Введение в моделирование и верификацию аппаратных и программных систем

Лекция 3: Логика линейного времени

Борис Юрьевич Конев

konev@liverpool.ac.uk

Liverpool University

Октябрь-Ноябрь 2007

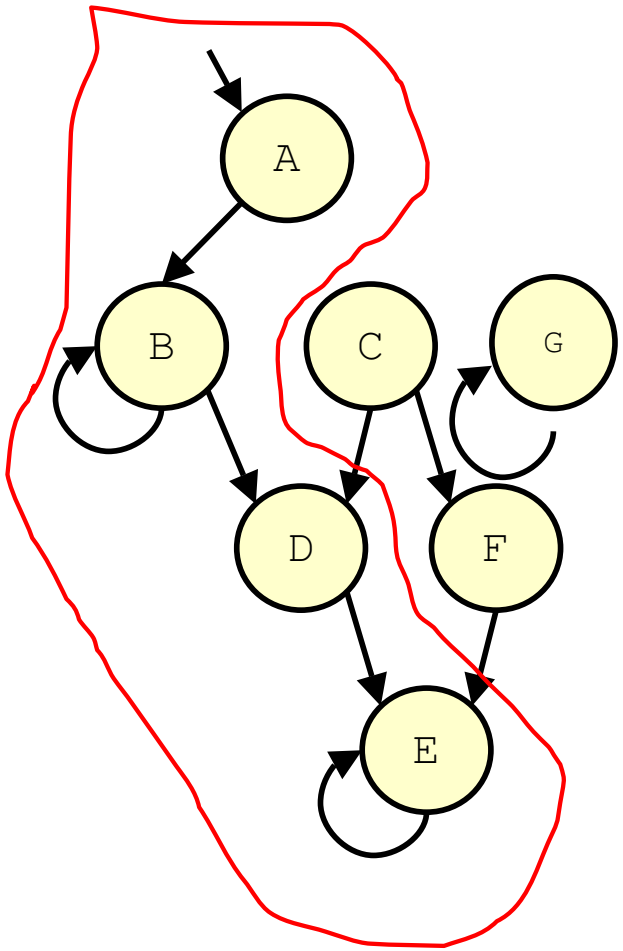
Что было вчера

- Историческая необходимость верификации
- Реактивные системы
  - Взаимодействуют с окружением не завершая своей работы
  - Например, системы управления полетами, драйверы, протоколы,. . .
- Системы переходов как модель реактивных систем
- Параллельная (синхронизированная) композиция
- Анализ достижимости для верификации простых свойств
  - Система достигнет желаемого состояния
  - Система не попадет в нежелательное состояние

Система переходов & анализ достижимости

(Q, E, T, q_0, L)

- Q — конечное множество состояний
- E — конечное множество меток переходов
- $T \subseteq Q \times E \times Q$ — множество переходов
- q_0 — начальное состояние
- $L : Q \to Prop$ — пометка пропозициональными переменными



Основные вопросы на сегодня

- Как описать более сложные свойства
  - “Каждый процесс, который хочет войти в критическую секция, рано или поздно войдет в неё”
  - “каждое посланное сообщение будет получено”
  - “устройство готово к работе бесконечно часто”
- Как проверить, что модель обладает этими свойствами?

Логический формализм

- Синтаксис
  - Что такое синтаксически допустимые объекты (правильно сформированные формулы)?
- Семантика
  - Смысл формул
- Дедуктивная система
  - Правила получения “истинных” формул

Пример: пропозициональная логика. Синтаксис

- (Потенциально бесконечное) множество пропозициональных переменных *Prop*
- Связки: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\equiv$
- Формулы (множество *WFF*):
  - Пропозициональные переменные p, q, r,...
  - Если φ и ψ – формулы, то
    - $\neg\varphi$
    - $\varphi \wedge \psi$
    - $\varphi \vee \psi$
    - $\varphi \rightarrow \psi$
    - $\varphi \equiv \psi$

    $\Big\}$ формулы
  - Никакой другой объект формулой не является

Пример: пропозициональная логика. Семантика

- Интерпретация пропозициональных переменных

$$\mathcal{I} : Prop \to \{\mathbf{T}, \mathbf{F}\}$$

- Расширяется до интерпретация формул

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \to Q$	$P \equiv Q$
T	**T**	**F**	**T**	**T**	**T**	**T**
T	**F**	**F**	**F**	**T**	**F**	**F**
F	**T**	**T**	**F**	**T**	**T**	**F**
F	**F**	**T**	**F**	**F**	**T**	**T**

Пример: пропозициональная логика. Дедуктивная теория

- Гильбертовская система
- Метод резолюции

Временные пропозициональные логики

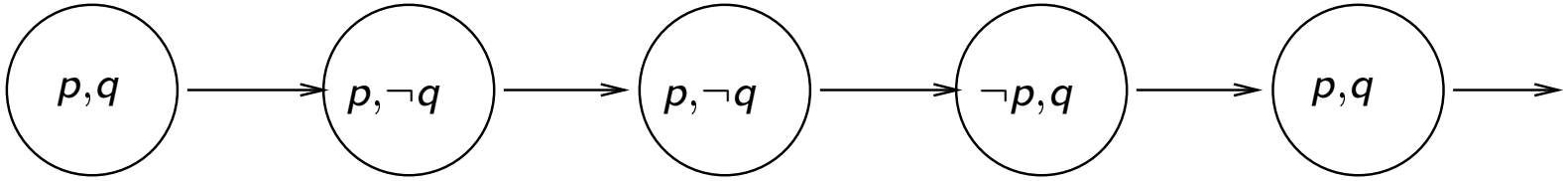
- В классической пропозициональной логике интерпретации статические. Каждая формула может быть либо истинной, либо ложно.
- Интерпретации временных логик представляют собой множество “миров”, связанных неким отношением. Пропозициональная переменная может быть истинной в одном мире и ложной в другом.
- Пропозициональные временные логики расширяют классическую пропозициональную логику: те же пропозициональные переменные, те же классические связки $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\equiv$.
- Дополнительные связки, чтобы судить о разных моментах времени

Но что есть время?

Логика линейного времени (LTL)

Linear Time Logic

- Существует много разных временных логик. Мы начнем с пропозициональной логики линейного времени с конечным прошлым и бесконечным будущим.
  - Время изоморфно $\mathbb{N}$
- Модели представляют собой последовательности состояний, индексированные натуральными числами; пропозициональные переменные могут быть истинными или ложными в каждом состоянии.



Операторы временной логики

- **X** ‘в следующий момент’ (ne**X**t);
- **F** ’в некоторый момент в будущем’ (in the **F**uture);
- **G** ’всегда в будущем’; (**G**lobally in the future);

а также

- **U** ’**U**ntil’.
- **W** ’**W**eak until’

(Также существуют операторы для прошлого, но мы их сейчас не рассматриваем)

Синтаксис LTL

- Пропозициональных переменных *Prop*
- Пропозициональных связок **true**, **false**, $\neg$, $\vee$, $\wedge$, $\rightarrow$ и $\equiv$.
- Временных операторов, **X**, **F**, **G**, **U** и **W**.
- Множество формул (*WFF*)
  - Пропозициональные переменные
  - **true** и **false**
  - Если φ и ψ – формулы, то
    - $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, $\varphi \equiv \psi$
    - $\mathbf{X}\varphi$
    - $\mathbf{F}\varphi$
    - $\mathbf{G}\varphi$
    - $\varphi\mathbf{U}\psi$
    - $\varphi\mathbf{W}\psi$

$\left.\right\}$ формулы

Семантика

- LTL интерпретируется над дискретными линейными порядками, изоморфными множеству натуральных чисел $\mathbb{N}$.
- Интерпретация LTL, σ, это последовательность состояний

$$\sigma = s_0, s_1, s_2, s_3, \ldots$$

  где каждое состояние s_i представляет собой множество пропозициональных переменных, истинных в i-й момент времени.
- Формулы интерпретируются в момент времени $i \in \mathbb{N}$.

$$(\sigma, i) \models A$$

  означает истинность формулы A в модели σ в момент времени $i \in \mathbb{N}$.

Пропозициональные переменные и связки

Для пропозициональных переменных, просто проверяем, что они верны в этот момент времени

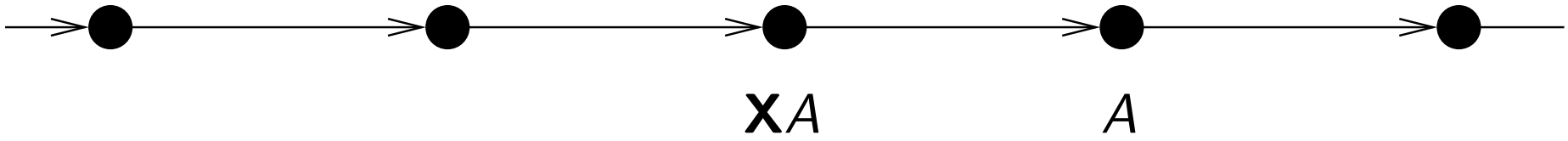
$$(\sigma, i) \models p \iff p \in s_i \qquad [\text{где } p \in Prop]$$

Для пропозициональных связок, семантика такая же, как в пропозициональной логике.

$$(\sigma, i) \models \textbf{true}$$
$$(\sigma, i) \not\models \textbf{false}$$
$$(\sigma, i) \models A \wedge B \iff (\sigma, i) \models A \text{ и } (\sigma, i) \models B$$
$$(\sigma, i) \models A \vee B \iff (\sigma, i) \models A \text{ или } (\sigma, i) \models B$$
$$(\sigma, i) \models A \to B \iff (\sigma, i) \models \neg A \text{ или } (\sigma, i) \models B$$
$$(\sigma, i) \models A \equiv B \iff (\sigma, i) \models A \to B \text{ и } (\sigma, i) \models B \to A$$
$$(\sigma, i) \models \neg A \iff (\sigma, i) \not\models A$$

Временные операторы: Next

$$(\sigma, i) \models \mathbf{X}A \iff (\sigma, i+1) \models A$$



Накладывает условие на следующий момент времени.

Примеры:

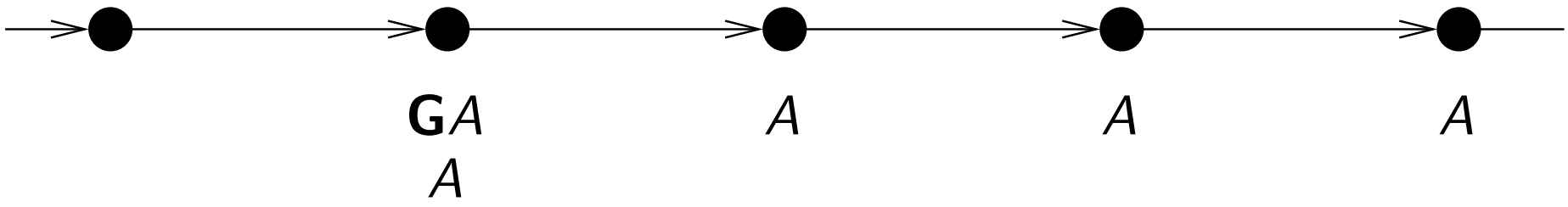
$$(is_friday \vee is_saturday) \rightarrow \mathbf{X}\neg work$$

$$(at_level_crossing_approach \wedge gate_up) \rightarrow \mathbf{X}on_level_crossing$$

$$ready \wedge \mathbf{X}steady \wedge \mathbf{XX}go$$

Временные операторы: Always

$$(\sigma, i) \models \mathbf{G}A \iff \text{для любого } j \in \mathbb{N}, \text{ если } j \geqslant i \text{ то } (\sigma, j) \models A$$



Оператор *always* накладывает условие на текущий и все последующие моменты времени

Пример:

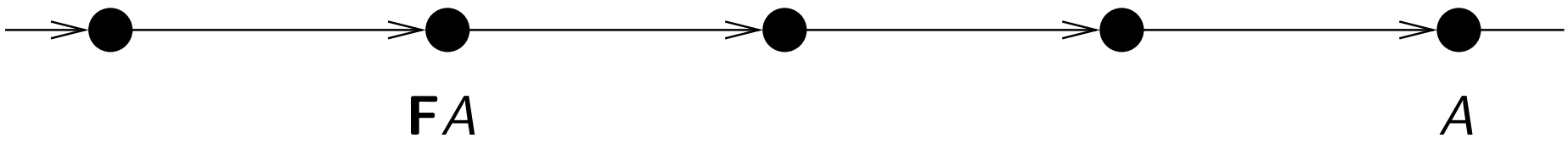
$$jane_holds_king_spades \rightarrow \mathbf{G}jane_holds_king_spades$$

$$\mathbf{G}temp_high$$

$$\mathbf{G}\neg(process_a_write_file \wedge process_b_write_file)$$

Временные операторы: Sometime

$$(\sigma, i) \models \mathbf{F}A \iff \text{существует } k \in \mathbb{N} \text{ такое что } k \geqslant i$$
$$\text{и } (\sigma, k) \models A$$



Оператор *sometime* требует, чтобы A было истинно в текущий момент времени, или когда-нибудь в будущем но не говорит когда

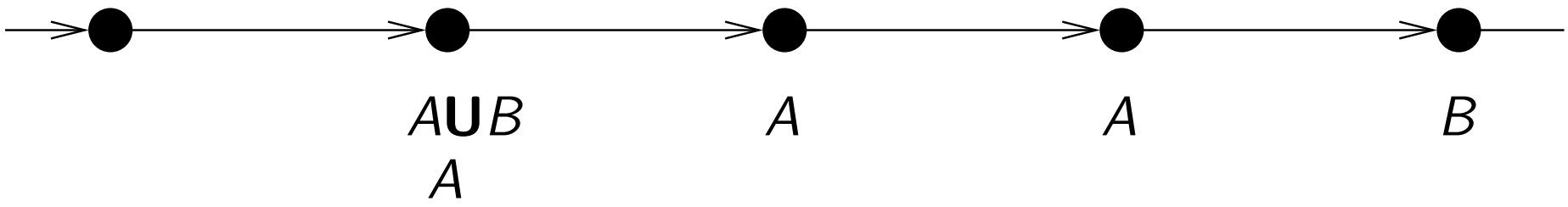
Пример:

$$request_to_print_doc \to \mathbf{F}print_doc$$

$$process_a_receives_msg_1 \to \mathbf{F}process_a_sends_msg_2$$

Временные операторы: Until

$(\sigma, i) \models A\mathbf{U}B \iff$ существует такое $k \in \mathbb{N}, k \geqslant i$ что
$(\sigma, k) \models B$ и для всех $j \in \mathbb{N}, i \leqslant j < k$ верно $(\sigma, j) \models A$

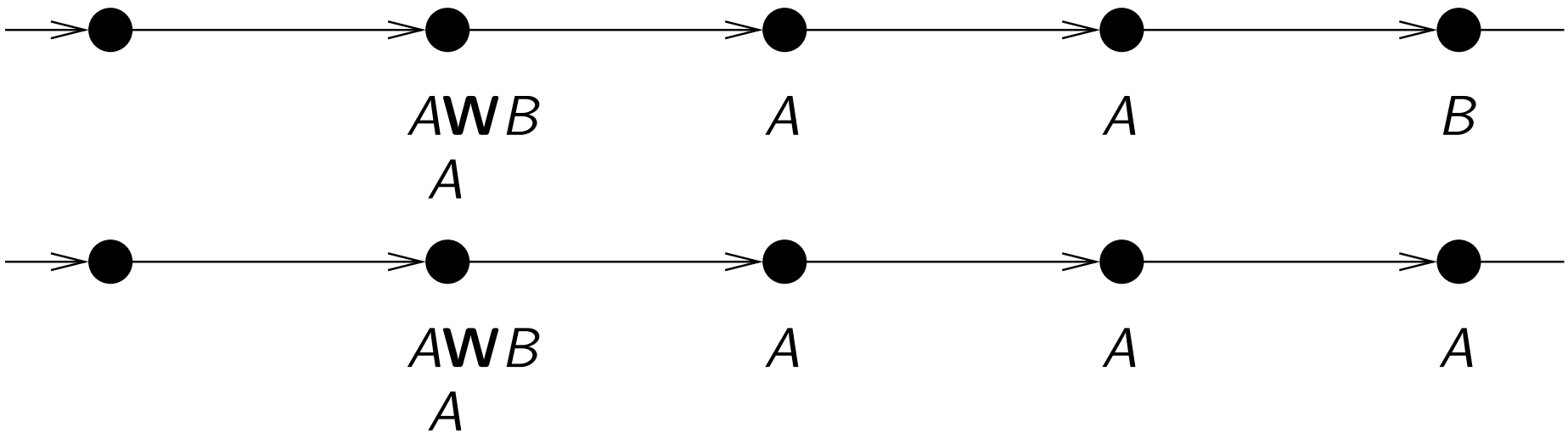


Аналогичен *sometime* так как B должен выполниться в текущий или какой-то будущий момент времени, но неизвестно когда.

Пример: $rainy\mathbf{U}monday$ $\quad$ $send_bit\mathbf{U}(receive_bit \vee abort)$

Временные операторы: Unless

$$(\sigma, i) \models A\mathbf{W}B \iff (\sigma, i) \models A\mathbf{U}B \text{ или } (\sigma, i) \models \mathbf{G}A$$



Оператор *unless* похож на **U** с той разницей, что B может так и не выполниться.

Пример:

$$program_executing\mathbf{W}program_terminating$$

Истинность в интерпретации, выполнимость, общезначимость

- Формула φ истинна в интерпретации σ если она верна в начальный момент времени

$$\sigma \models \varphi \iff (\sigma, 0) \models \varphi.$$

- Формула φ выполнима если существует интерпретация, в которой она истинна.
- Формула φ общезначима если она истинна в любой интерпретации

$$\models \varphi \iff \forall \sigma (\sigma \models \varphi).$$

LTL в верификации

- Система переходов моделирует реальную систему
- LTL описывает свойства модели
  - “Каждый процесс, который хочет войти в критическую секция, рано или поздно войдет в неё”

$$\mathbf{G}(\mathit{wants} \rightarrow \mathbf{F}\mathit{enters})$$

  - “каждое посланное сообщение будет получено”

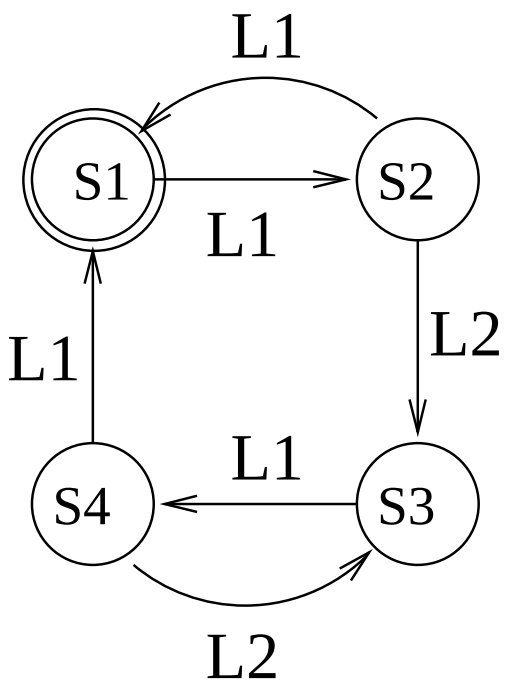
$$\mathbf{G}(\mathit{send} \rightarrow \mathbf{F}\mathit{receive})$$

  - “устройство готово к работе бесконечно часто”

$$\mathbf{GF}\mathit{ready}$$

- Проверка модели (Model Checking) состоит в проверке того, что система переходов обладает свойствами.

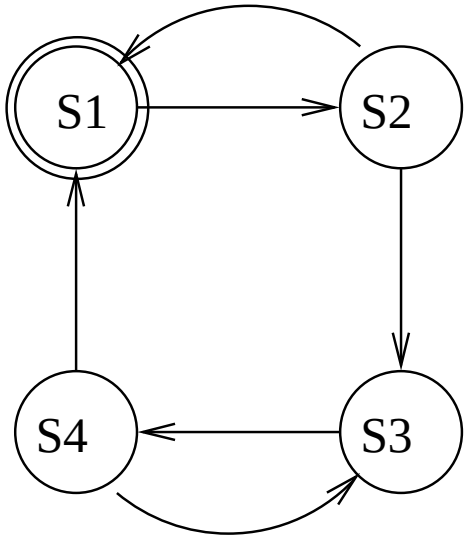
Система переходов (transition system)



Система переходов (Q, E, T, q_0, L)

- Q — конечное множество состояний
- E — конечное множество меток переходов
- $T \subseteq Q \times E \times Q$ — множество переходов
- q_0 — начальное состояние
- $L : Q \to Prop$ — пометка пропозициональными переменными

Система переходов (transition system)

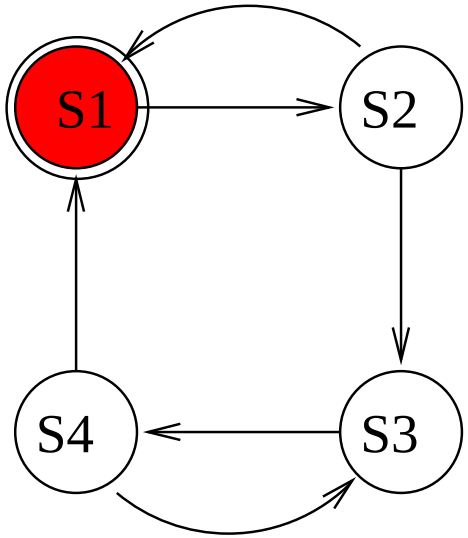


Система переходов (Q, T, q_0, L)

- Q — конечное множество состояний
- $T \subseteq Q \times Q$ — множество переходов
- q_0 — начальное состояние
- $L : Q \to Prop$ — пометка пропозициональными переменными

- Путь в системе переходов из состояния q — последовательность состояний q_i т.ч. $q_0 = q$ и $(q_i, q_{i+1}) \in T$

Система переходов (transition system)



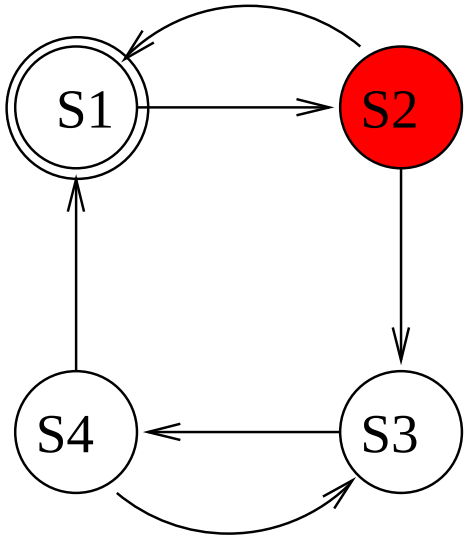
Система переходов (Q, T, q_0, L)

- Q — конечное множество состояний
- $T \subseteq Q \times Q$ — множество переходов
- q_0 — начальное состояние
- $L : Q \to Prop$ — пометка пропозициональными переменными

- Путь в системе переходов из состояния q — последовательность состояний q_i т.ч. $q_0 = q$ и $(q_i, q_{i+1}) \in T$

Путь $\pi = S1$,

Система переходов (transition system)



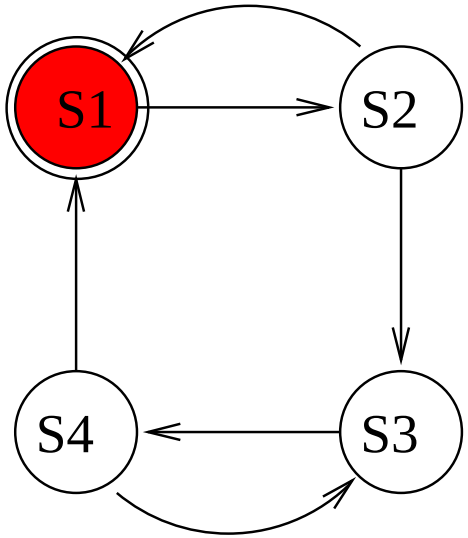
Система переходов (Q, T, q_0, L)

- Q — конечное множество состояний
- $T \subseteq Q \times Q$ — множество переходов
- q_0 — начальное состояние
- $L : Q \to Prop$ — пометка пропозициональными переменными

- Путь в системе переходов из состояния q — последовательность состояний q_i т.ч. $q_0 = q$ и $(q_i, q_{i+1}) \in T$

Путь $\pi = S1, S2,$

Система переходов (transition system)



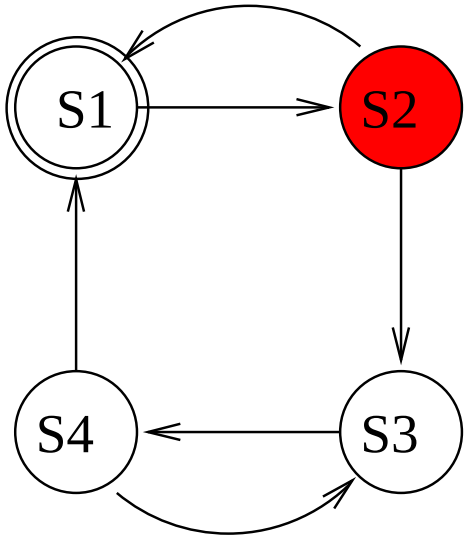
Система переходов (Q, T, q_0, L)

- Q — конечное множество состояний
- $T \subseteq Q \times Q$ — множество переходов
- q_0 — начальное состояние
- $L : Q \to Prop$ — пометка пропозициональными переменными

- Путь в системе переходов из состояния q — последовательность состояний q_i т.ч. $q_0 = q$ и $(q_i, q_{i+1}) \in T$

Путь $\pi = S1, S2, S1,$

Система переходов (transition system)



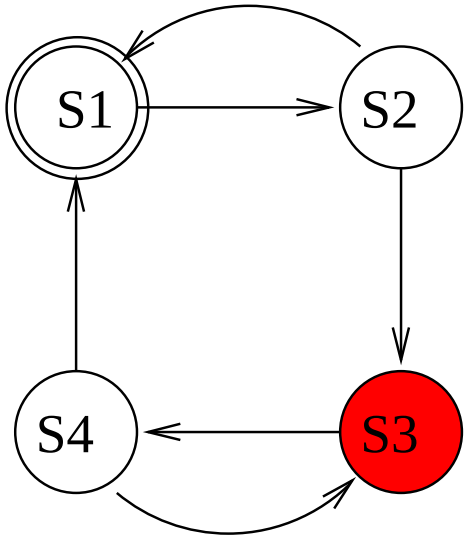
Система переходов (Q, T, q_0, L)

- Q — конечное множество состояний
- $T \subseteq Q \times Q$ — множество переходов
- q_0 — начальное состояние
- $L : Q \to Prop$ — пометка пропозициональными переменными

- Путь в системе переходов из состояния q — последовательность состояний q_i т.ч. $q_0 = q$ и $(q_i, q_{i+1}) \in T$

Путь $\pi = S1, S2, S1, S2,$

Система переходов (transition system)



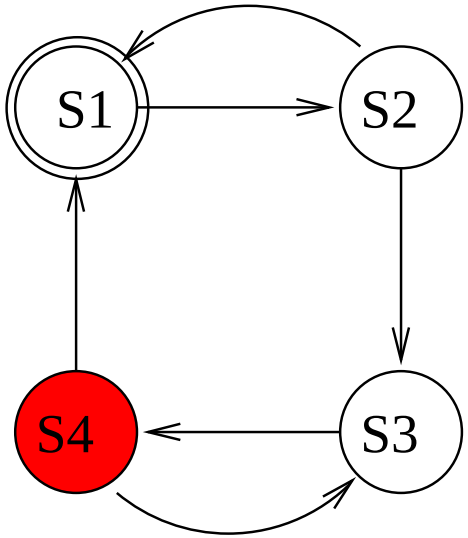
Система переходов (Q, T, q_0, L)

- Q — конечное множество состояний
- $T \subseteq Q \times Q$ — множество переходов
- q_0 — начальное состояние
- $L : Q \to Prop$ — пометка пропозициональными переменными

- Путь в системе переходов из состояния q — последовательность состояний q_i т.ч. $q_0 = q$ и $(q_i, q_{i+1}) \in T$

Путь $\pi = S1, S2, S1, S2, S3,$

Система переходов (transition system)



Система переходов (Q, T, q_0, L)

- Q — конечное множество состояний
- $T \subseteq Q \times Q$ — множество переходов
- q_0 — начальное состояние
- $L : Q \to Prop$ — пометка пропозициональными переменными

- Путь в системе переходов из состояния q — последовательность состояний q_i т.ч. $q_0 = q$ и $(q_i, q_{i+1}) \in T$

Путь $\pi = S1, S2, S1, S2, S3, S4 \ldots$

Пути и модели

- Путь $\pi = \{q_i\}_{i \in \mathbb{N}}$ в системе переходов можно рассматривать как интерпретацию временной логики:

$$\sigma_\pi = L(q(0)), L(q(1)), L(q(2)), \ldots$$

  последовательность меток вершин пути.
- Система переходов $S = (Q, T, q_0, L)$ обладает свойством φ, если для любого пути π в системе S из состояния q_0

$$\sigma_\pi \models \varphi$$

- Как это проверить?
  - Насколько это просто?

Совсем непросто...

Пусть $G = (V, E)$ — ориентированный граф.
Построим систему переходов $S = (Q, T, q_0, L)$:

- $Q = V \cup \{q_0, q_f\}$
- $T = E \cup \{(q_0, v) \mid v \in V\} \cup \{(v, q_f \mid v \in V)\} \cup \{(q_f, q_f)\}$
- $L(v_i) = \{p_i\}$; $L(q_0) = L(q_f) = \{\}$

$$S \models (\mathbf{F}p_1 \wedge \cdots \wedge \mathbf{F}p_n \wedge \mathbf{G}(p_1 \rightarrow \mathbf{G}\neg p_1) \wedge \cdots \wedge \mathbf{G}(p_n \rightarrow \mathbf{G}\neg p_n))$$

тогда и только тогда, когда в графе G существует путь проходящий через все вершины, но не проходящий ни через одну вершину дважды.

- Показали, что задача NP-трудная
- На самом деле, PSPACE-полная