Temporal Logic in Model Checking

Michael Huth, Mark Ryan. 2004. Logic in Computere Science

Докладчик: Владимир Гладштейн

Dynamic Logic vs Temporal Logic

	Dynamic Logic	Temporal Logic
Для	Hoare Logic...	Model Checking
Основа	$\Gamma \vdash \varphi$	$\mathcal{M} \models \varphi$
Автоматика	$\mp$	$+$.
Что	(не)все что угодно	только конечные модели

План

- Linear-time Temporal Logic (LTL)
- NuSMV
  - request-busy
  - the ferryman
  - critical section
- Computation Tree Logic (CTL)
- Model-checking CTL algorithm

Linear-time Temporal Logic (LTL)

Синтаксис

$\varphi ::=$

- $\top$, $\bot$
- $p \in Prop$
- $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \to \psi$, $\neg\varphi$
- $X\varphi$
- $F\varphi$, $G\varphi$
- $\varphi U \psi$, $\varphi W \psi$

Семантика

Definition (Модель TLT)

$\mathcal{M} = (S, \to, L)$, где

- S - множество состяний
- $\to \subset \mathcal{M}^2$
- $L : S \to \mathcal{P}(Prop)$

называется множеством состяний, если
$\forall s \in S \; \exists s' \in S : s \to s'$

Семантика

Definition (Путь в $\mathcal{M} = (S, \to, L)$)

$s_1 \to s_2 \to \ldots :=$ последовательность $s_1, s_2, \ldots$

где $\forall i, s_i \to s_{i+1}$

Definition (суффикс пути)

пусть $\pi = s_1 \to s_2 \to \ldots$, тогда

$$\pi^i := s_i \to s_{i+1} \to \ldots$$

Семантика

Definition ($\pi \models \varphi$)

пусть $\pi = s_1 \to s_2 \to \dots$

- $\pi \models p \;:\Leftrightarrow\; p \in L(s_1)$
- $\pi \models \varphi \wedge \psi \;:\Leftrightarrow\; \pi \models \varphi$ и $\pi \models \psi$
- $\pi \models \varphi \vee \psi \;:\Leftrightarrow\; \pi \models \varphi$ или $\pi \models \psi$
- $\pi \models \neg\varphi \;:\Leftrightarrow\; \pi \not\models \varphi$

Семантика

Definition (продолжение)

- $\pi \models X\varphi \;:\Leftrightarrow\; \pi^2 \models \varphi$
- $\pi \models G\varphi \;:\Leftrightarrow\; \pi^i \models \varphi$, для всех $i \geqslant 1$
- $\pi \models F\varphi \;:\Leftrightarrow\; \pi^i \models \varphi$, для некоторого $i \geqslant 1$

Семантика

Definition (продолжение)

- $\pi \models \varphi U \psi \ :\Leftrightarrow$
  $\pi^i \models \psi$, для некоторого $i \geqslant 1$ и
  $\pi^j \models \varphi$, для всех $i - 1 \geqslant j \geqslant 1$
- $\pi \models \varphi W \psi \ :\Leftrightarrow$
  $\pi \models \varphi U \psi$ или $\pi \models G \varphi$

Семантика

Definition ($\mathcal{M}, s \models \varphi$)

$$\mathcal{M}, s \models \varphi :\Leftrightarrow$$

для всех путей π, начинающихся в s, $\pi \models \varphi$

Тавтологии

Для всех миров во всех моделях верно:

- $Gp \to p$
- $p \to qUp$
- $p \to Fp$

Примеры

- "Естественный" Until — $\varphi U(\psi \wedge G\neg\varphi)$
- $\neg F\varphi \equiv G\neg\varphi$
- $F\varphi \equiv \top U\varphi$
- $\varphi W\psi \equiv \varphi U\psi \vee G\varphi$
- $\varphi U\psi \equiv \varphi W\psi \wedge F\psi$

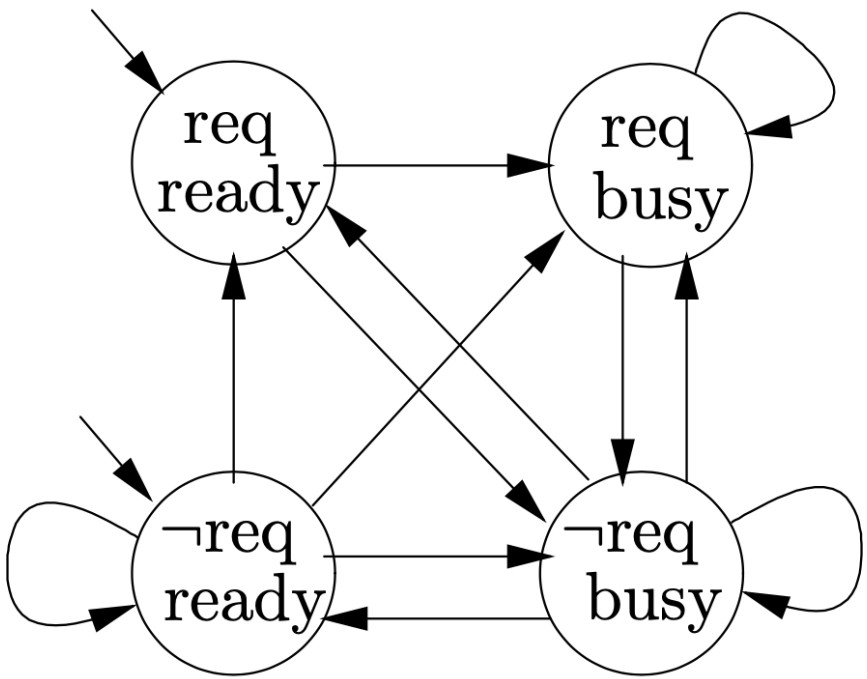
NuSMV

request-busy

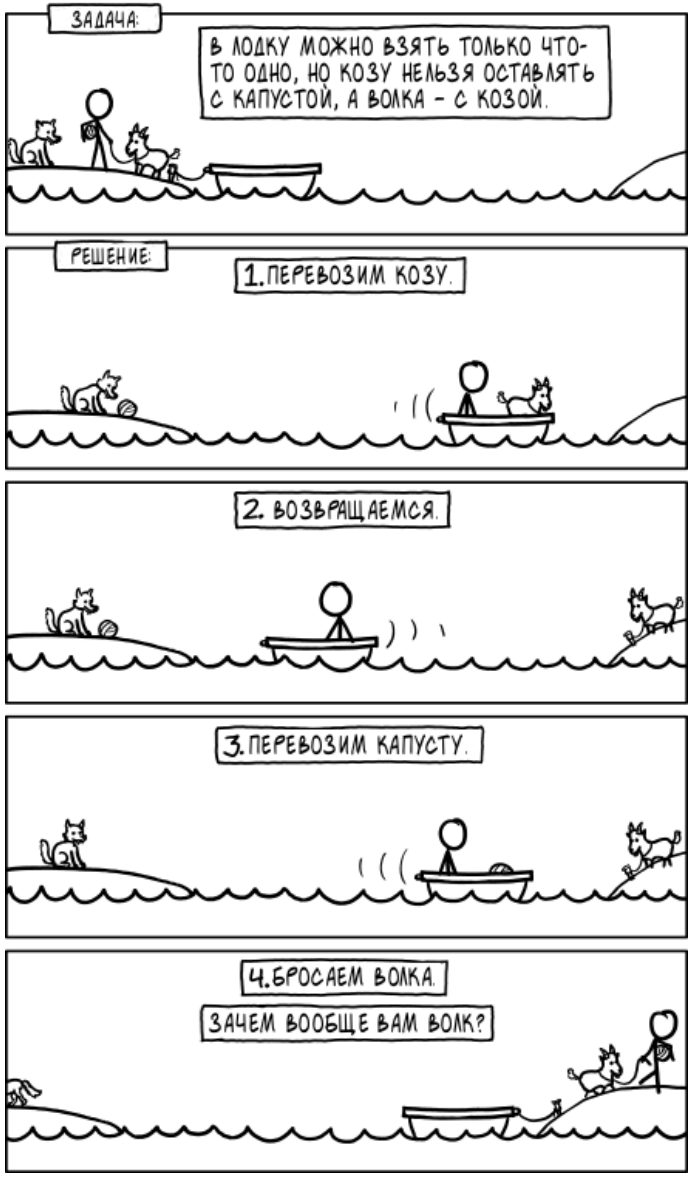
Пусть $Prop := \{requested, busy, ready\}$, хотим $G(request \to Xbusy)$

request-busy

Пусть $Prop := \{requested, busy, ready\}$, хотим $G(request \rightarrow Xbusy)$



the ferryman



critical section (cs)

Пусть $Prop := \{n_1, n_2, t_1, t_2, c_1, c_2\}$, хотим:

critical section (cs)

Пусть $Prop := \{n_1, n_2, t_1, t_2, c_1, c_2\}$, хотим:

- Безопаснось: только один из процессов находтся в cs

critical section (cs)

Пусть $Prop := \{n_1, n_2, t_1, t_2, c_1, c_2\}$, хотим:

- Безопаснось: только один из процессов находтся в cs
- Живучесть: если процесс сделал запрос на cs, то он туда в итоге попадет

critical section (cs)

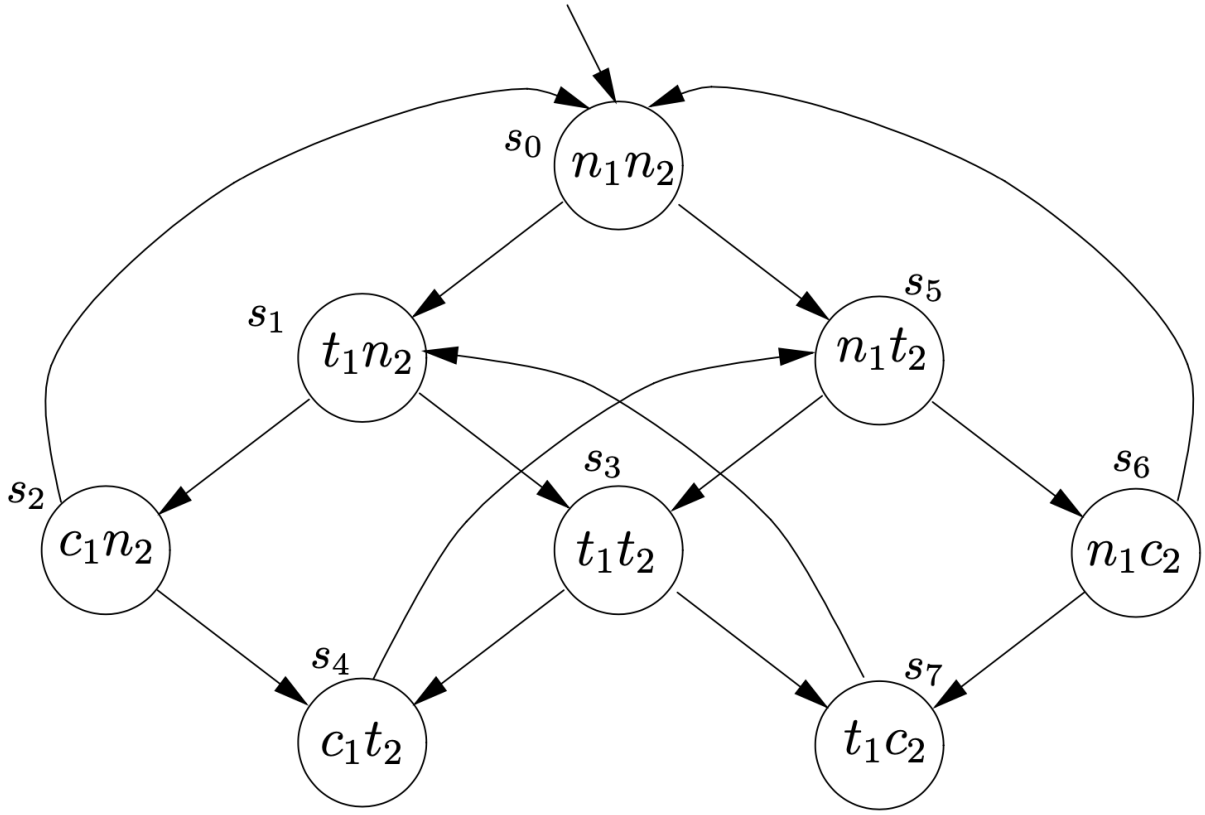
Пусть $Prop := \{n_1, n_2, t_1, t_2, c_1, c_2\}$, хотим:

- Безопаснось: только один из процессов находтся в cs
- Живучесть: если процесс сделал запрос на cs, то он туда в итоге попадет
- Нестрого-последовательный: если процесс вышел из cs, то он может туда снова сразу же зайти

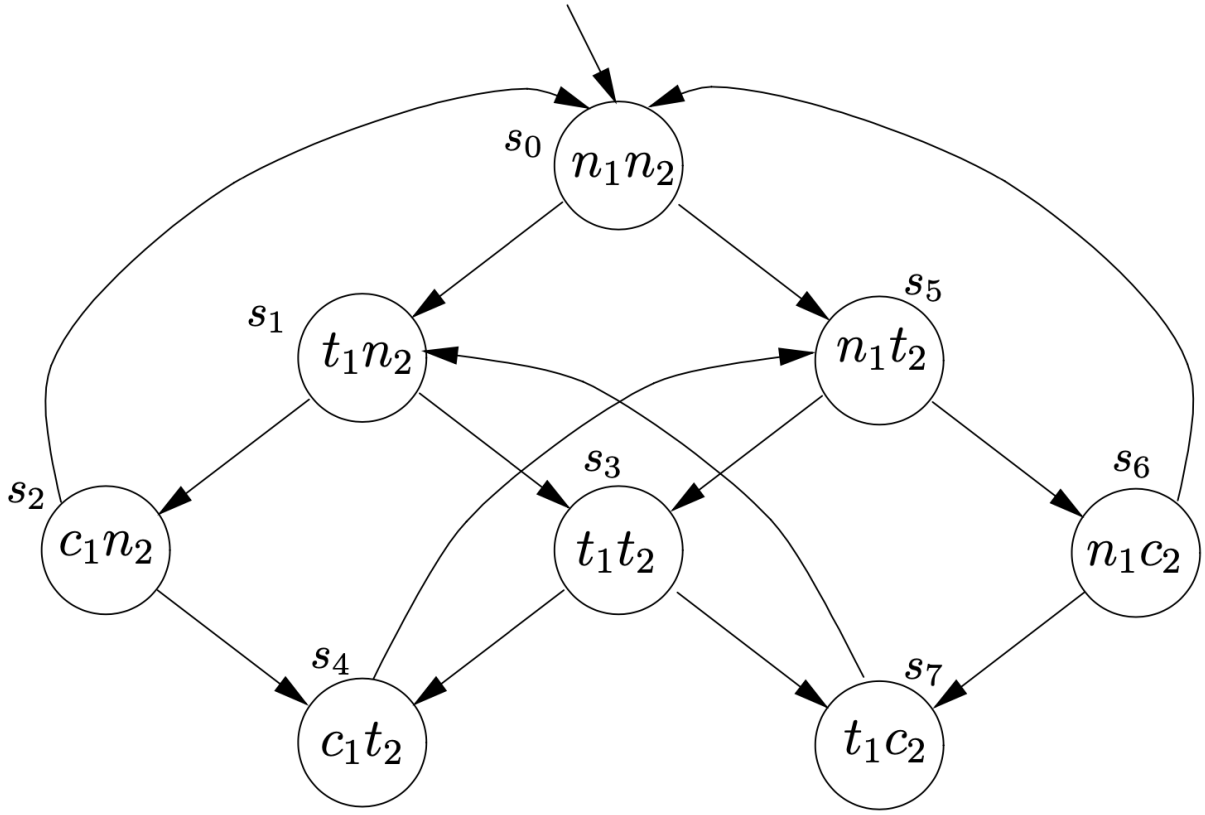
critical section (cs)

Пусть $Prop := \{n_1, n_2, t_1, t_2, c_1, c_2\}$, хотим:

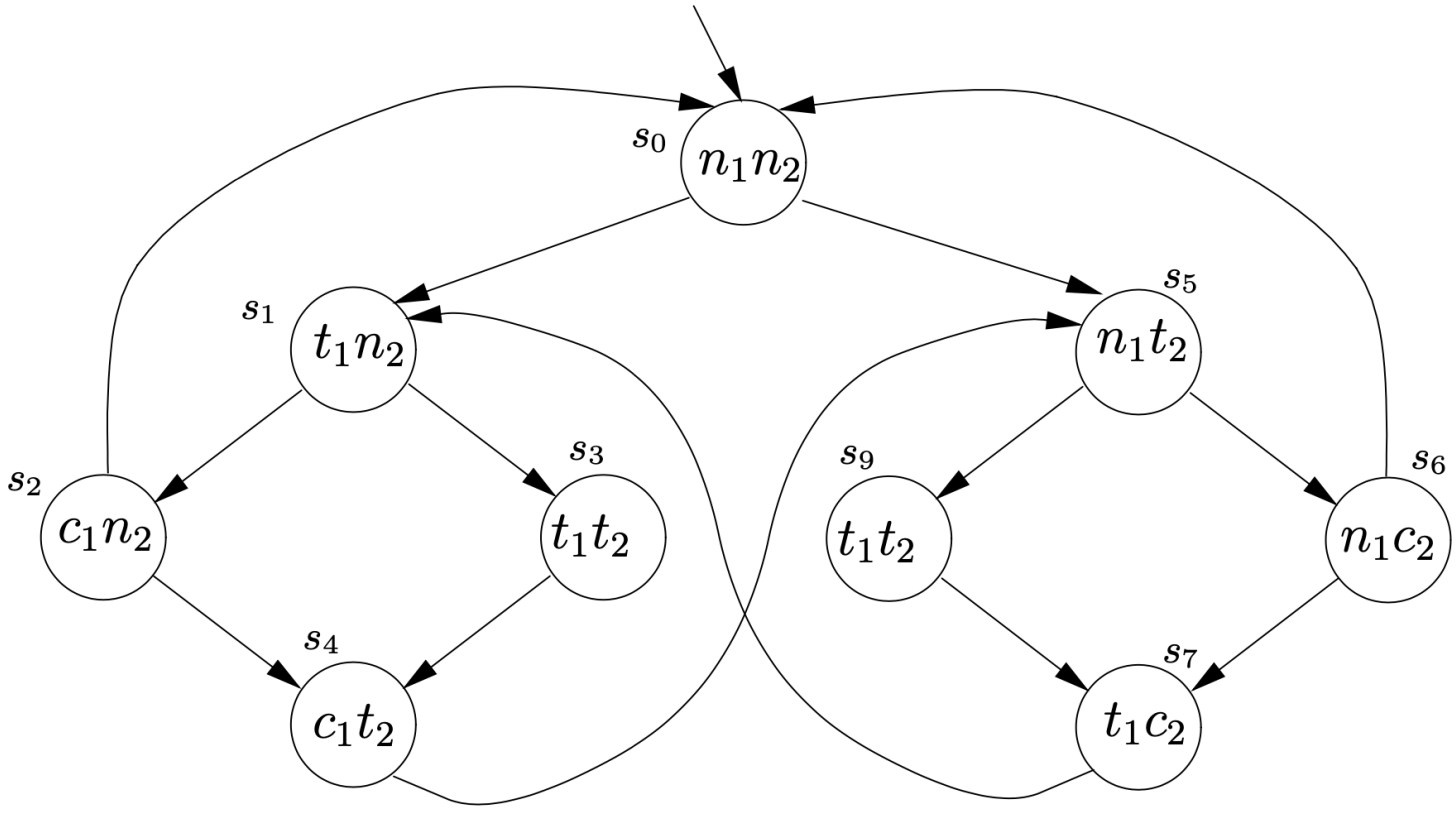
- Безопаснось: только один из процессов находтся в cs
- Живучесть: если процесс сделал запрос на cs, то он туда в итоге попадет
- Нестрого-последовательный: если процесс вышел из cs, то он может туда снова сразу же зайти
- Нет блокировок: процесс всегда может сделать запрос на cs



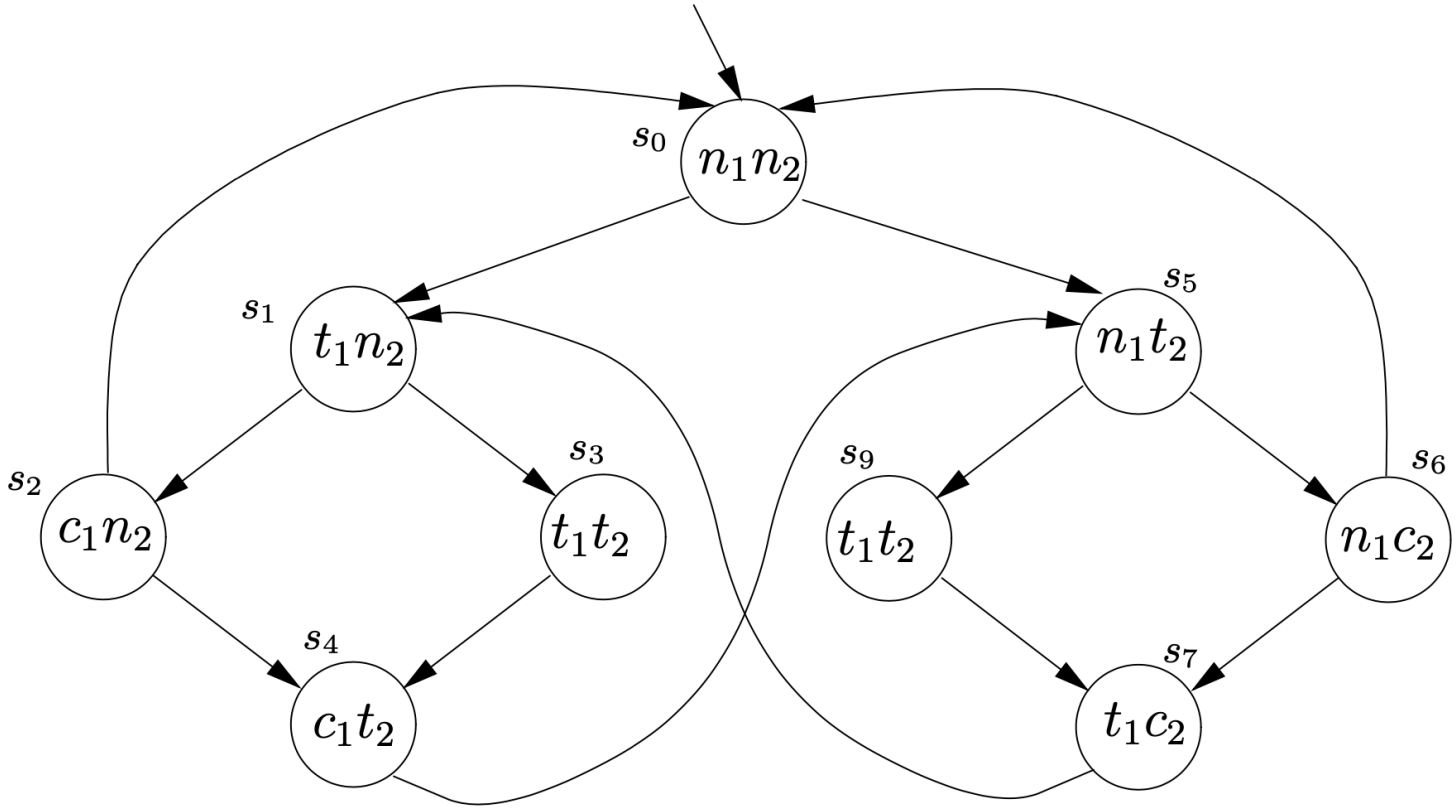
Безопасность: $G\neg(c_1 \wedge c_2)$



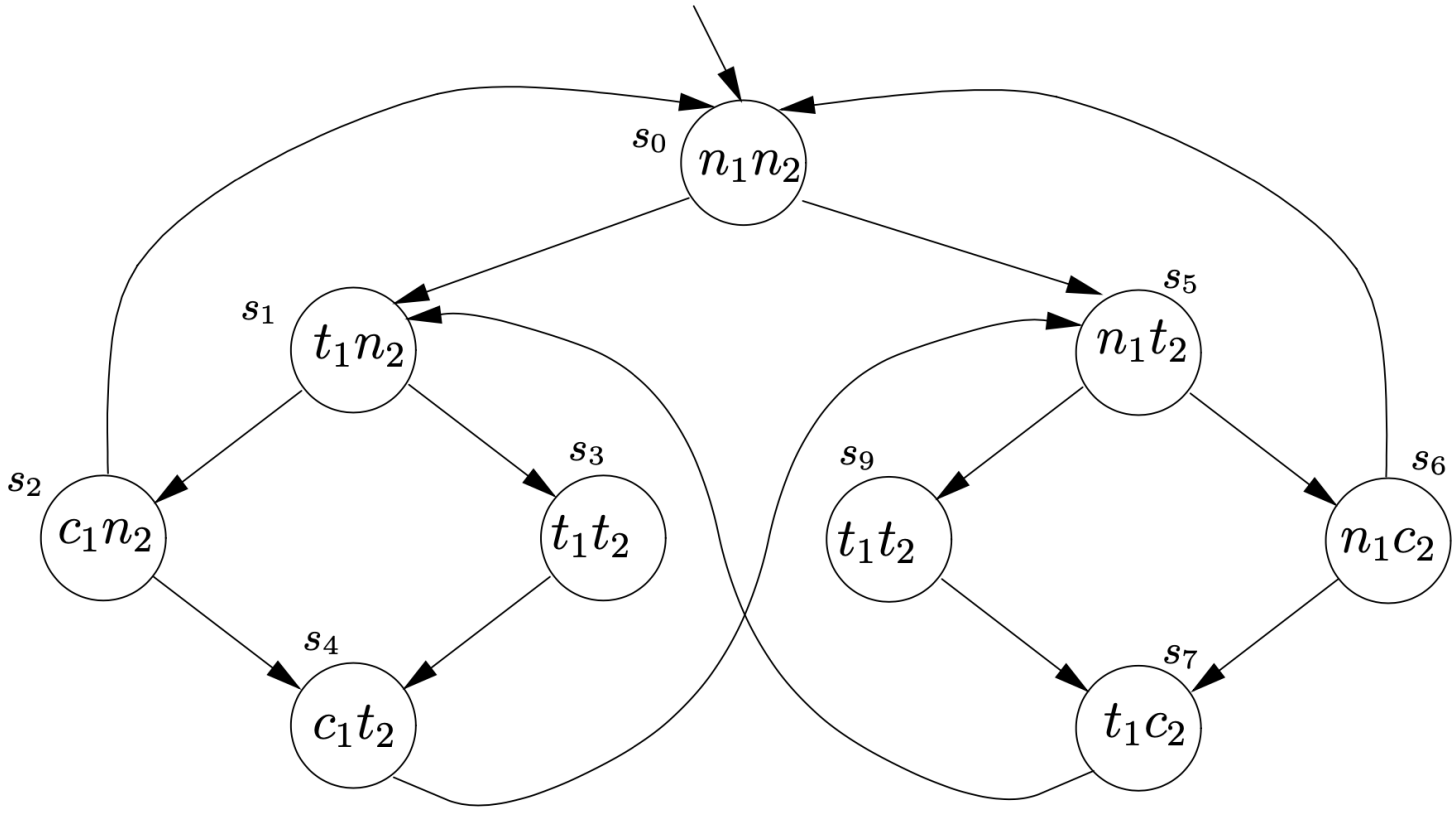
Живучесть: $G(t_i \to Fc_i)$



Живучесть: $G(t_i \to Fc_i)$

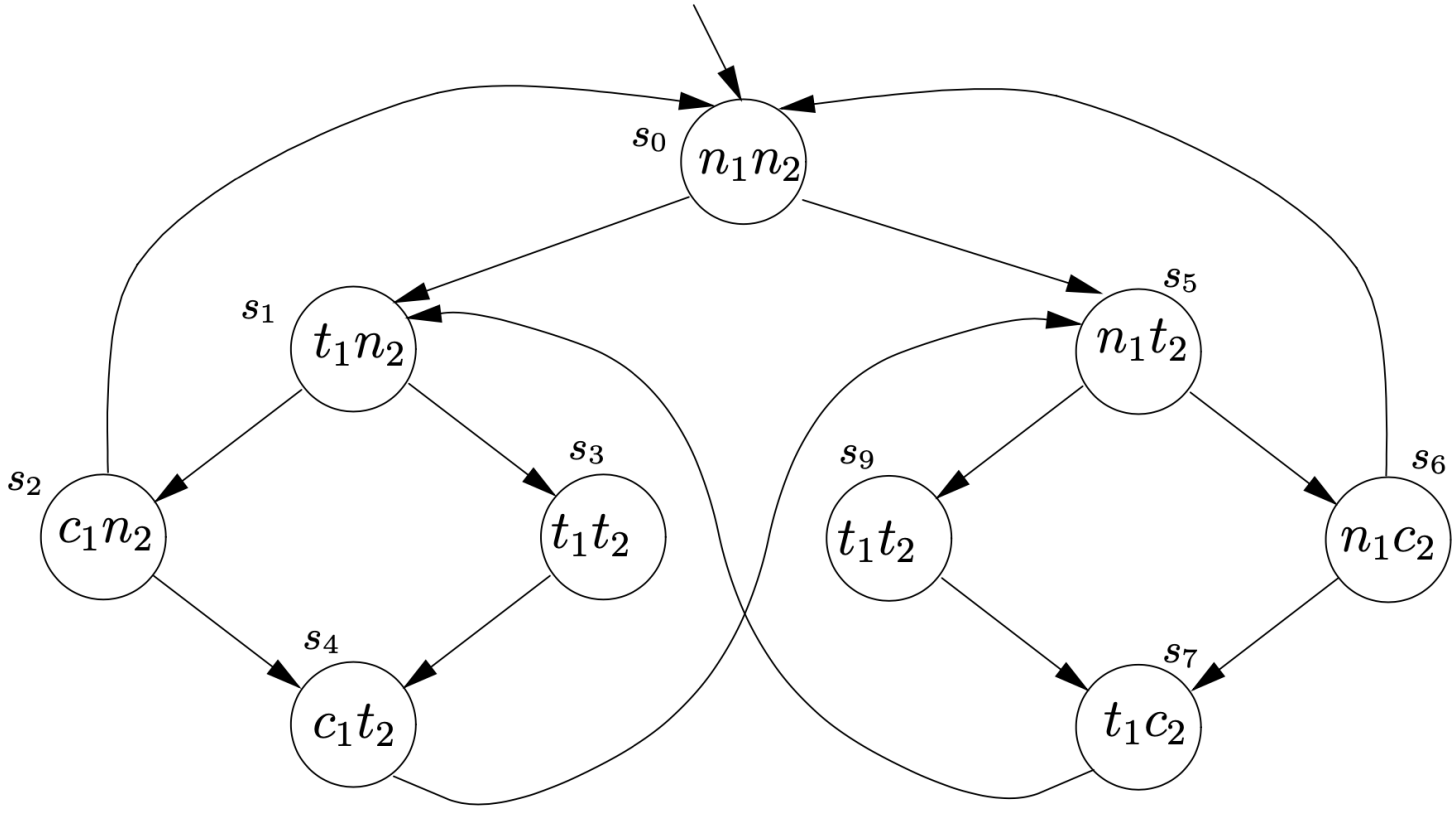


Нестрого-последовательный: нужен $\exists$ по путям, но его нет



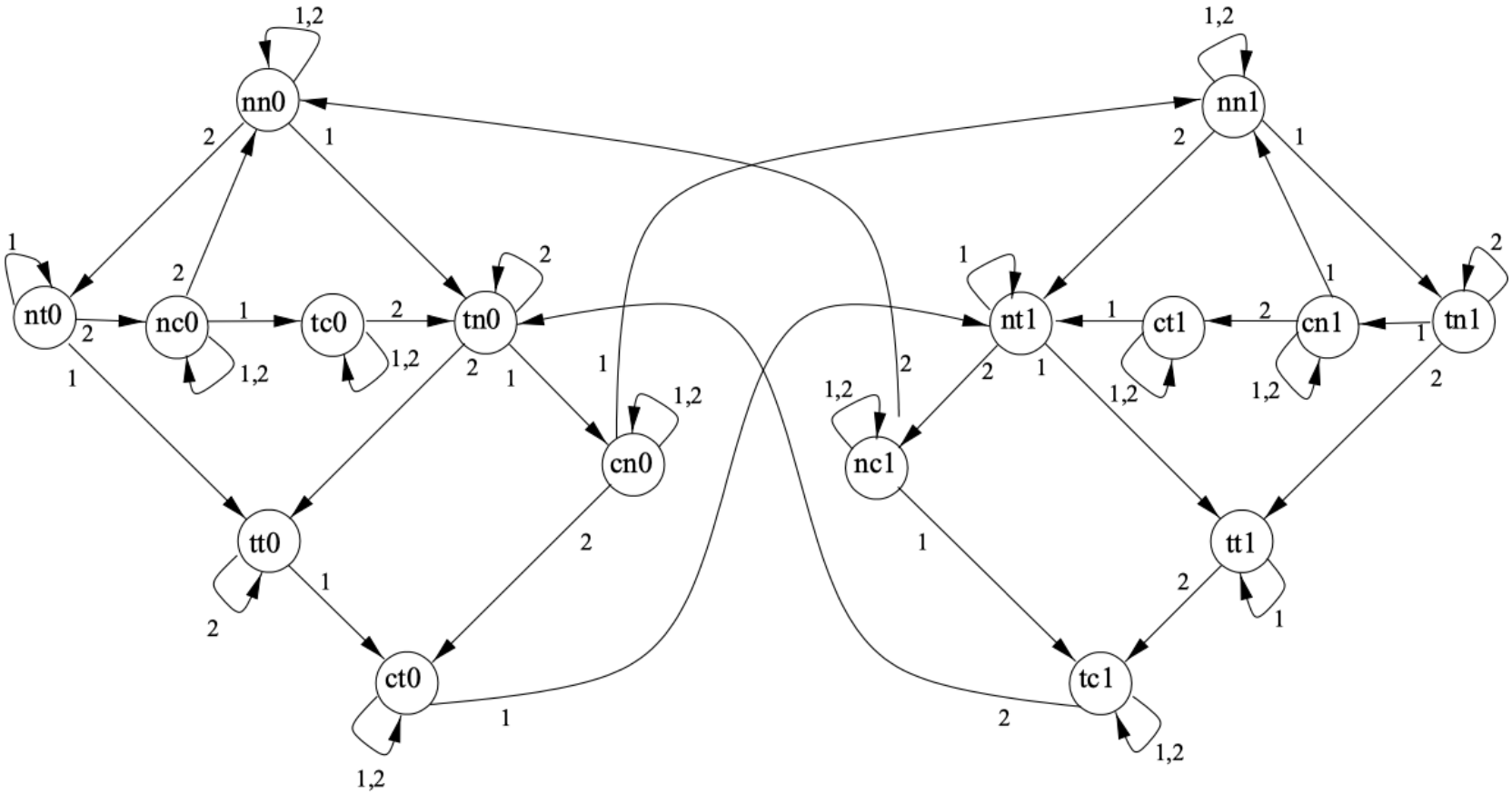
Нестрого-последовательный:

$\mathcal{M}, s_0 \not\models G(c_1 \rightarrow c_1 W(\neg c_1 \wedge \neg c_1 W c_2))$



Нет блокировок: любой путь к n_1, может быть продолжен путем к t_1

Final result



Computation Tree Logic (CTL)

Синтаксис

$\varphi ::=$

- $\top$, $\bot$
- $p \in Prop$
- $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \to \psi$, $\neg\varphi$
- $AX\varphi$, $EX\varphi$
- $AF\varphi$, $EF\varphi$, $AG\varphi$, $EG\varphi$
- $A[\varphi U \psi]$, $E[\varphi U \psi]$

Семантика X

Definition ($\mathcal{M}, s_0 \models \varphi$)

- $$\mathcal{M}, s_0 \models AX\varphi \ :\Leftrightarrow$$
$$\mathcal{M}, s_1 \models \varphi \text{ для всех } s_0 \to s_1$$
- $$\mathcal{M}, s_0 \models EX\varphi \ :\Leftrightarrow$$
$$\mathcal{M}, s_1 \models \varphi \text{ для некоторого } s_0 \to s_1$$

Семантика G

Definition ($\mathcal{M}, s_0 \models \varphi$)

- $$\mathcal{M}, s_0 \models AG\varphi \;:\Leftrightarrow$$
$$\mathcal{M}, s_i \models \varphi \text{ для всех } s_i \text{ во всех}$$
$$s_0 \to s_1 \to s_2 \to \dots$$
- $$\mathcal{M}, s_0 \models EG\varphi \;:\Leftrightarrow$$
$$\mathcal{M}, s_i \models \varphi \text{ для всех } s_i \text{ в нек.}$$
$$s_0 \to s_1 \to s_2 \to \dots$$

Семантика F

Definition ($\mathcal{M}, s_0 \models \varphi$)

- $$\mathcal{M}, s_0 \models AF\varphi \;:\Leftrightarrow$$
  $$\mathcal{M}, s_i \models \varphi \text{ для нек. } s_i \text{ во всех}$$
  $$s_0 \to s_1 \to s_2 \to \dots$$
- $$\mathcal{M}, s_0 \models EF\varphi \;:\Leftrightarrow$$
  $$\mathcal{M}, s_i \models \varphi \text{ для нек. } s_i \text{ в нек.}$$
  $$s_0 \to s_1 \to s_2 \to \dots$$

Семантика U

Definition ($\mathcal{M}, s_0 \models \varphi$)

- $$\mathcal{M}, s_0 \models A[\varphi U \psi] \ :\Leftrightarrow$$

  $\pi \models \varphi U \psi$, для всех $\pi = s_0 \to s_1 \to \ldots$

- $$\mathcal{M}, s_0 \models E[\varphi U \psi] \ :\Leftrightarrow$$

  $\pi \models \varphi U \psi$, для нек. $\pi = s_0 \to s_1 \to \ldots$

critical section в CTL

- Нет блокировок: $AG(n_i \rightarrow EFt_i)$

critical section в CTL

- Нет блокировок: $AG(n_i \rightarrow EFt_i)$
- Нестрого-последовательный:
  $EF\,(c_1 \wedge [c_1\ U\ (\neg c_1 \wedge E\,[\neg c_2\ U\ c_1])])$

critical section в CTL

- Нет блокировок: $AG(n_i \rightarrow EFt_i)$
- Нестрого-последовательный:
  $EF\,(c_1 \wedge [c_1\ U\ (\neg c_1 \wedge E\,[\neg c_2\ U\ c_1])])$
- Безопасность: $AG\neg(c_1 \wedge c_2)$

critical section в CTL

- Нет блокировок: $AG(n_i \to EFt_i)$
- Нестрого-последовательный:
  $EF\,(c_1 \wedge [c_1\ U\ (\neg c_1 \wedge E\,[\neg c_2\ U\ c_1])])$
- Безопасность: $AG\neg(c_1 \wedge c_2)$
- Живучесть: $AG(t_i \to AFc_i)$

critical section в CTL

- Нет блокировок: $AG(n_i \to EFt_i)$
- Нестрого-последовательный:
  $EF\,(c_1 \wedge [c_1\ U\ (\neg c_1 \wedge E\,[\neg c_2\ U\ c_1])])$
- Безопасность: $AG\neg(c_1 \wedge c_2)$
- Живучесть: $AG(t_i \to AFc_i)$
- Если бы мы захотели поросить в LTL
  $GFt_i \to GFc_i$

critical section в CTL

- Нет блокировок: $AG(n_i \to EFt_i)$
- Нестрого-последовательный:
  $EF\,(c_1 \wedge [c_1\ U\ (\neg c_1 \wedge E\,[\neg c_2\ U\ c_1])])$
- Безопасность: $AG\neg(c_1 \wedge c_2)$
- Живучесть: $AG(t_i \to AFc_i)$
- Если бы мы захотели поросить в LTL
  $GFt_i \to GFc_i$, то в CTL мы такое выразить бы не смогли: $AG\ AFt_i \to AG\ AFc_i$ не походит

Model-checking CTL algorithm

The labelling algorithm

- Хотим понять: $\mathcal{M}, s_0 \models \varphi$

The labelling algorithm

- Хотим понять: $\mathcal{M}, s_0 \models \varphi$
- INPUT: $\mathcal{M}, \varphi$

The labelling algorithm

- Хотим понять: $\mathcal{M}, s_0 \models \varphi$
- INPUT: $\mathcal{M}, \varphi$
- OUTPUT: $\{s \in \mathcal{M} \mid \mathcal{M}, s \models \varphi\}$

Как?

- `Step 1:` TRANSLATE(φ)
  (в теминах $AF, EU, EX, \wedge, \neg, \bot$)
- `Step 2:` отметь все состояния $\mathcal{M}$,
  подформулами φ, которое в них верны
  (начиная с самых маленьких подформул)

Step 1

У нас есть: AF, EU, EX

- $AX\psi \equiv \neg EX\neg\psi$
- $EF\psi \equiv E[\top U\psi]$
- $EG\psi \equiv \neg AF\neg\psi$
- $AG\psi \equiv \neg EF\neg\psi$
- $A[\varphi U\psi] \equiv \neg(E[\neg\psi U(\neg\varphi \wedge \neg\psi)] \vee EG\neg\psi)$

Step 2

Пусть ψ, подформула φ, имеет вид

- $\bot$: никого помечать не надо

Step 2

Пусть ψ, подформула φ, имеет вид

- $\bot$: никого помечать не надо
- p: пометь s, если $p \in L(s)$

Step 2

Пусть ψ, подформула φ, имеет вид

- $\bot$: никого помечать не надо
- p: пометь s, если $p \in L(s)$
- $\psi_1 \wedge \psi_2$: пометь s, если оно помечено ψ_1 и ψ_2

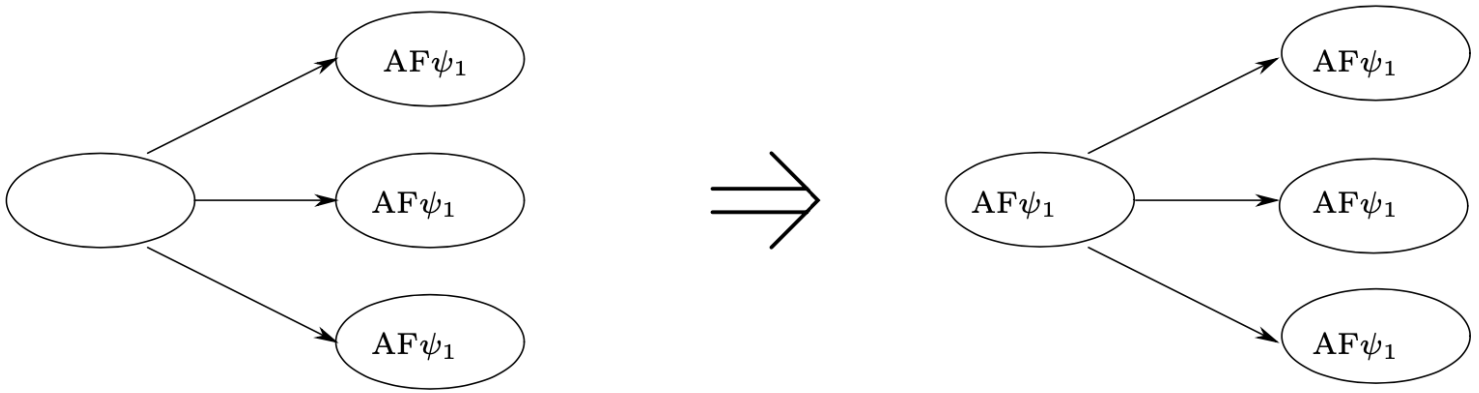
Step 2

Пусть ψ, подформула φ, имеет вид

- $\bot$: никого помечать не надо
- p: пометь s, если $p \in L(s)$
- $\psi_1 \wedge \psi_2$: пометь s, если оно помечено ψ_1 и ψ_2
- $\neg\psi$: пометь s, если оно непомечено ψ

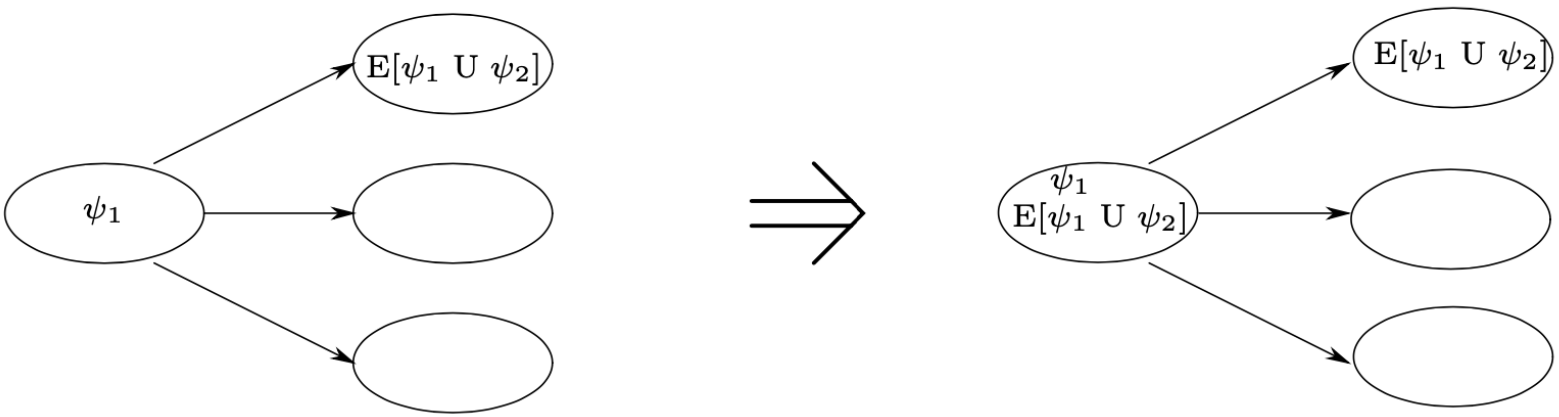
Step 2

- $AF\psi_1$
  - если s помечено ψ_1, пометь его $AF\psi_1$
  - Повторяй пока будут изменения: пометь s $AF\psi_1$, если все его потомки помечены



Step 2

- $E[\psi_1 \; U \; \psi_2]$
  - если s помечено ψ_2, пометь его $E[\psi_1 \; U \; \psi_2]$
  - Повторяй пока будут изменения: пометь s $E[\psi_1 \; U \; \psi_2]$, если оно помечено ψ_1, и хоть один его потомок помечен $E[\psi_1 \; U \; \psi_2]$



Step 2

- $EX\psi_1$: пометь s $EX\psi_1$, если хоть его потомк помечен ψ_1

Step 2

- $EX\psi_1$: пометь s $EX\psi_1$, если хоть его потомк помечен ψ_1

Далее этот алгоритм можно улучшать