II.a)显式状态模型检测

M=(S,R,I,L)

φ

M |= φ ?

考虑: $\neg$, $\vee$, EX, EG, EU

标号算法

$M=(S,R,I,L)$

φ

$M \models \varphi$?

将状态 对应到 φ 的子公式的子集

目标:

$M,s \models \varphi$ iff $\varphi \in L'(s)$

标号算法(a)

L’(s) := {} for all s

考虑φ的长度为1的子公式p

L’(s) := L’(s) $\cup$ {p} if p $\in$ L(s)

标号算法(b)

考虑φ的长度为k+1的子公式ψ

(1) $\psi = \neg\psi_0$

$L'(s) = L'(s) \cup \{\psi\}$ if $\psi_0 \notin L'(s)$

(2) $\psi = \psi_0 \vee \psi_1$

$L'(s) = L'(s) \cup \{\psi\}$ if $\psi_0 \in L'(s)$ or $\psi_1 \in L'(s)$

(3) $\psi = EX\psi_0$

$L'(s) = L'(s) \cup \{\psi\}$ if $\exists s'.(s \rightarrow s')$ and $\psi_0 \in L'(s')$

标号算法(c)

(4) $\psi =$ $E(\psi_0 U \psi_1)$

$L'(s) = L'(s) \cup \{\psi\}$ if $\psi_1 \in L'(s)$
$L'(s) = L'(s) \cup \{\psi\}$ if $\psi_0 \in L'(s)$ and $\exists s'.(s \rightarrow s')$ and $\psi \in L'(s')$

(5) $\psi =$ $EG\psi_0$

a. 计算由满足ψ_0的状态构成的有向图的强连通分量
b. 将所有非平凡强连通分量中的状态标上ψ
c. $L'(s) = L'(s) \cup \{\psi\}$ if $\psi_0 \in L'(s)$ and $\exists s'.(s \rightarrow s')$ and $\psi \in L'(s')$

标号算法

M=(S,R,I,L)

M,s |= φ iff $\varphi \in L'(s)$

M |= φ iff $\varphi \in L'(s)$ for all $s \in I$

不动点算法

M=(S,R,I,L)

φ

M |= φ ?

将 φ 的子公式对应到S的集合

目标:

M,s |= φ　　iff　　s ∈ [[φ]]

不动点算法(1)

$ex(Z) = \{ s \mid s \rightarrow s' \text{ and } s' \in Z \}$

$[[p]] = \{ s \mid p \in L(s) \}$

$[[\neg\psi_0]] = S \setminus [[\psi_0]]$

$[[\psi_0 \vee \psi_1]] = [[\psi_0]] \cup [[\psi_1]]$

$[[EX\ \psi_0]] = ex([[\psi_0]])$

$[[EG\ \psi_0]] = \nu Z.([[\psi_0]] \cap ex(Z))$

$[[E(\psi_0\ U\ \psi_1)]] = \mu Z.([[\psi_1]] \cup ([[\psi_0]] \cap ex(Z)))$

不动点算法

M=(S,R,I,L)

$M,s \models \varphi$ iff $s \in [[\varphi]]$

$M \models \varphi$ iff $I \subseteq [[\varphi]]$

(II.b)符号模型检测

状态集合 -- 布尔公式

状态集合的运算 -- 布尔公式的运算

符号模型

AP上的符号模型

M=(V,ρ,Θ,N)

V={v1,…,vn} 布尔变量集合 --- S

ρ: {v1,…,vn,v1’,…,vn’} 上的布尔公式 --- R

Θ: {v1,…,vn} 上的布尔公式 --- I

N: AP→BF(V) --- L

符号模型检测：不动点算法

$M=(V,\rho,\Theta,N)$

φ

$M \models \varphi$?

将 φ 的子公式对应到V上的布尔公式

不动点算法(1)

$ex(Z) = \exists v1', \ldots, vn'.(\rho \wedge Z')$

$[[p]] = Np$

$[[\neg\psi_0]] = \neg[[\psi_0]]$

$[[\psi_0 \vee \psi_1]] = [[\psi_0]] \vee [[\psi_1]]$

$[[EX\ \psi_0]] = ex([[\psi_0]])$

$[[EG\ \psi_0]] = \nu Z.([[\psi_0]] \wedge ex(Z))$

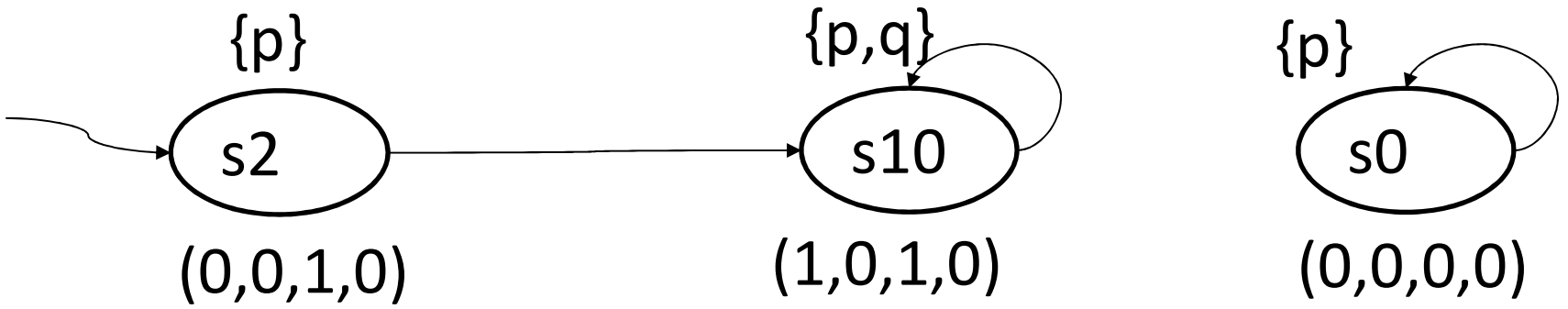
$[[E(\psi_0\ U\ \psi_1)]] = \mu Z.([[\psi_1]] \vee ([[\psi_0]] \wedge ex(Z)))$

不动点算法

$M=(V,\rho,\Theta,N)$

$M \models \varphi$
iff
$\Theta \rightarrow [[\varphi]]$
iff
$\Theta \wedge [[\neg \varphi]] = 0$

例子



M=(V,ρ,Θ,N) .....

V={v1,v2,v3,v4}

ρ = ((¬v2∧v3∧¬v4) ∧(v1'∧¬v2'∧v3'∧¬v4')) ∨

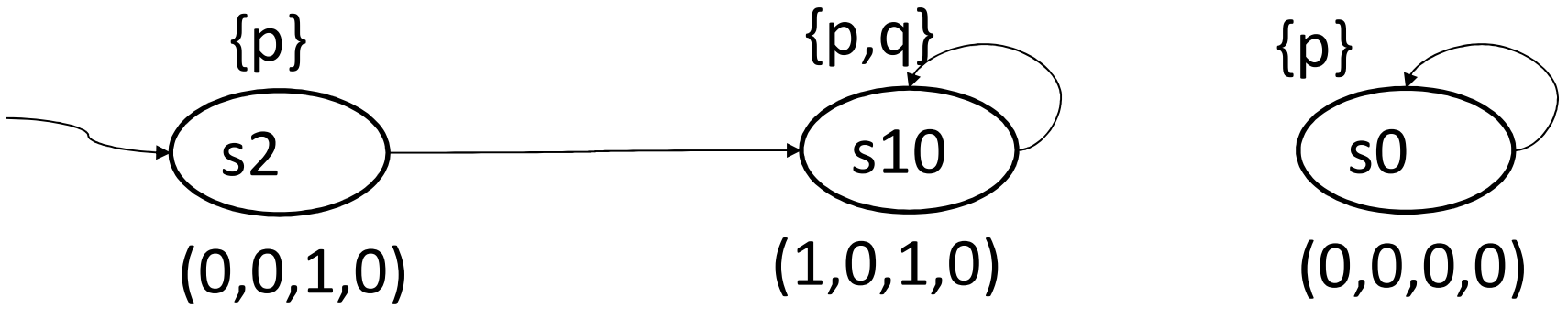
(¬(¬v2∧v3∧¬v4) ∧((v1'↔ v1)∧…∧(v4' ↔ v4))

Θ = (¬v1∧¬v2∧v3∧¬v4)

N(p) = (¬v1∧¬v2)∨(v1∧¬ v2);

N(q) = (¬v1∧v2)∨(v1∧¬v2)

例子



M |= AG(p∨q)

[[AG(p∨q)]] = [[¬E(true U ¬(p∨q))]] = ¬μZ.([[¬(p∨q)]]∨ex(Z))

f(Z) = [[¬(p∨q)]]∨ex(Z)

Z0=false

Z1= [[¬(p∨q)]]= ¬(N(p) ∨ N(q)) =

¬((¬v1∧¬v2)∨(v1∧¬v2) ∨ (¬v1∧v2)∨(v1∧¬v2)) = (v2∧v1)

Z2= (v2∧v1)

因此 [[AG(p∨q)]] = ¬(v2∧v1)

由于 Θ→¬(v2∧v1)，因而M |= AG(p∨q)。

不动点算法的问题及进一步考虑

优点：集合运算 – 逻辑运算

需要高效的：

布尔公式的表示方法

量词消去算法

布尔公式等价算法

布尔公式的OBDD表示

(II.c)限界模型检测 / 限界正确性检查

k-路径: 长度为k+1的路径

M = (S,R,I,L)

M_k = (S,P_k,I,L) 其中 P_k为M的所有k-路径的集合

rs(π): 路径π中有相同状态

CTL

考虑NNF公式：

$$\Phi == p \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \neg p \mid$$
$$AX\,\Phi \mid A(\Phi\ R\ \Phi) \mid A(\Phi\ U\ \Phi) \mid$$
$$EX\,\Phi \mid E(\Phi\ R\ \Phi) \mid E(\Phi\ U\ \Phi)$$

语义：$M,u \models \phi$

$M,u \models p$, if $p \in AP$ and $p \in L(u)$

$M,u \models \neg p$, if $p \in AP$ and $p \notin L(u)$

$M,u \models \phi \vee \psi$, if $M,u \models \phi$ or $M,u \models \psi$

$M,u \models \phi \wedge \psi$, if $M,u \models \phi$ and $M,u \models \psi$

$M,u \models A\ \psi$, if for every path π of u, $(M,\pi \models \psi)$

$M,u \models E\ \psi$, if there is a path π of u, $(M,\pi \models \psi)$

$M,\pi \models X\ \phi$, if $M,\pi_1 \models \phi$

$M,\pi \models \phi U \psi$, if $\exists i \geq 0$, $M,\pi_i \models \psi$ and $\forall j<i$, $M,\pi_j \models \phi$

$M,\pi \models \phi R \psi$, if $\forall i \geq 0$, $(\forall j<i, M,\pi_j \models \phi) \rightarrow M,\pi_i \models \psi$

语义：F,G

$M,\pi \models F\psi$, if $\exists i \geq 0$, $M,\pi_i \models \psi$

$M,\pi \models G\psi$, if $\forall i \geq 0$, $M,\pi_i \models \psi$

限界语义： $M,u \models_k \phi$

$M,u \models_k p$, if $p \in AP$ and $p \in L(u)$

$M,u \models_k \neg p$, if $p \in AP$ and $p \notin L(u)$

$M,u \models_k \phi \vee \psi$, if $M,u \models_k \phi$ or $M,u \models_k \psi$

$M,u \models_k \phi \wedge \psi$, if $M,u \models_k \phi$ and $M,u \models_k \psi$

$M,u \models_k A\ \psi$, if for every k-path π of u, $(M,\pi \models_k \psi)$

$M,u \models_k E\ \psi$, if there is a k-path π of u, $(M,\pi \models_k \psi)$

$M,\pi \models_k X\ \phi$, if $k \geq 1$ and $M,\pi_1 \models_k \phi$

$M,\pi \models_k \phi U \psi$, if $\exists i \leq k$, $M,\pi_i \models_k \psi$ and $\forall j<i$, $M,\pi_j \models_k \phi$

$M,\pi \models_k \phi R \psi$, if $\forall i \leq k$, $(\forall j<i, M,\pi_j \models_k \phi) \rightarrow M,\pi_i \models_k \psi$ and

$(rs(\pi)$ or $\exists i \leq k, (M,\pi_i \models_k \phi))$

限界语义：F,G

$M,\pi \models_k F\psi$, if $\exists i \leq k$, $M,\pi_i \models_k \psi$

$M,\pi \models_k G\psi$, if $\forall i \leq k$, $(M,\pi_i \models_k \psi)$ and $rs(\pi)$

限界语义： $M \models_k \phi$

Definition

$M \models_k \phi$ if $M,u \models_k \phi$ for every initial state u.

Soundness

For every $i \geq 0$, if $M \models_i \varphi$, then $M \models \varphi$.

Completeness

If $M \models \varphi$, then there is a $i \geq 0$ such that if $M \models_i \varphi$.

限界正确性检查(BCC)

1. k=0;
2. if $M|=_k \varphi$, then report $M|= \varphi$;
3. if $M,s|=_k \neg\varphi$ for some $s\in I$, then report $M|\not= \varphi$;
4. k=k+1; goto step 2;

限界正确性检查的问题及进一步考虑

优点：局部检查CTL性质的正确性

需要高效的：

检查 $\forall s \in I,\ M,s|=_k \varphi$的算法
检查 $\exists s \in I,\ M,s|=_k \varphi$的算法

基于QBF的 CTL 限界正确性检查
基于SAT的 ACTL 限界正确性检查

k-模型： $\Theta(V^0) \wedge \rho(V^0,V^1) \wedge \ldots \wedge \rho(V^{k-1},V^k)$

(II.d)公平约束下的CTL模型检测

标号算法

标号公平Kripke结构

模型

$M=(S,R,I,L,\Phi)$

$\Phi=\{\Phi 1,\ldots, \Phi k\}$

标号算法

$M=(S,R,I,L,\Phi)$

a. 计算由满足Φ的状态构成的有向图的强连通分量
b. 将所有非平凡强连通分量中的状态标上Φ
c. 将所有能到达非平凡强连通分量的状态标上Φ

$M'= (S',R',I',L^*)$ – 考虑模型的公平状态部分

$S'= \{ s \mid \Phi \in L'(s) \}$为公平状态的集合
$R' = R \cap (S' \times S')$
$I' = I \cap S'$
$L^* = L \cap (S' \times 2^{AP})$

标号算法(a)

只考虑$s \in S'$:

$L'(s) := \{\}$ for all $s \in$

考虑φ的长度为1的子公式p

$L'(s) := L'(s) \cup \{p\}$ if $p \in L^*(s)$

标号算法(b)

考虑φ的长度为k+1的子公式ψ

(1) $\psi = \neg\psi_0$

$L'(s) = L'(s) \cup \{\psi\}$ if $\psi_0 \notin L'(s)$

(2) $\psi = \psi_0 \vee \psi_1$

$L'(s) = L'(s) \cup \{\psi\}$ if $\psi_0 \in L'(s)$ or $\psi_1 \in L'(s)$

(3) $\psi = EX\psi_0$

$L'(s) = L'(s) \cup \{\psi\}$ if $\exists s'.(s \rightarrow s')$ and $\psi_0 \in L'(s')$

标号算法(c)

(4) $\psi =$ $E(\psi_0 U \psi_1)$

$L'(s) = L'(s) \cup \{\psi\}$ if $\psi_1 \in L'(s)$
$L'(s) = L'(s) \cup \{\psi\}$ if $\psi_0 \in L'(s)$ and $\exists s'.(s \rightarrow s')$ and $\psi \in L'(s')$

(5) $\psi =$ $EG\psi_0$

a. 计算满足ψ_0的状态集合的满足Φ的强连通分量
b. 将所有非平凡强连通分量中的状态标上ψ
c. $L'(s) = L'(s) \cup \{\psi\}$ if $\psi_0 \in L'(s)$ and $\exists s'.(s \rightarrow s')$ and $\psi \in L'(s')$

标号算法

$M=(S,R,I,L,\Phi)$

$M'= (S',R',I',L^*)$

$M \models \varphi$
iff
$\varphi \in L'(s)$ for all $s \in I'$

不动点算法

符号模型的不动点算法

公平约束下的符号模型

符号模型

$M=(V,\rho,\Theta,N,\Phi)$

$V=\{v1,\ldots,vn\}$ 布尔变量集合

ρ: $\{v1,\ldots,vn,v1',\ldots,vn'\}$ 上的布尔公式

Θ: $\{v1,\ldots,vn\}$ 上的布尔公式

N: AP→BF(V)

$\Phi=\{\Phi1,\ldots, \Phi k\}$

公平约束下的符号模型检测

$Fair = \nu Z.(\wedge\{ ex[[EF (Z \wedge \Phi j)]] \mid j=1,...,k \})$

$[[EX_F \varphi]] = ex([[\varphi]] \wedge Fair)$

$[[E(\varphi U_F \psi)]] = [[E(\varphi U (\psi \wedge Fair))]]$

$[[EG_F \varphi]] = \nu Z.([[\varphi]] \wedge (\wedge\{ ex[[E(\varphi U (Z \wedge \Phi j))]] \mid j=1,...,k \}))$

不动点算法

$[[p]] = Np$

$[[\neg\psi_0]] = \neg[[\psi_0]]$

$[[\psi_0 \vee \psi_1]] = [[\psi_0]] \vee [[\psi_1]]$

$[[EG_F\, \varphi\,]] = \nu Z.([[\varphi]] \wedge (\, \wedge\{\, ex\, [[E(\varphi\, U\, (Z \wedge \Phi j))\,]] |\, j=1,\ldots,k\, \}\,))$

$[[EX_F\, \varphi]] = ex\, ([[\varphi]] \wedge Fair)$

$[[E(\varphi\, U_F\, \psi)]] = [[E(\varphi\, U\, (\psi \wedge Fair))]]$

$Fair = [[EG_F\, (true)]]$

$[[E(\psi_0\, U\, \psi_1)]] = \mu Z.([[\psi_1]] \vee ([[\psi_0]] \wedge ex(Z)\,))$

不动点算法

$M=(V,\rho,\Theta,N,\Phi)$

ψ^* 为 ψ 的具有公平约束的版本(X,G,U替换为X_F,G_F,U_F)

$M \models \varphi$
iff
$\Theta \wedge [[(\neg \varphi)^*]] = false$

(III) LTL模型检测

- 显式状态模型检测
- 符号模型检测
- 限界模型检测
- 公平约束下的LTL模型检测

(III.a)显式状态模型检测

M=(S,R,I,L)

φ

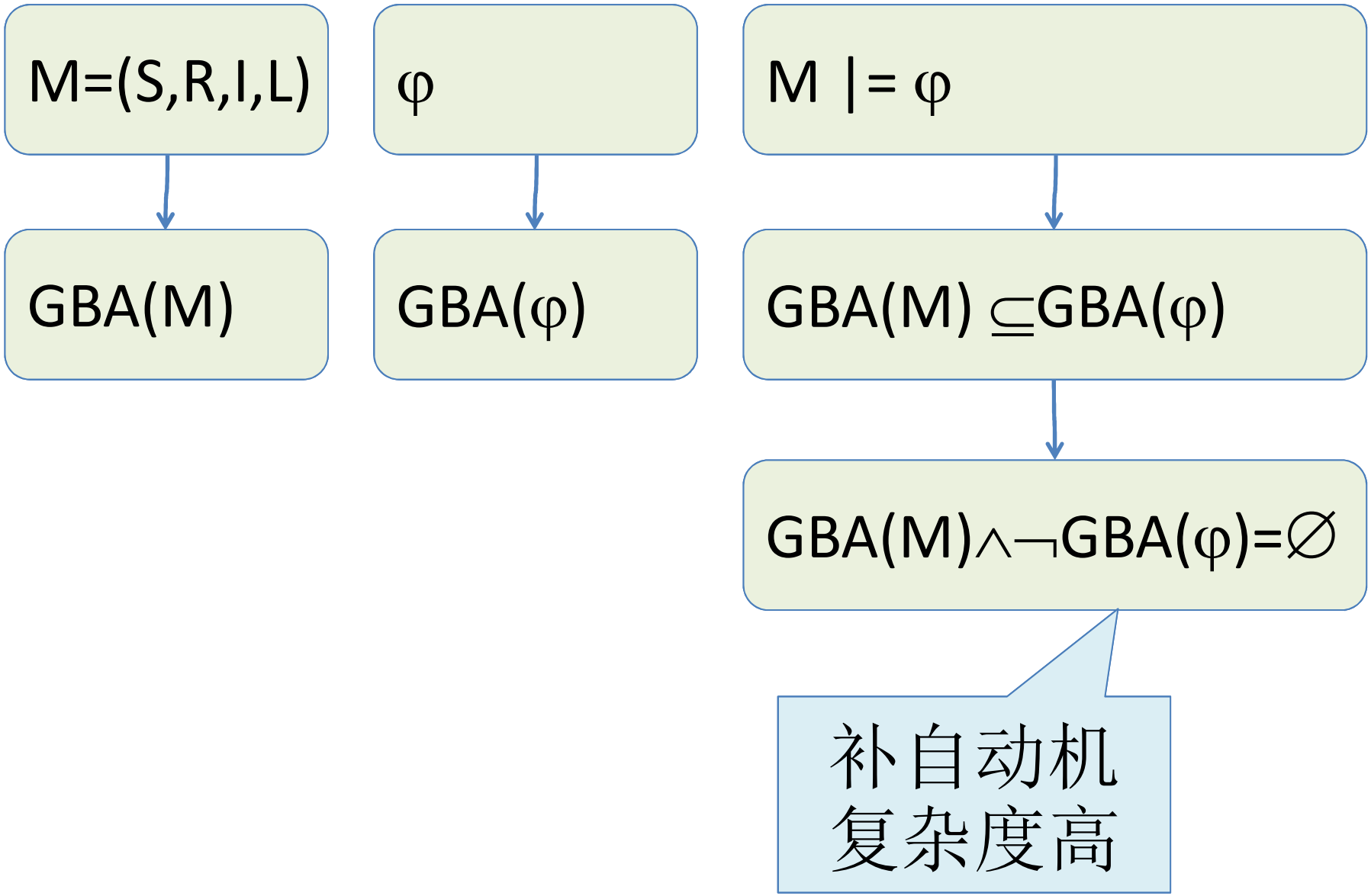
考虑NNF公式：

Φ == p | $\Phi \wedge \Phi$ | $\Phi \vee \Phi$ | $\neg$p | X Φ | (Φ R Φ) | (Φ U Φ)

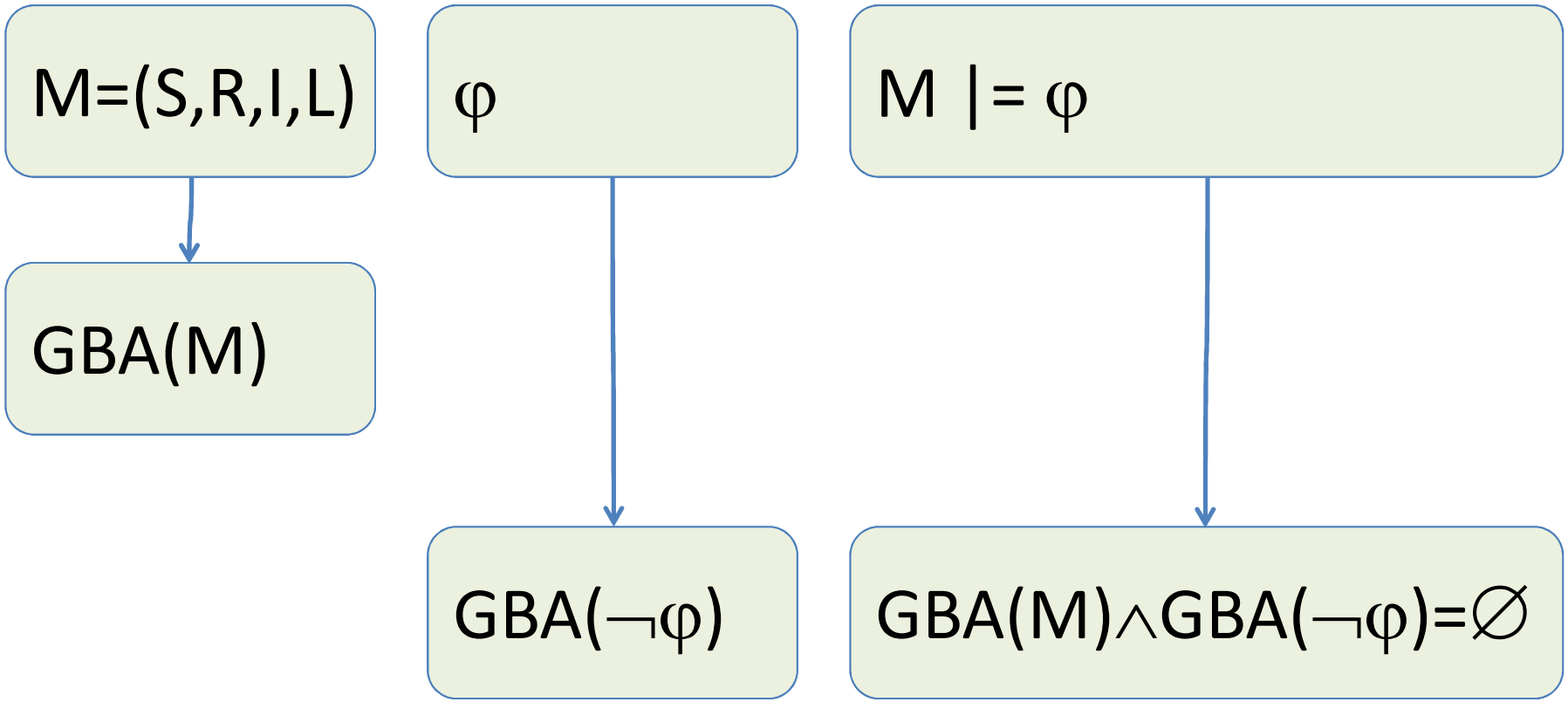
基于自动机的模型检测(1)

基于GBA

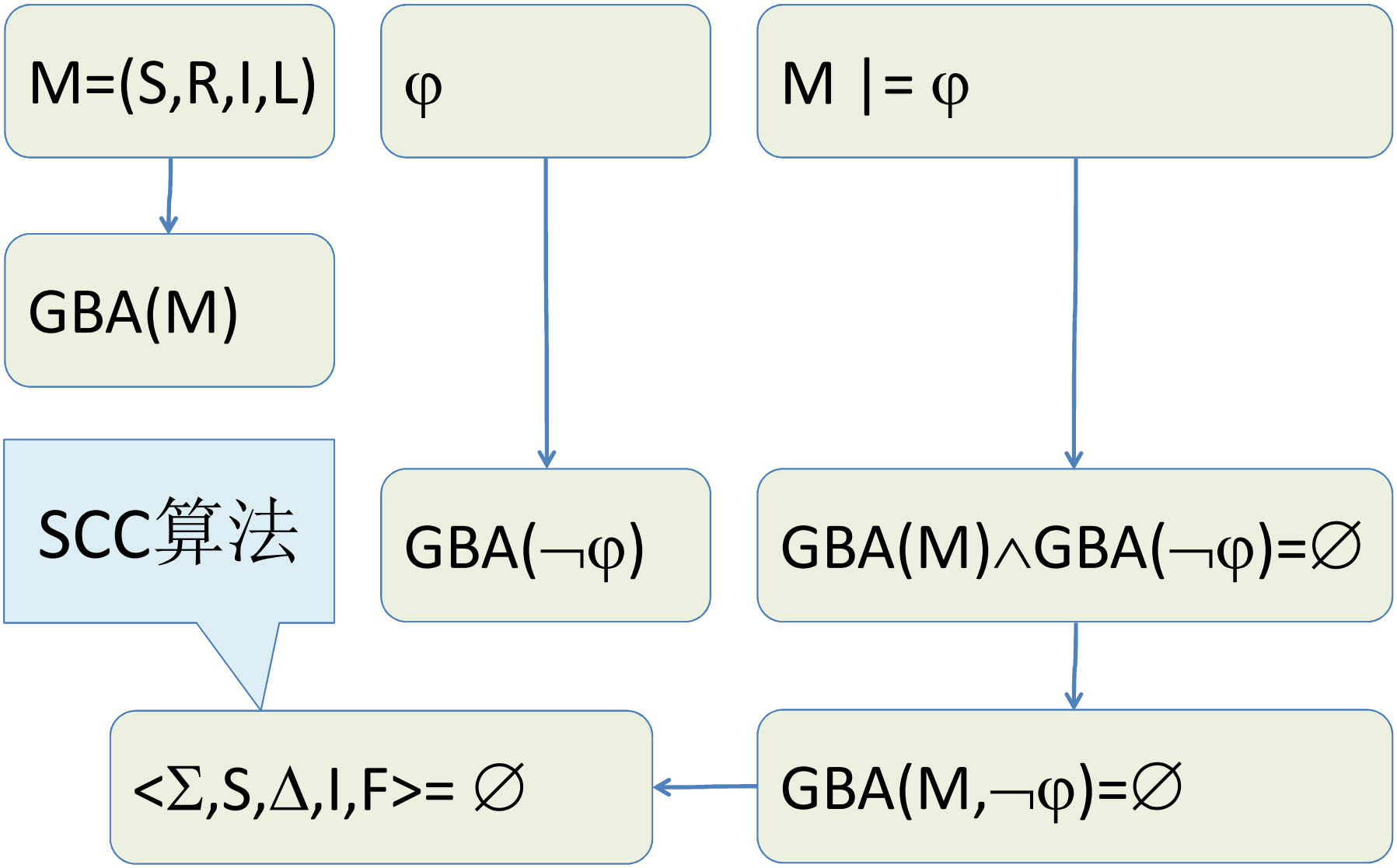
基于自动机的模型检测(1a)



基于自动机的模型检测(1b)



基于自动机的模型检测(1b)

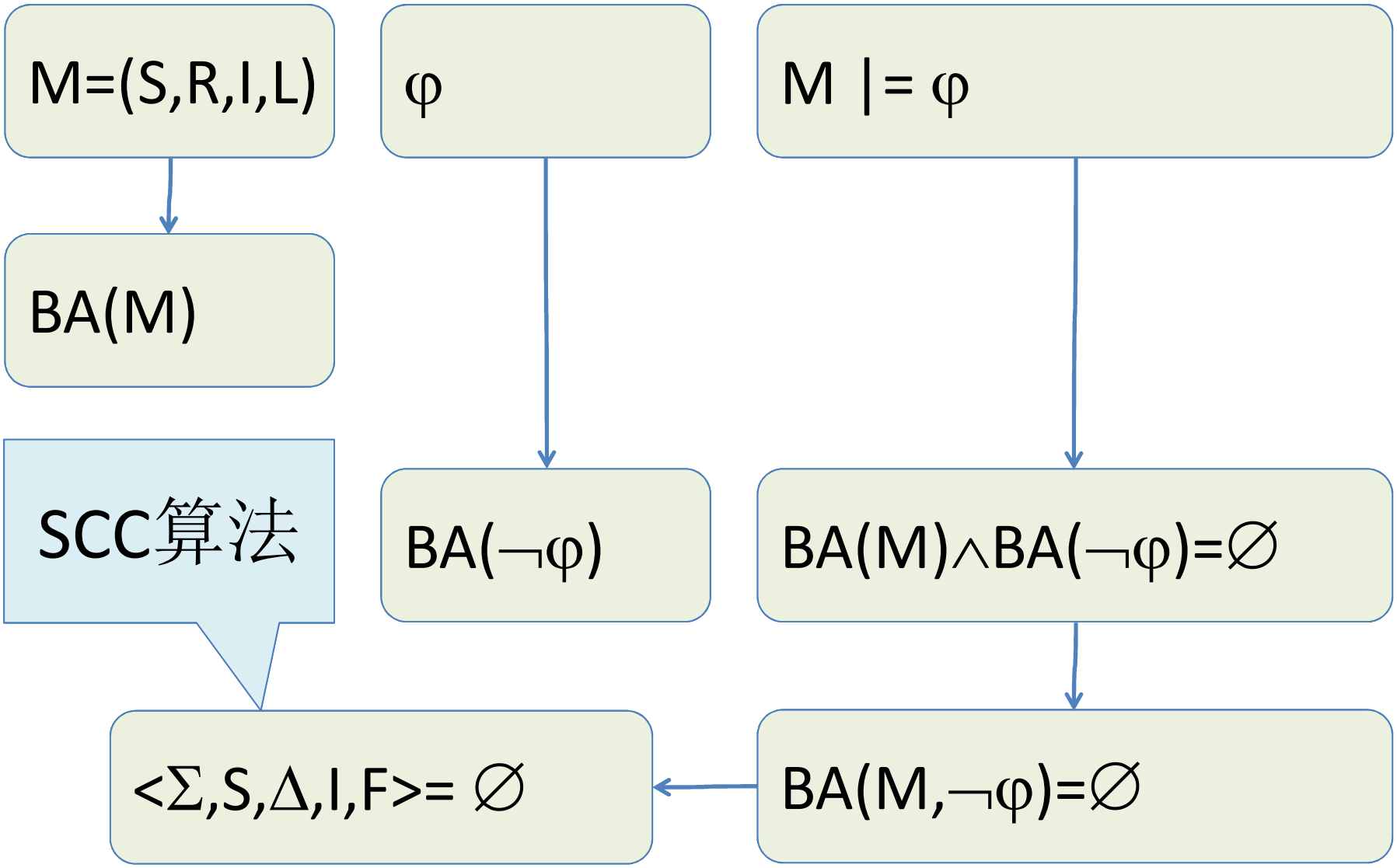


基于自动机的模型检测(2)

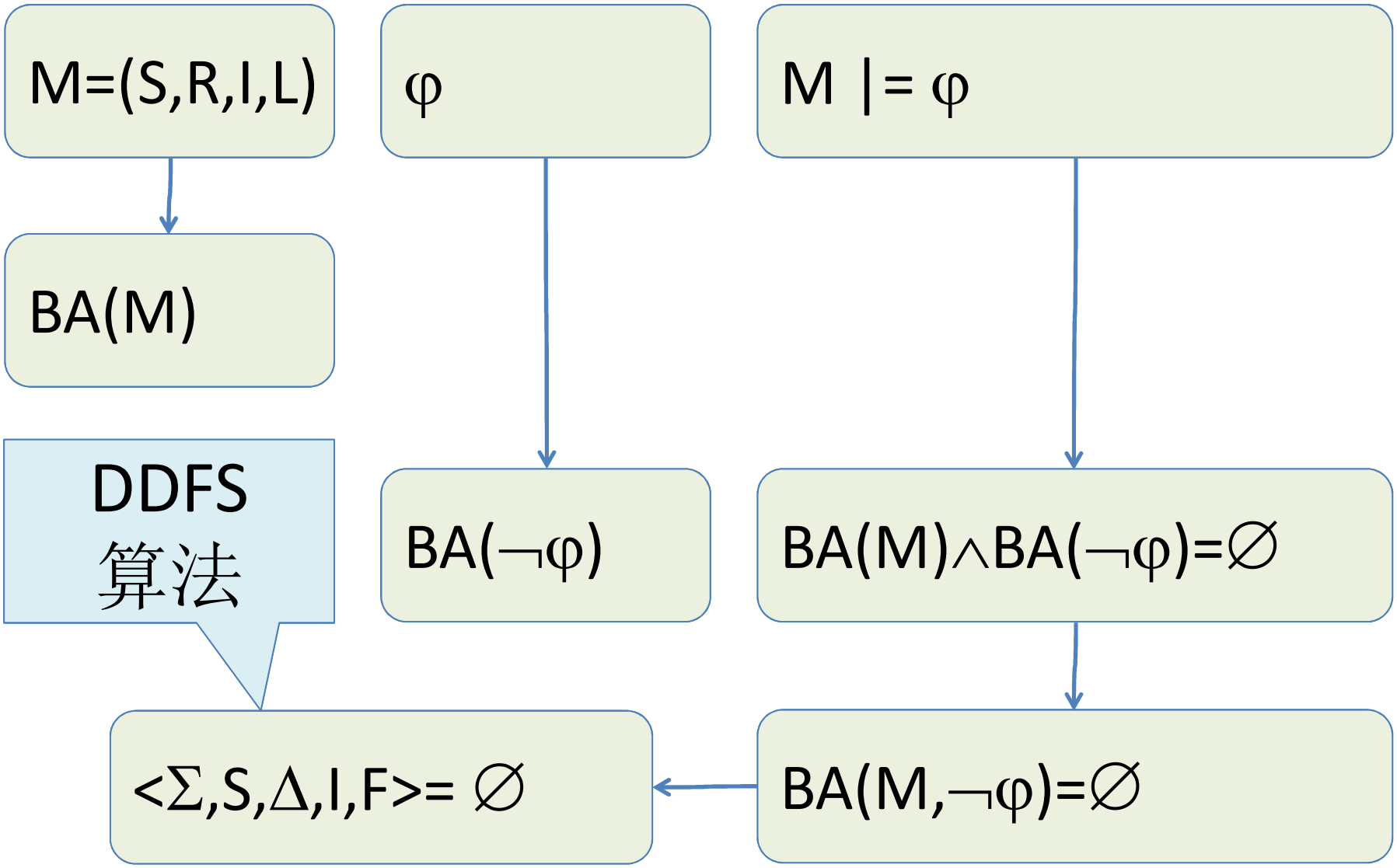
On-the-fly 技术

基于BA

基于自动机的模型检测(2a)



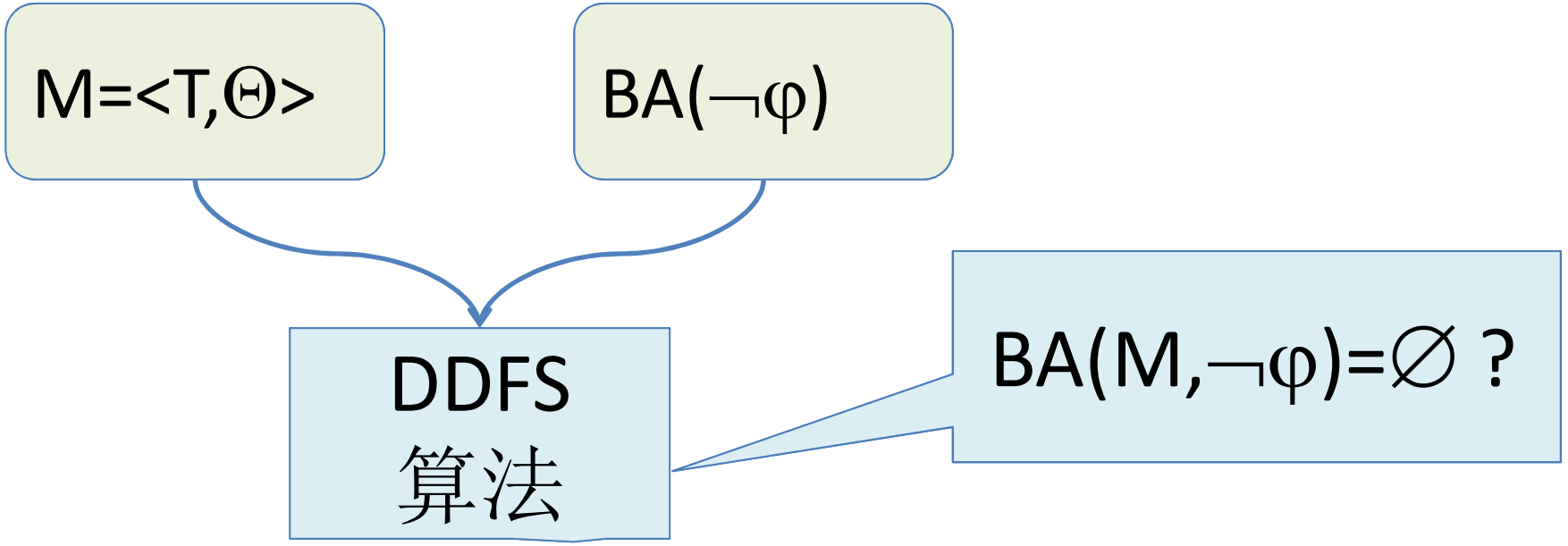
基于自动机的模型检测(2b)



基于自动机的模型检测(2)

SCC算法：全局

DDFS算法：适用于动态(on-the-fly)模型检测



进一步考虑

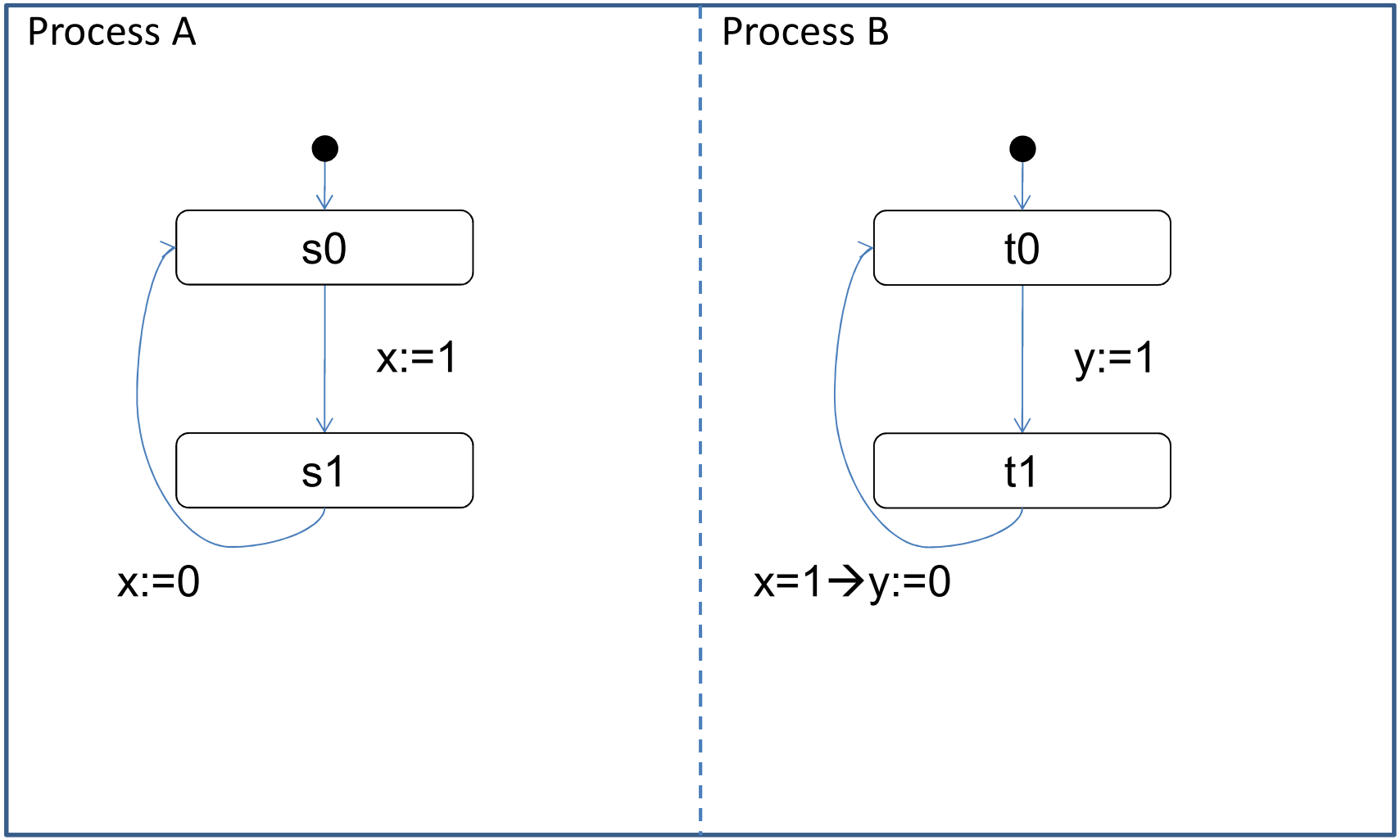
优点：对部分问题，可快速查到问题

构造BA(φ)的高效算法

Partial-order reduction

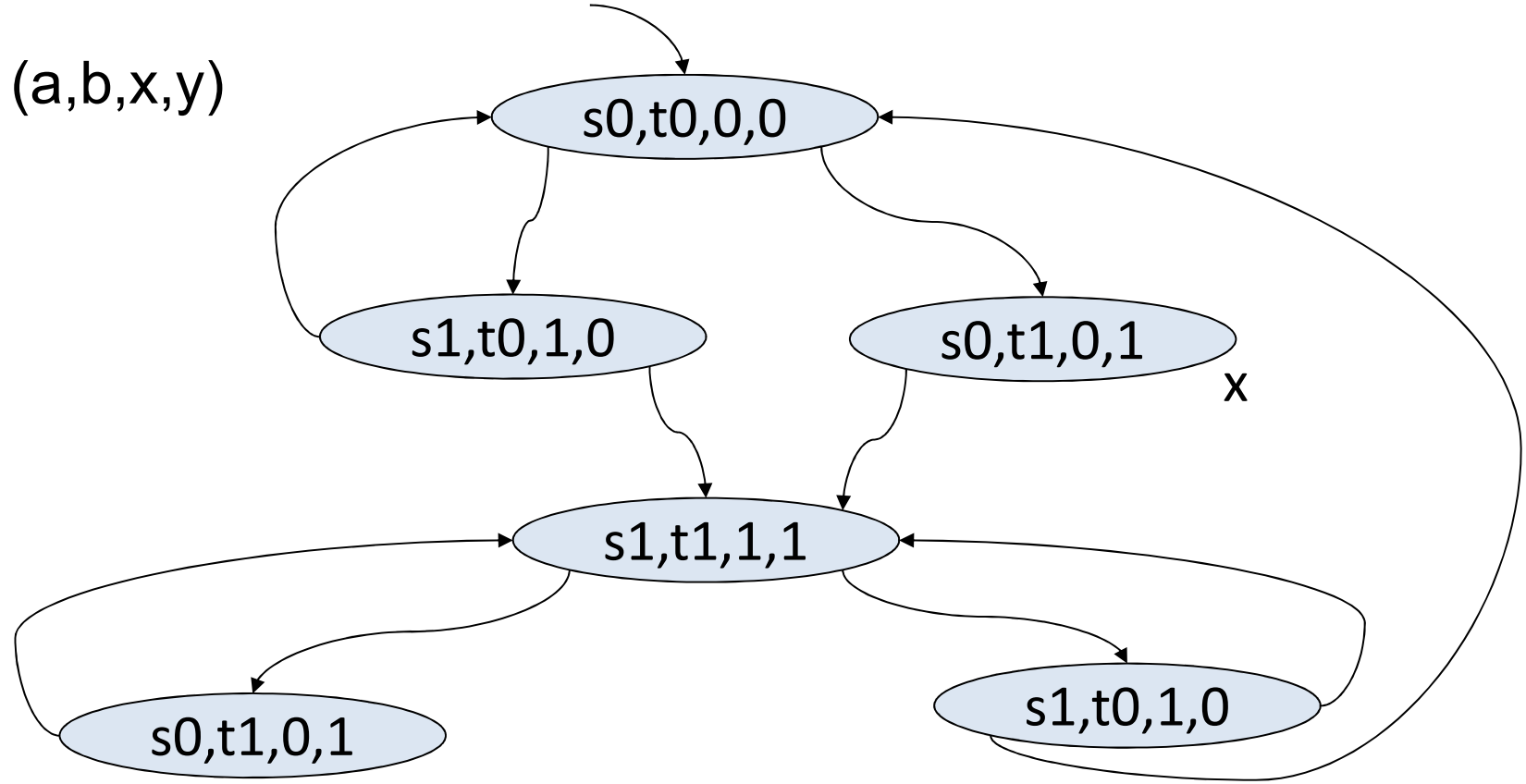
- Stuttering equivalent properties
- LTL \ X

例子



Kripke 结构

AP={x=1}, φ∈LTL\X



001 …..
011 ….

(III.b)符号模型检测

符号模型检测：符号模型

符号模型

$M=(V,\rho,\Theta,N,\Phi)$

$V=\{v1,\ldots,vn\}$ 布尔变量集合

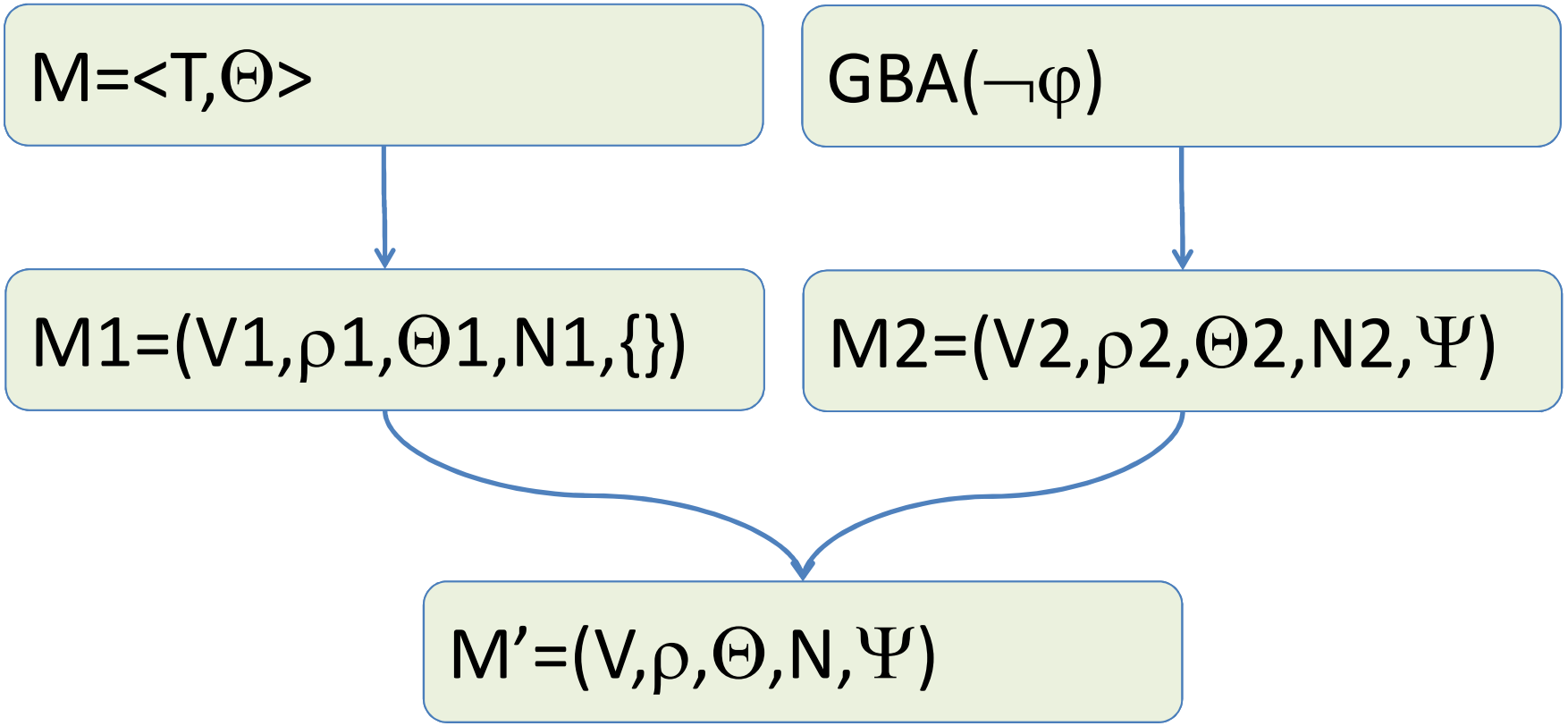
ρ: $\{v1,\ldots,vn,v1',\ldots,vn'\}$ 上的布尔公式

Θ: $\{v1,\ldots,vn\}$ 上的布尔公式

N: $\{ Np \mid p \in AP \}$, Np 为$\{v1,\ldots,vn\}$上的布尔公式

$\Phi=\{\Phi1,\ldots, \Phi k\}$

符号模型检测



$M \models \varphi$ iff $M' \not\models EG(true)$

进一步考虑

$M'=(V,\rho,\Theta,N,\Psi)$ 的高效构造

(III.c)限界模型检测

k路径π：　　长度为k+1的路径。

(k,l)环π：　　k路径 且 $R(\pi_k, \pi_l)$成立。

存在 计算π使得$M,\pi \models \phi$
当且仅且当存在l起点的 (k,l)环π'使得
$M, \pi_0' \ldots \pi_{l-1}' (\pi_l' \ldots \pi_k')^\omega \models \phi$ 且 $k \leq |M| \times 2^{|\phi|}$.

$M \not\models \phi$
当且仅且当存在l起点的 (k,l)环π使得
$M, \pi_0 \ldots \pi_{l-1} (\pi_l \ldots \pi_k)^\omega \models \neg\phi$

LTL

考虑NNF-LTL：

Φ == p | $\Phi \wedge \Phi$ | $\Phi \vee \Phi$ | $\neg$p | X Φ | GΦ | Φ U Φ

(正常)语义：$M,\pi \models \phi$

$M,\pi \models p$, if $p \in AP$ and $p \in L(\pi_0)$

$M,\pi \models \neg\phi$, if $M,\pi \not\models \phi$

$M,\pi \models \phi \vee \psi$, if $M,\pi \models \phi$ or $M,\pi \models \psi$

$M,\pi \models \phi \wedge \psi$, if $M,\pi \models \phi$ and $M,\pi \models \psi$

$M,\pi \models X\,\phi$, if $M,\pi^1 \models \phi$

$M,\pi \models \phi U \psi$, if $\exists i \geq 0$, $M,\pi^i \models \psi$ and $\forall j < i$, $M,\pi^j \models \phi$

$M,\pi \models G\psi$, if $\forall i \geq 0$, $M,\pi^i \models \psi$

Existential Model Checking Problem

$M \models_E \phi$, if there a computation π such that $M,\pi \models \phi$

限界语义(non-loop)： $M, \pi \models_k \phi$

$M,\pi \models_k^m p$, if $p \in AP$ and $p \in L(\pi_m)$

$M,\pi \models_k^m \neg p$, if $p \in AP$ and $p \notin L(\pi_m)$

$M,\pi \models_k^m \phi \vee \psi$, if $M,\pi \models_k^m \phi$ or $M,\pi \models_k^m \psi$

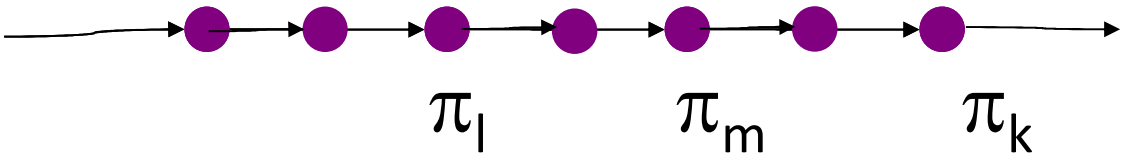
$M,\pi \models_k^m \phi \wedge \psi$, if $M,\pi \models_k^m \phi$ and $M,\pi \models_k^m \psi$

$M,\pi \models_k^m X \phi$, if $k \geq m+1$ and $M,\pi \models_k^{m+1} \phi$

$M,\pi \models_k^m \phi U \psi$, if $\exists m \leq i \leq k$, $M,\pi \models_k^i \psi$ and $\forall m \leq j < i$, $M,\pi \models_k^j \phi$

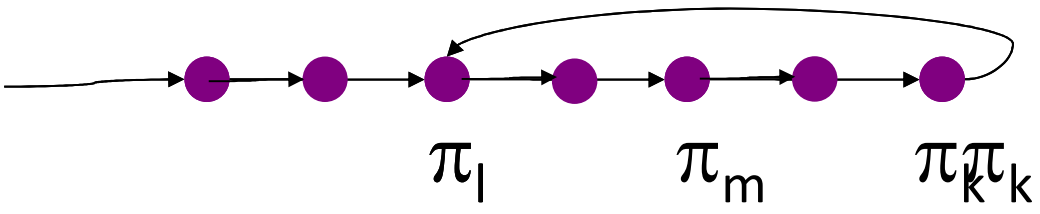
$M,\pi \models_k^m G\psi$, if false

$M,\pi \models_{k,-1} \phi$ if $M,\pi \models_k^0 \phi$



限界语义(loop)： $M, \pi \models_k \phi$

(k,l)-loop: $\pi = \pi_0 \ldots \pi_{l-1}(\pi_l \ldots \pi_k)^\omega$



限界语义(loop)： $M, \pi \models_k \phi$

$M,\pi \models_k^{l,m} p$, if $p \in AP$ and $p \in L(\pi_m)$

$M,\pi \models_k^{l,m} \neg p$, if $p \in AP$ and $p \notin L(\pi_m)$

$M,\pi \models_k^{l,m} \phi \vee \psi$, if $M,\pi \models_k^{l,m} \phi$ or $M,\pi \models_k^{l,m} \psi$

$M,\pi \models_k^{l,m} \phi \wedge \psi$, if $M,\pi \models_k^{l,m} \phi$ and $M,\pi \models_k^{l,m} \psi$

$M,\pi \models_k^{l,m} X\,\phi$, if $k \geq m+1$ and $M,\pi \models_k^{l,m+1} \phi$,

or $k=m$ and $M,\pi \models_k^{l,l} \phi$

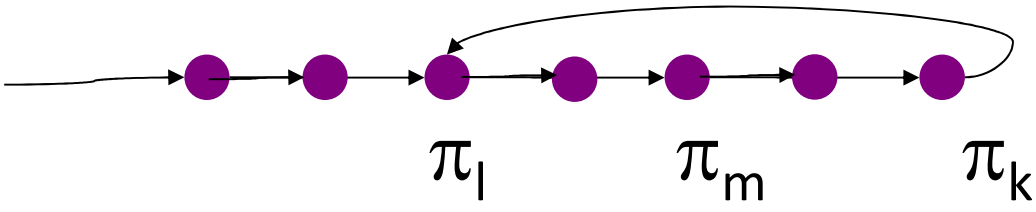
$M,\pi \models_k^{l,m} \phi U \psi$, if $\exists m \leq i \leq k$, $M,\pi \models_k^{l,i} \psi$ and $\forall m \leq j < i$, $M,\pi \models_k^{l,j} \phi$,

or $\forall m \leq j \leq k$, $M,\pi \models_k^{l,j} \phi$ and

$\exists l \leq i < m$, $M,\pi \models_k^{l,i} \psi$ and $\forall l \leq j < i$, $M,\pi \models_k^{l,j} \phi$

$M,\pi \models_k^{l,m} G\psi$, if $\forall \min(l,m) \leq i \leq k$, $M,\pi \models_k^{l,i} \psi$

$M,\pi \models_{k,l} \phi$ if $M,\pi \models_k^{l,0} \phi$



限界语义： $M, \pi \models_k \phi$

DEFINITION

$M,\pi \models_k \phi$, if $M,\pi \models_{k,-1} \phi$ or $M,\pi \models_{k,l} \phi$ for some $l \in \{0,\ldots,k\}$

DEFINITION

$M \models_{E,k} \phi$, if $M,\pi \models_k \phi$ for some computation π.

LEMMA (Soundness)

If $M \models_{E,k} \phi$, then $M \models_E \phi$.

LEMMA (Completeness)

If $M \models_E \phi$, then $M \models_{E,k} \phi$ for some $k \geq 0$.

限界语义

THEOREM

$M \models_E \phi$ iff

there is $k \geq 0$ such that $M \models_{E,k} \phi$

Corollary

$M \not\models \phi$ iff

there is $k \geq 0$ such that $M \models_{E,k} \neg\phi$

完备阈值

k是$M \models \varphi$的完备阈值，当且仅当
若$M \not\models_{E,k} \neg\varphi$，则对$k' \geq k$，$M \not\models_{E,k'} \neg\varphi$

设lct是$M \models \varphi$的最小完备阈值:
若 $b \geq lct$且$M \not\models_{E,b} \neg\varphi$, 则 $M \models \varphi$.

(小模型定理)
最小完备阈值
(least completeness threshold $\leq |M| * 2^{|\varphi|}$)

限界模型检测(BMC)

Let b be a completeness threshold for $M \models \varphi$ or infinity.

1. k=0;
2. if $M \models_{E,k} \neg\varphi$, then report $M \models \varphi$;
3. if k=b, then report $M \models \varphi$;
4. k=k+1; goto step 2;

限界模型检测的问题及进一步考虑

完备阈值(completeness threshold <= $|M|*2^{|\varphi|}$)

需要高效检查$M|=_{E,k}\varphi$的算法

基于SAT的 LTL限界模型检测

(III.d)公平约束下的LTL模型检测

标号公平Kripke结构模型

$M=(S,R,I,L,\Phi)$

$\Phi=\{\Phi 1,\ldots, \Phi k\}$

模型检测

$M=(S,R,I,L,\Phi)$

$M'=(S,R,I,L)$

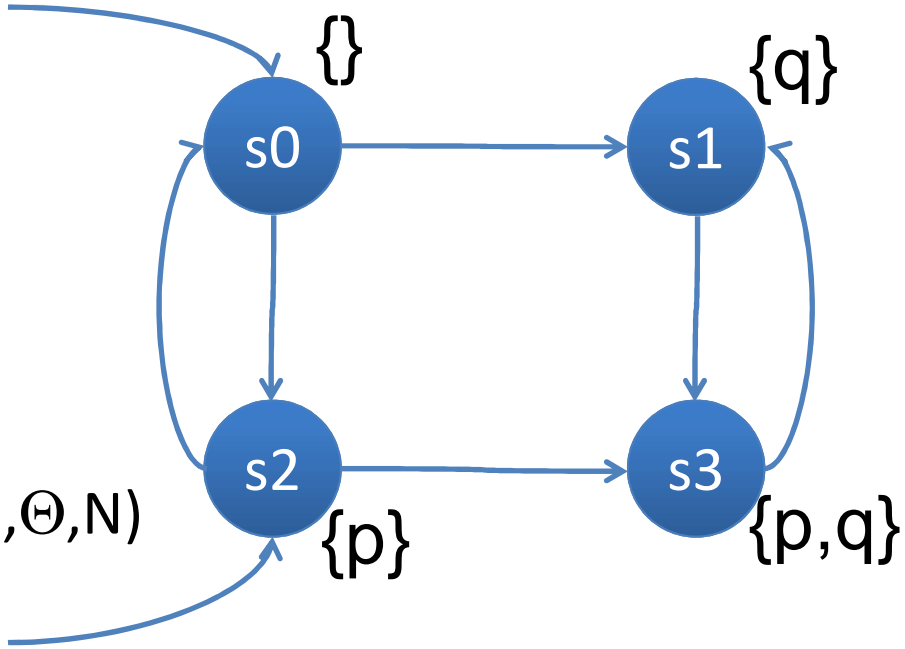
$M \models \varphi$
iff
$M' \models (GF(\Phi 1) \wedge \ldots \wedge GF(\Phi k)) \rightarrow \varphi$

(IV) Summary

- 符号模型
- CTL模型检测方法
- LTL模型检测方法

练习题：

（1）
给定如图所示标号Kripke模型。
将其转换成符号模型，写清楚
符号模型的各分量。



（2）
给定AP={p,q}上的符号模型M=(V,ρ,Θ,N)
其中

V= {v1,v2},

ρ = $(\neg v1 \wedge \neg v2 \wedge (v1' \leftrightarrow \neg v2')) \vee (\neg v1 \wedge v2 \wedge (v1' \wedge v2')) \vee$
$(v1 \wedge \neg v2 \wedge (v1' \leftrightarrow v2')) \vee (v1 \wedge v2 \wedge (\neg\, v1' \wedge v2'))$,

Θ = $(\neg v2)$,

N(p)= (v1),

N(q) = (v2)。

用符号模型检测方法计算[[EG(p ∨q)]]并说明模型是否满足EG(p∨q)。

课程总结

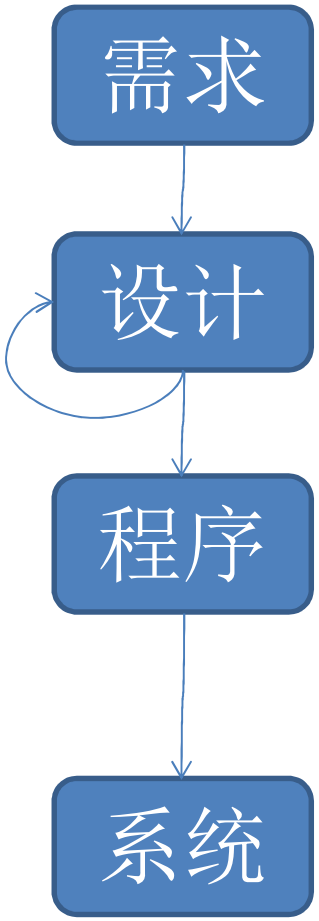
中国科学院软件研究所
计算机科学国家重点实验室
张文辉
http://lcs.ios.ac.cn/~zwh/

内容

- 课程的关注点(例子)
- 课程主要内容：模型、逻辑、推理验证、模型检测

(I)课程的关注点(例子)

形式化方法：软件正确性



需求 ➔ 一致

设计 ➔ 设计
设计 ➔ 需求

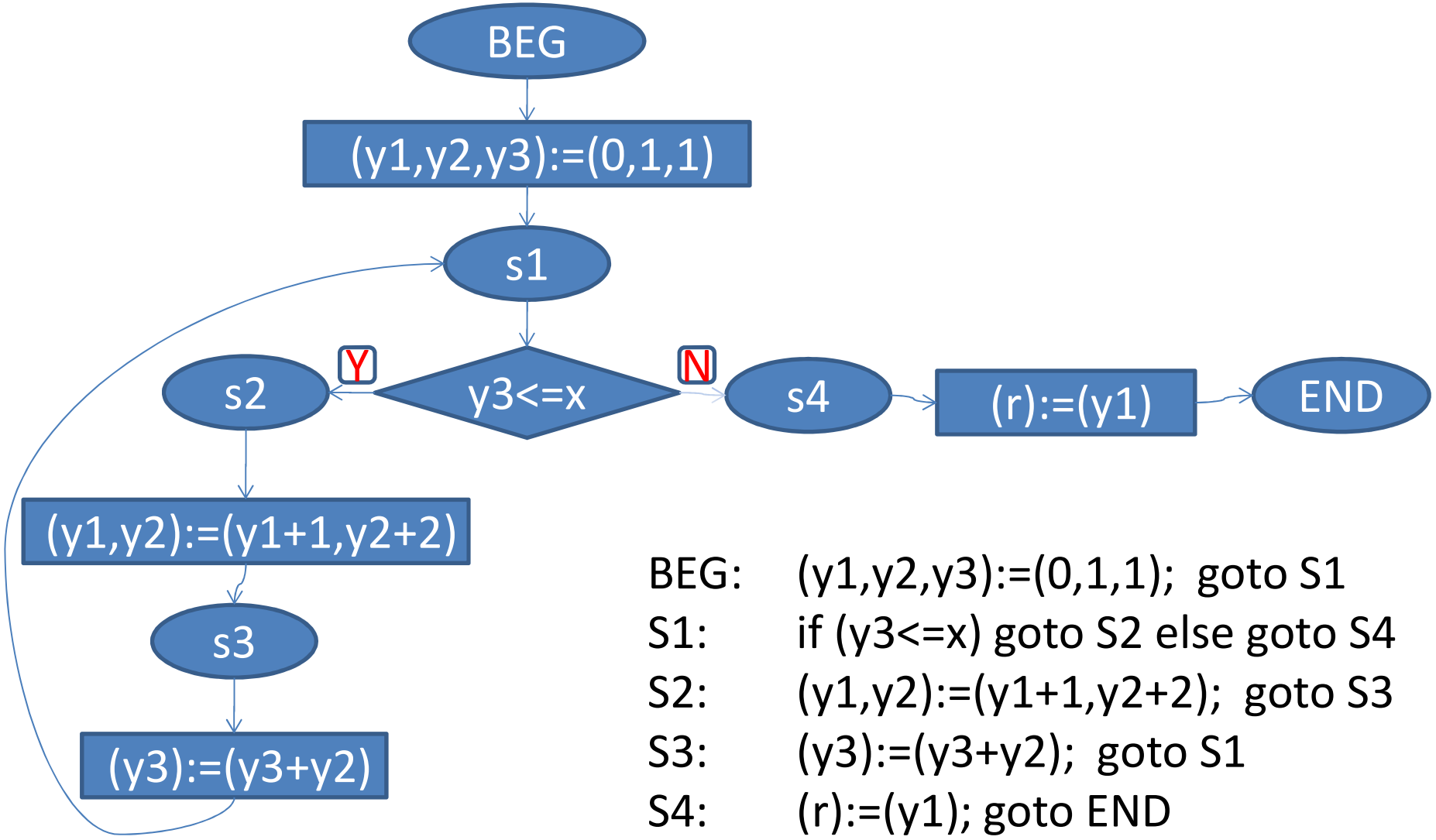
程序 ➔ 程序
程序 ➔ 设计
程序 ➔ 需求

系统 ➔ 需求

例1 -整树平方根：需求

输入: r 满足 x>=0
输出: r 满足 x>=r*r ∧ x<(r+1)*(r+1)

例1 -整树平方根：设计



BEG: (y1,y2,y3):=(0,1,1); goto S1
S1: if (y3<=x) goto S2 else goto S4
S2: (y1,y2):=(y1+1,y2+2); goto S3
S3: (y3):=(y3+y2); goto S1
S4: (r):=(y1); goto END

例1 -整树平方根：设计

迁移关系		初始状态
pc=BEG →	(y1,y2,y3,pc):=(0,1,1,S1)	pc=BEG
pc=S1∧(y3<=x) →	(pc):=(S2)	
pc=S1∧¬ (y3<=x) →	(pc):=(S4)	
pc=S2→	(y1,y2,pc):=(y1+1,y2+2,S3)	
pc=S3→	(y3,pc):=(y3+y2,S1)	
pc=S4→	(r,pc):=(y1,END)	

终止性：　　AF(pc=END)

部分正确性：　　AG(pc=END → (x>=r*r ∧ x<(r+1)*(r+1)))

例1 -整树平方根：模型检测

例1 -整树平方根(有穷状态)

迁移关系		初始状态
pc=BEG →	(y1,y2,y3,pc):=(0,1,1,S1)	pc=BEG∧
pc=S1∧(y3<=x) →	(pc):=(S2)	x<=15 ∧
pc=S1∧¬ (y3<=x) →	(pc):=(S4)	y1=0 ∧
pc=S2→	(y1,y2,pc):=(y1+1,y2+2,S3)	y2=0 ∧
pc=S3→	(y3,pc):=(y3+y2,S1)	y3=0 ∧
pc=S4→	(r,pc):=(y1,END)	r=0

终止性： AF(pc=END)

部分正确性： AG(pc=END → (x>=r*r ∧ x<(r+1)*(r+1)))

符号模型检测(不动点算法)

```
verds  -b -ck 1 isqrt15.vvm
VERSION:    verds 1.46 - JAN 2015
FILE:       isqrt15.vvm
INFO:       int=i0
PROPERTY:   A G (! (pc = 5 )| ((x } (r * r ))& (x < ((r + 1 )* (r + 1 )))))
check:   0
----------
check:   1
----------
check:   2
----------
check:   3
----------
check:   4
----------
check:   5
----------
check:   6
----------
check:   7
----------
…
…
check:  12
----------
check:  13
----------
CONCLUSION: TRUE

real    0m2.420s
user    0m1.659s
sys     0m0.755s
```

```
verds -b -ck 2 isqrt15.vvm
VERSION:    verds 1.46 - JAN 2015
FILE:       isqrt15.vvm
INFO:       int=i0
PROPERTY:   A F (pc = 5 )
check:   0
----------
check:   1
----------
check:   2
----------
check:   3
----------
check:   4
----------
check:   5
----------
check:   6
----------
check:   7
----------
…
…
check:  13
----------
check:  14
----------
CONCLUSION: TRUE

real    0m2.230s
user    0m1.469s
sys     0m0.744s
```

限界正确性检查(QBF)

```
verds -QBF -ck 1 isqrt15.vvm
VERSION:    verds 1.46 - JAN 2015
FILE:       isqrt15.vvm
INFO:       int=i0
PROPERTY:   A G (! (pc = 5 )| ((x } (r * r ))& (x < ((r + 1 )* (r + 1 )))))
INFO:       applying an internal QBF-solver
check:   0
----------
check:   1
----------
check:   2
----------
check:   3
----------
check:   4
----------
check:   5
----------
check:   6
----------
check:   7
----------
check:   8
----------
check:   9
----------
check:  10
CONCLUSION: TRUE

real    0m4.369s
user    0m3.534s
sys     0m0.819s
```

```
verds -QBF -ck 2 isqrt15.vvm
VERSION:    verds 1.46 - JAN 2015
FILE:       isqrt15.vvm
INFO:       int=i0
PROPERTY:   A F (pc = 5 )
INFO:       applying an internal QBF-solver
check:   0
----------
check:   1
----------
check:   2
----------
check:   3
----------
check:   4
----------
check:   5
----------
check:   6
----------
check:   7
----------
check:   8
----------
check:   9
----------
check:  10
CONCLUSION: TRUE

real    0m2.617s
user    0m1.825s
sys     0m0.770s
```

限界正确性检查(SAT)

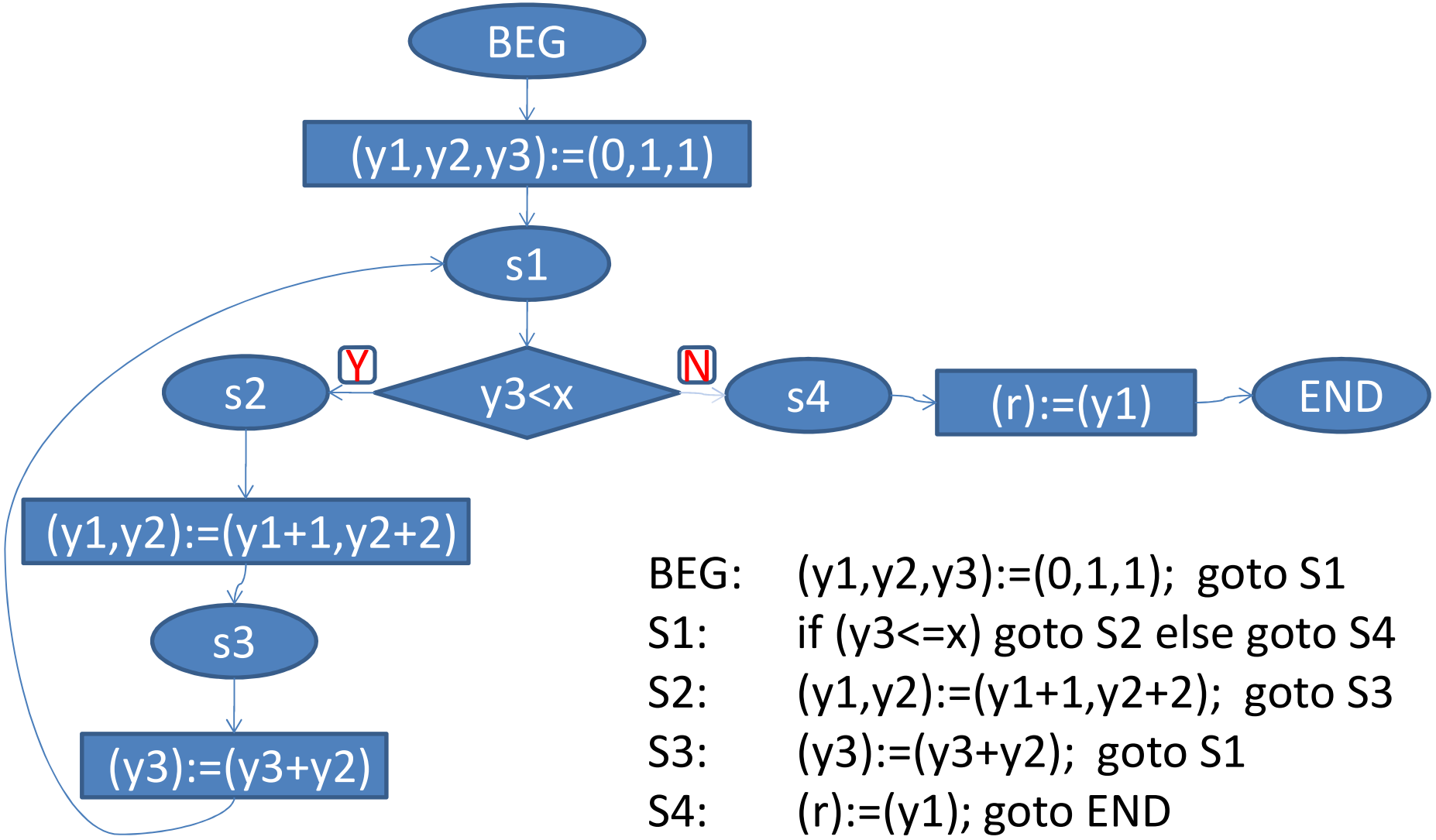
```
verds -SAT-ck 1 isqrt15.vvm
VERSION:    verds 1.46 - JAN 2015
FILE:       isqrt15.vvm
INFO:       int=i0
PROPERTY:   A G (! (pc = 5 )| ((x } (r * r ))& (x < ((r + 1 )* (r + 1 )))))
INFO:       applying an internal SAT-solver
check:   0
----------
check:   1
----------
check:   2
----------
check:   3
----------
check:   4
----------
check:   5
----------
check:   6
----------
check:   7
----------
check:   8
----------
check:   9
----------
check:  10
CONCLUSION: TRUE

real    0m3.935s
user    0m3.026s
sys     0m0.792s
```

```
verds -SAT-ck 2 isqrt15.vvm
VERSION:    verds 1.46 - JAN 2015
FILE:       isqrt15.vvm
INFO:       int=i0
PROPERTY:   A F (pc = 5 )
INFO:       applying an internal SAT-solver
check:   0
----------
check:   1
----------
check:   2
----------
check:   3
----------
check:   4
----------
check:   5
----------
check:   6
----------
check:   7
----------
check:   8
----------
check:   9
----------
check:  10
CONCLUSION: TRUE

real    0m2.531s
user    0m1.598s
sys     0m0.842s
```

例1 -整树平方根：错误设计



BEG: (y1,y2,y3):=(0,1,1); goto S1
S1: if (y3<=x) goto S2 else goto S4
S2: (y1,y2):=(y1+1,y2+2); goto S3
S3: (y3):=(y3+y2); goto S1
S4: (r):=(y1); goto END

例1 -整树平方根

迁移关系		初始状态
pc=BEG →	(y1,y2,y3,pc):=(0,1,1,S1)	pc=BEG∧
pc=S1∧(y3<x) →	(pc):=(S2)	x<=15 ∧
pc=S1∧¬ (y3<x) →	(pc):=(S4)	y1=0 ∧
pc=S2→	(y1,y2,pc):=(y1+1,y2+2,S3)	y2=0 ∧
pc=S3→	(y3,pc):=(y3+y2,S1)	y3=0 ∧
pc=S4→	(r,pc):=(y1,END)	r=0

终止性： AF(pc=END)

部分正确性： AG(pc=END → (x>=r*r ∧ x<(r+1)*(r+1)))

验证

```
verds -ck 1 isqrt15a.vvm
VERSION:    verds 1.46 - JAN 2015
FILE:       isqrt15a.vvm
INFO:       int=i0
PROPERTY:   A G (! (pc = 5 )| ((x } (r * r ))& (x < ((r + 1 )* (r + 1 )))))
check:   0
----------
check:   1
----------
check:   2
----------
check:   3
----------
The property is false, preparing isqrt15a.cex ...
CONCLUSION: FALSE
```

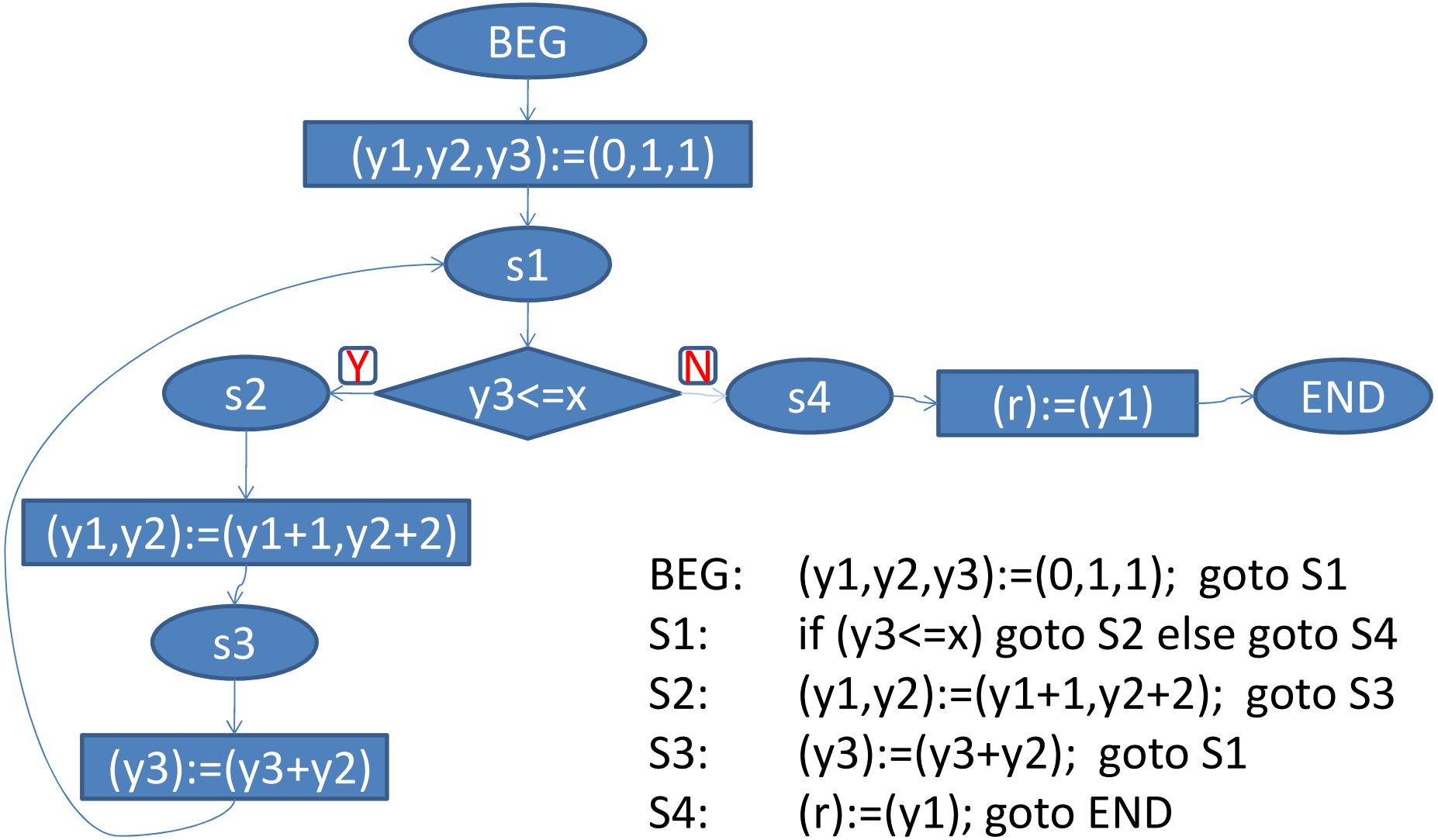
```
verds -ck 2 isqrt15a.vvm
VERSION:    verds 1.46 - JAN 2015
FILE:       isqrt15a.vvm
INFO:       int=i0
PROPERTY:   A F (pc = 5 )
check:   0
----------
check:   1
----------
check:   2
----------
check:   3
----------
check:   4
----------
check:   5
----------
check:   6
----------
check:   7
----------
check:   8
----------
check:   9
----------
check:  10
----------
CONCLUSION: TRUE
```

反例(isqrt15a.cex)

```
--- STATE 0 ---
pc      =0
x       =1
y1      =0
y2      =0
y3      =0
r       =0
--- TRANS 1 ---
--- STATE 1 ---
pc      =1
x       =1
y1      =0
y2      =1
y3      =1
r       =0
--- TRANS 3 ---
--- STATE 2 ---
pc      =4
x       =1
y1      =0
y2      =1
y3      =1
r       =0
--- TRANS 6 ---
```

```
--- STATE 3 ---
pc      =5
x       =1
y1      =0
y2      =1
y3      =1
r       =0
```

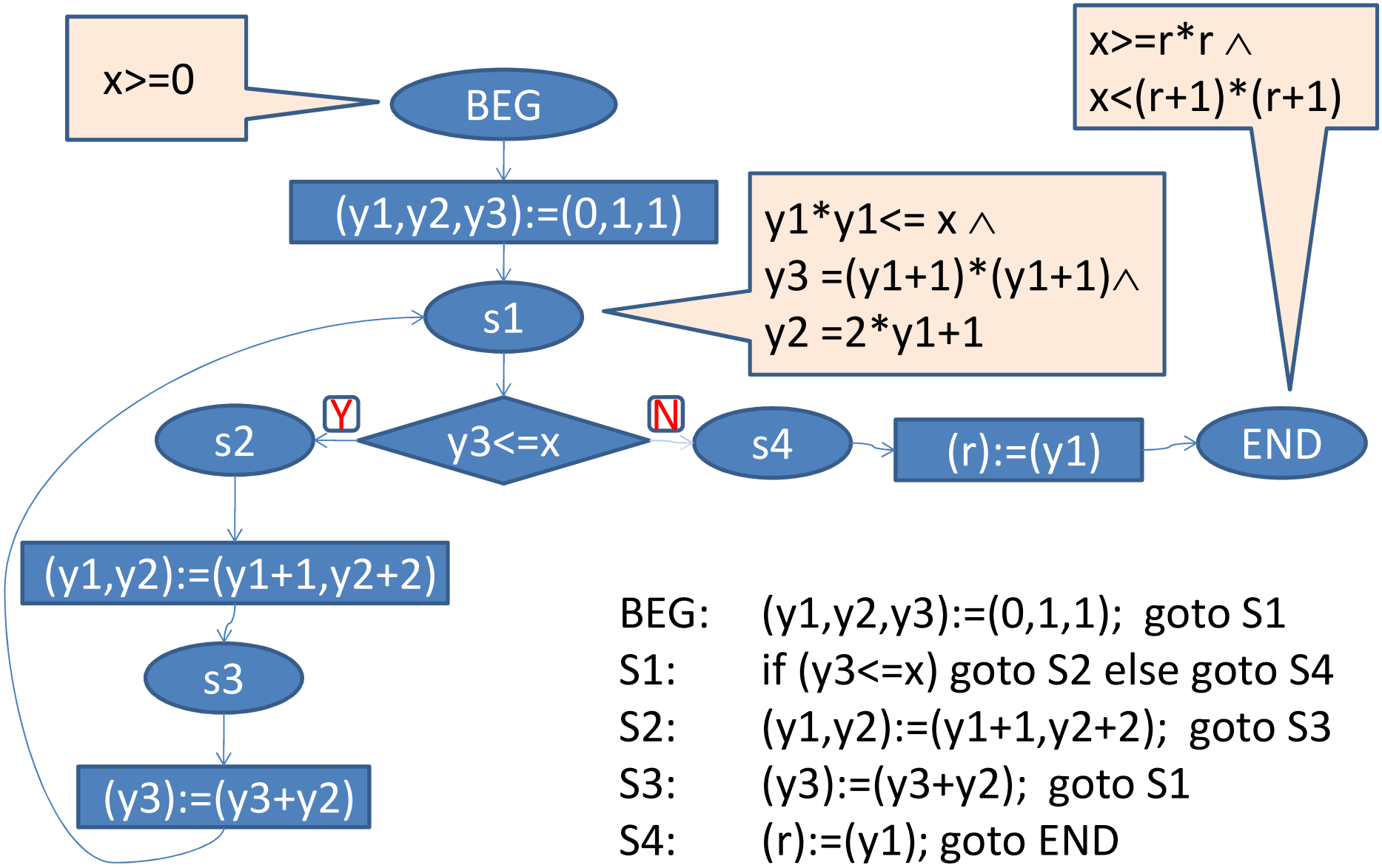
例1 -整树平方根：正确的设计



BEG: (y1,y2,y3):=(0,1,1); goto S1
S1: if (y3<=x) goto S2 else goto S4
S2: (y1,y2):=(y1+1,y2+2); goto S3
S3: (y3):=(y3+y2); goto S1
S4: (r):=(y1); goto END

例1 -整树平方根：推理验证

例1 -整树平方根：部分正确性



BEG: (y1,y2,y3):=(0,1,1); goto S1
S1: if (y3<=x) goto S2 else goto S4
S2: (y1,y2):=(y1+1,y2+2); goto S3
S3: (y3):=(y3+y2); goto S1
S4: (r):=(y1); goto END

验证条件(路径组合方法)

$y1*y1<=x \wedge y3=(y1+1)*(y1+1) \wedge y2=2*y1+1$

$\rightarrow$ $(y3 <=x \rightarrow (y1+1)*(y1+1)<=x \wedge$

$(y2+2)+y3=((y1+1)+1)*((y1+1)+1) \wedge y2+2=2*(y1+1)+1)$

$y1*y1<=x \wedge y3=(y1+1)*(y1+1) \wedge y2=2*y1+1$

$\rightarrow (\neg y3<=x \rightarrow y1*y1<=x \wedge x<(y1+1)*(y1+1))$

$x>=0$

$\rightarrow 0*0<=x \wedge 1=(0+1)*(0+1) \wedge 1=2*0+1$

例1 -整树平方根：部分正确性

```
{ x>=0 }
   y1:=0; y2:=1; y3:=1;
   while (y3<=x) {
        y1:=(y1+1); y2 := (y2+2);
        y3:=(y2+y3);
       { y1*y1<=x∧y3=(y1+1)*(y1+1)∧ y2=2*y1+1}
   }
{ y1*y1<=x ∧ x<=(y1+1)*(y1+1) }
```

Pre-Condition

Invariant

Post-Condition

验证条件(Hoare逻辑规则)

$y3 <= x \wedge y1*y1 <= x \wedge y3 = (y1+1)*(y1+1) \wedge y2 = 2*y1+1$

$\rightarrow (y1+1)*(y1+1) <= x \wedge$

$(y2+2)+y3 = ((y1+1)+1)*((y1+1)+1) \wedge y2+2 = 2*(y1+1)+1$

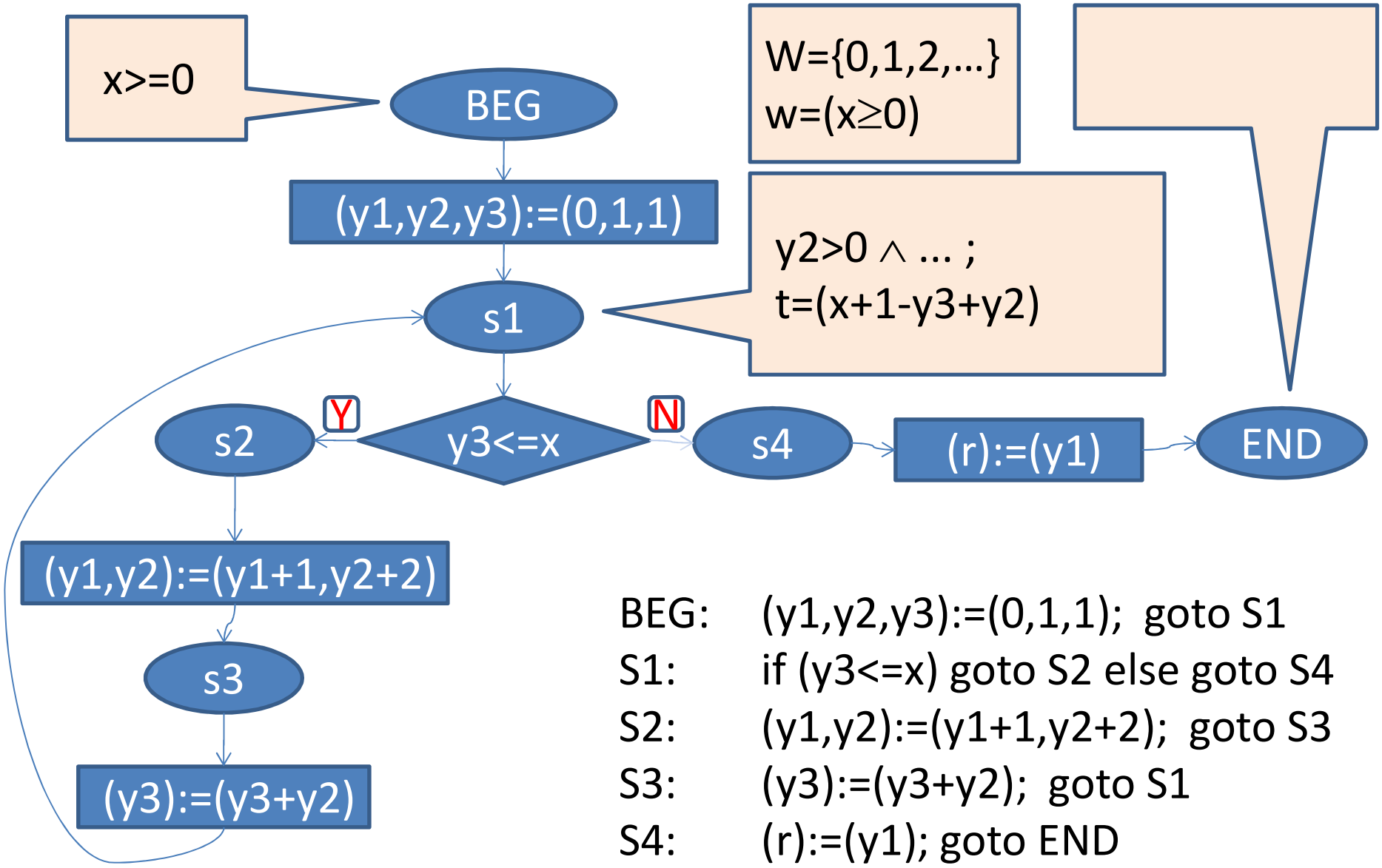
$\neg y3 <= x \wedge y1*y1 <= x \wedge y3 = (y1+1)*(y1+1) \wedge y2 = 2*y1+1$

$\rightarrow y1*y1 <= x \wedge x < (y1+1)*(y1+1)$

$x >= 0$

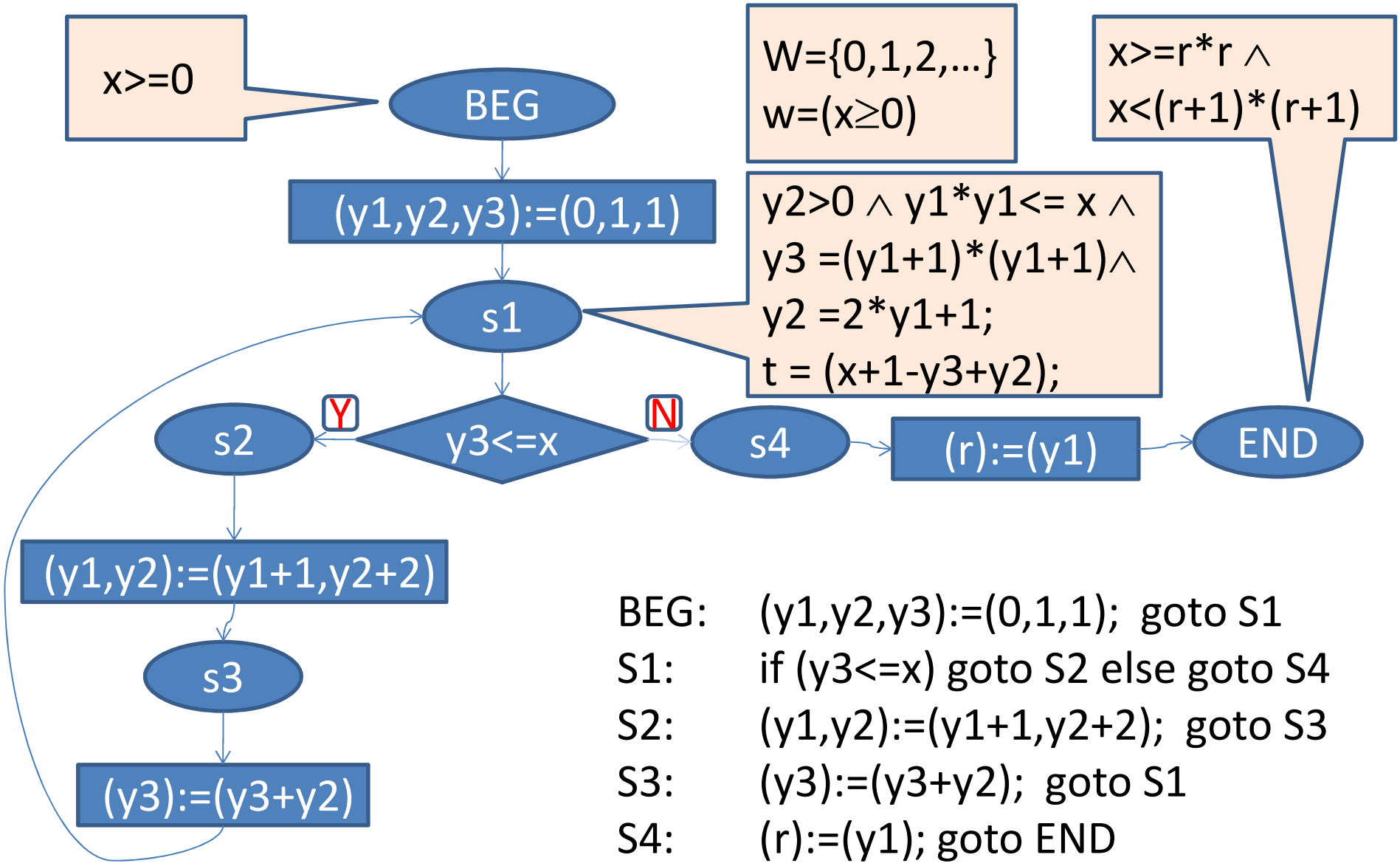
$\rightarrow 0*0 <= x \wedge 1 = (0+1)*(0+1) \wedge 1 = 2*0+1$

例1 -整树平方根：终止性



BEG: (y1,y2,y3):=(0,1,1); goto S1
S1: if (y3<=x) goto S2 else goto S4
S2: (y1,y2):=(y1+1,y2+2); goto S3
S3: (y3):=(y3+y2); goto S1
S4: (r):=(y1); goto END

例1 -整树平方根：完全正确性



BEG: (y1,y2,y3):=(0,1,1); goto S1
S1: if (y3<=x) goto S2 else goto S4
S2: (y1,y2):=(y1+1,y2+2); goto S3
S3: (y3):=(y3+y2); goto S1
S4: (r):=(y1); goto END

例1 -整树平方根：完全正确性

[x>=0]

Pre-Condition

刻画WFS

y1:=0; y2:=1; y3:=1;

while (y3<=x) { { w=(x>=0); term=(x+1-y3+y2) }

y1:=(y1+1); y2 := (y2+2);

y3:=(y2+y3);

Invariant

[y1*y1<=x∧y3=(y1+1)*(y1+1)∧ y2=2*y1+1

∧y2>0]

}

[y1*y1<=x ∧ x<=(y1+1)*(y1+1)]

Post-Condition

验证条件

$y3 <= x \wedge y2>0 \wedge \ldots\ldots \rightarrow (x+1-y3+y2>=0)$
$y3 <= x \wedge \ldots \wedge y2>0 \wedge (x+1-y3+y2)=v$ $\rightarrow (y1+1)*(y1+1)<=x \wedge$ $(y2+2)+y3=((y1+1)+1)*((y1+1)+1) \wedge y2+2=2*(y1+1)+1 \wedge y2>0 \wedge$ $(x+1-(y3+(y2+2)) +(y2+2)) < v$
$\neg y3<=x \wedge y1*y1<=x \wedge y3=(y1+1)*(y1+1) \wedge y2=2*y1+1 \wedge y2>0$ $\rightarrow y1*y1<=x \wedge x<(y1+1)*(y1+1)$
$x>=0$ $\rightarrow 0*0<=x \wedge 1=(0+1)*(0+1) \wedge 1=2*0+1$

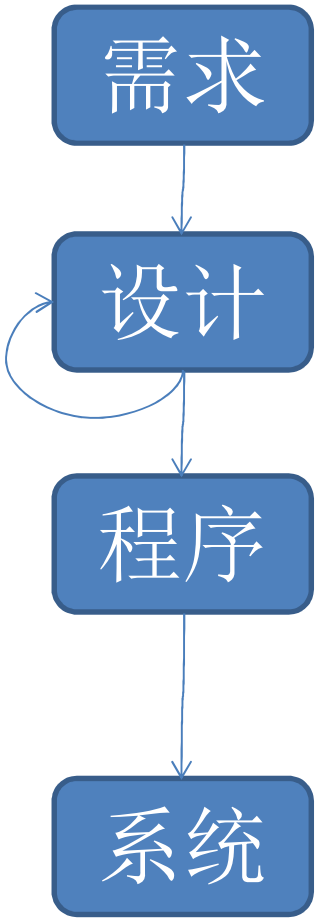
推理验证

构造断言(不变式)与良基域及相关部分

生成验证条件 (最弱宽松前断言与程序推理方法)

证明验证条件 (谓词逻辑推理方法)

程序(软件系统)正确性



需求 ➔ 一致

设计 ➔ 设计
设计 ➔ 需求

程序 ➔ 程序
程序 ➔ 设计
程序 ➔ 需求

系统 ➔ 需求

程序(软件系统)正确性验证方法

问题：

模型是否满足正确性需求 $M \models \varphi$

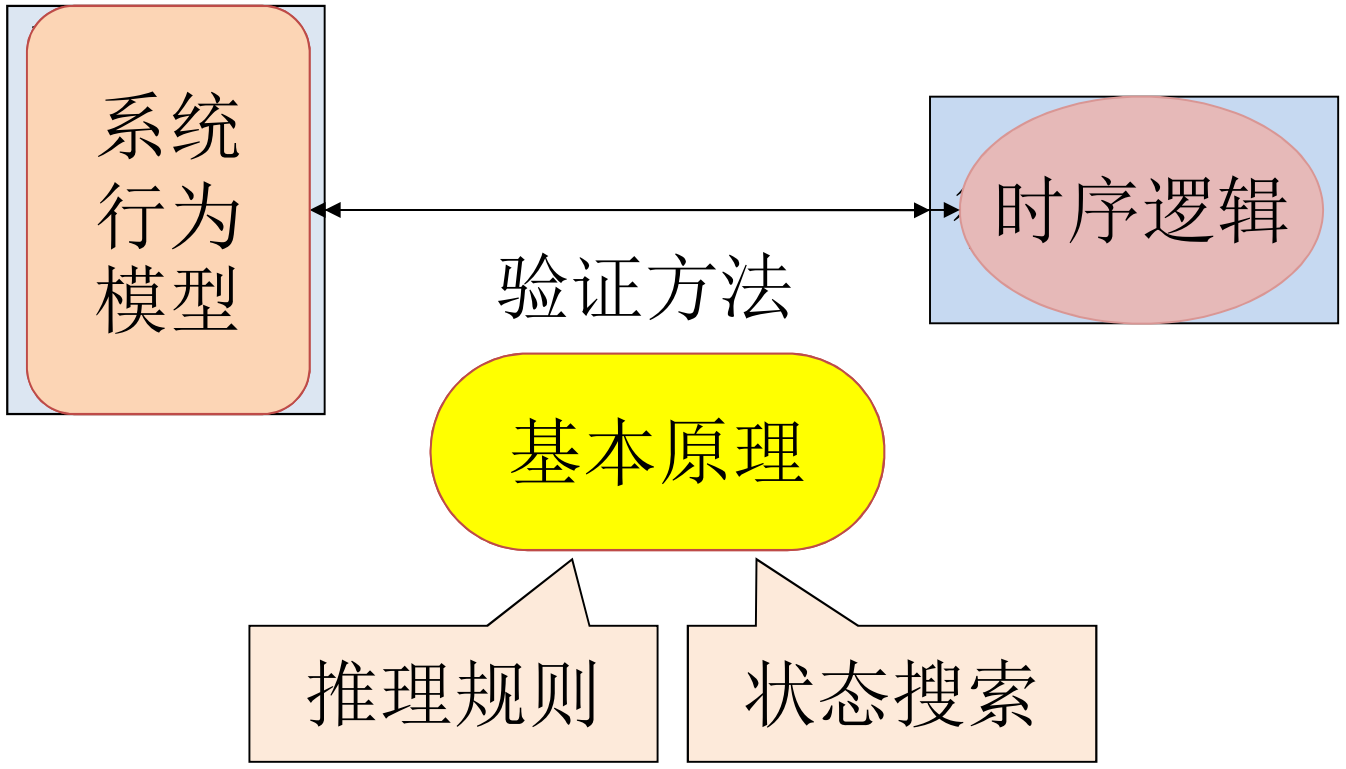
推理验证(可用于无穷状态系统、相对简单的性质)：

模型正确性问题 → 推理问题

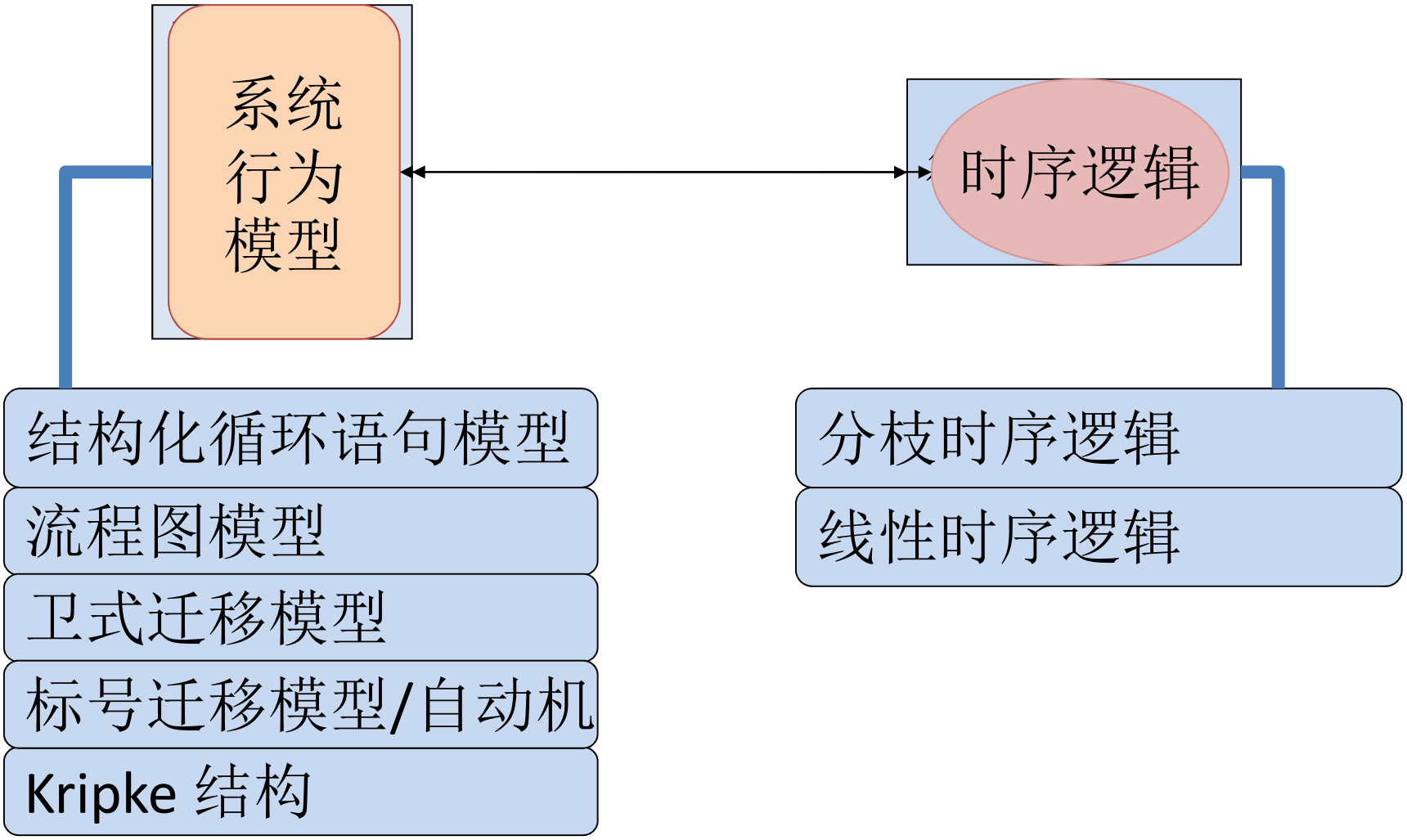
模型检测(适用于有穷状态系统、时序逻辑性质)：

模型正确性问题 → 计算问题

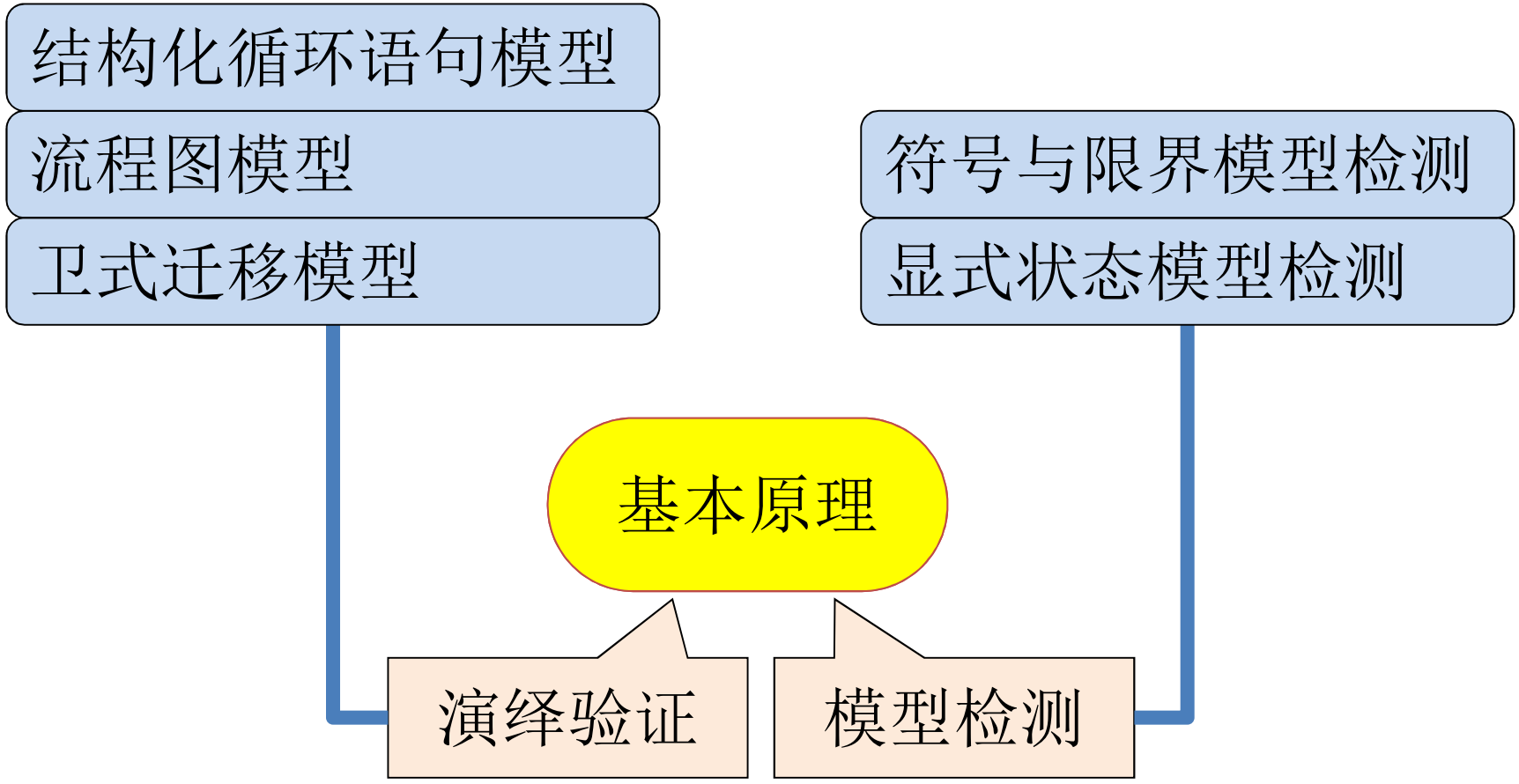
(II)课程主要内容



课程主要内容(1,2)


系统
行为
模型
时序逻辑
结构化循环语句模型
流程图模型
卫式迁移模型
标号迁移模型/自动机
Kripke 结构
分枝时序逻辑
线性时序逻辑

课程主要内容(3,4)



(II.a)程序与系统模型

Kripke结构(II,III)

基本概念

- Kripke结构K=<S,R,I>
- 公平Kripke结构K=<S,R,I,F>
- 标号Kripke结构K=<S,R,I,L>
- 公平标号Kripke结构K=<S,R,I,L,Φ>

- 状态、迁移、路径、计算/运行
- 命题公式与状态集合的关系
- 公平条件的含义、对表达能力的提升作用
- 相关性质(安全、必达)
- 相关验证算法与证明方法

基于谓词逻辑的变量赋值模型(IV)

- 卫式迁移模型M=<T,Θ>
- 流程图模型M
- 结构化循环语句模型M

基本概念、系统描述

- 符号、解释、赋值
- 状态、迁移、路径、计算/运行
- 基于谓词逻辑的模型到Kripke结构的转化
- 模型之间的关系
- 相关性质1(安全、必达)
- 相关性质2(终止性质、部分正确、完全正确)

标号迁移模型(V,VI)

- 标号迁移系统(LTS)
- Buchi自动机A=<Σ,S,Δ,I,F>

基本概念、系统描述

- 字符串、运行、接受条件、语言
- 运算：并、交、(补)
- 确定性与非确定性
- 不同的接受条件(扩展Buchi,Streett,Rabin,Muller)
- 不同种类自动机之间的关系(GBA→BA)
- 相关性质(空性)及验证算法

- 时间自动机 、混成自动机、Petri网

(II.b)时序逻辑

线性时序逻辑(VII,VIII)

- PLTL
- PLTL公式的不动点表示
- PLTL限界语义
- PLTL与自动机
- νTL
- FOLTL

基本概念
系统描述
各类算法

- 语义、可满足性、有效性、等价、最小完全集
- 推理系统
- 基本性质(安全、必达)的表达
- 相关模型检测问题

分枝时序逻辑(IX)

- CTL
- CTL公式的不动点与计算
- CTL限界语义
- μ-演算

基本概念
系统描述
各类算法

- 语义、可满足性、有效性、等价、最小完全集
- 推理系统
- 基本性质(安全、必达)的表达
- 相关模型检测问题

(II.c)程序推理

卫式迁移模型(X)

$M \models_I \phi \Rightarrow (\psi R \varphi)$

$M \models_I \phi \Rightarrow (G\varphi)$

$M \models_I \phi \Rightarrow (\psi U \varphi)$

$M \models_I \phi \Rightarrow (F\varphi)$

基本概念
推理方法
程序分析

- 基于时序逻辑规则的推理方法
- 最弱宽松前断言(迁移集合)：wlp, $[S]\varphi$
- 良基集合： W, w, t
- 部分证明规则前提内容的选择和构造
- 证明规则前提的验证

流程图模型(XI)

- 部分正确性： $\models_{I} \{\phi\} M \{\psi\}$
- 终止性： $\models_{I} [\phi] M [true]$
- 完全正确性： $\models_{I} [\phi] M [\psi]$

基本概念
推理方法
程序分析

- 基于路径组合的推理方法
- 最弱宽松前断言(路径)：wlp, $[\alpha]\varphi$
- 良基集合： W, w, t
- 部分证明规则前提内容的选择和构造
- 验证条件的生成及验证

结构化循环语句模型(XII)

- 部分正确性： $\models_I \{\phi\} M \{\psi\}$
- 终止性： $\models_I [\phi] M [true]$
- 完全正确性： $\models_I [\phi] M [\psi]$

基本概念
推理方法
程序分析

- 基于Hoare逻辑的推理方法
- 最弱宽松前断言(程序片段)：wlp, $[T]\varphi$, 推理规则
- 良基集合： W, w, t
- 部分证明规则前提内容的选择和构造
- 验证条件的生成及验证

(II.d)模型检测

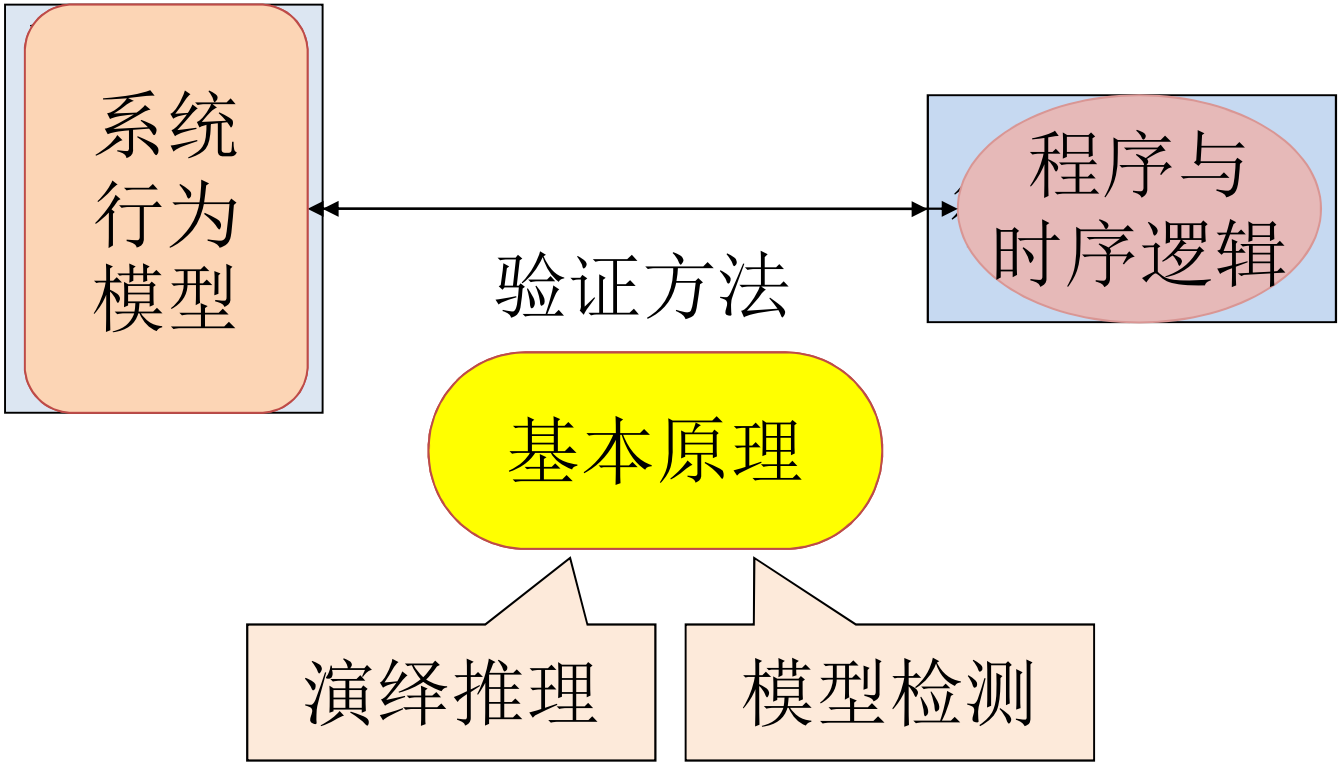
CTL和LTL性质的模型检测方法(XIII)

- 显式状态模型检测
- 符号模型与符号模型检测
- 限界正确性检查(限界模型检测)

基本概念
基本思想
各类算法

- Kripke结构
- 公平Kripke结构

课程内容小结



- 课程内容的综合理解
- 系统行为建模与性质描述
- 有穷状态系统模型的验证
- 基于一阶逻辑的系统模型的验证

概念
原理
应用
讨论

考试时间地点

- 时间：　　2个小时
- 地点：
- 方式：　　课堂开卷
- 参考资料：　课件、参考书等