Автоматы Бюхи и регулярные ω-языки

Занятие №7

Поскольку время бесконечно, до настоящего момента уже протекла бесконечность, т.е. всякое возможное развитие должно уже было осуществиться. Следовательно, наблюдаемое развитие должно быть повторением

Ф. Ницше.
Воля к власти

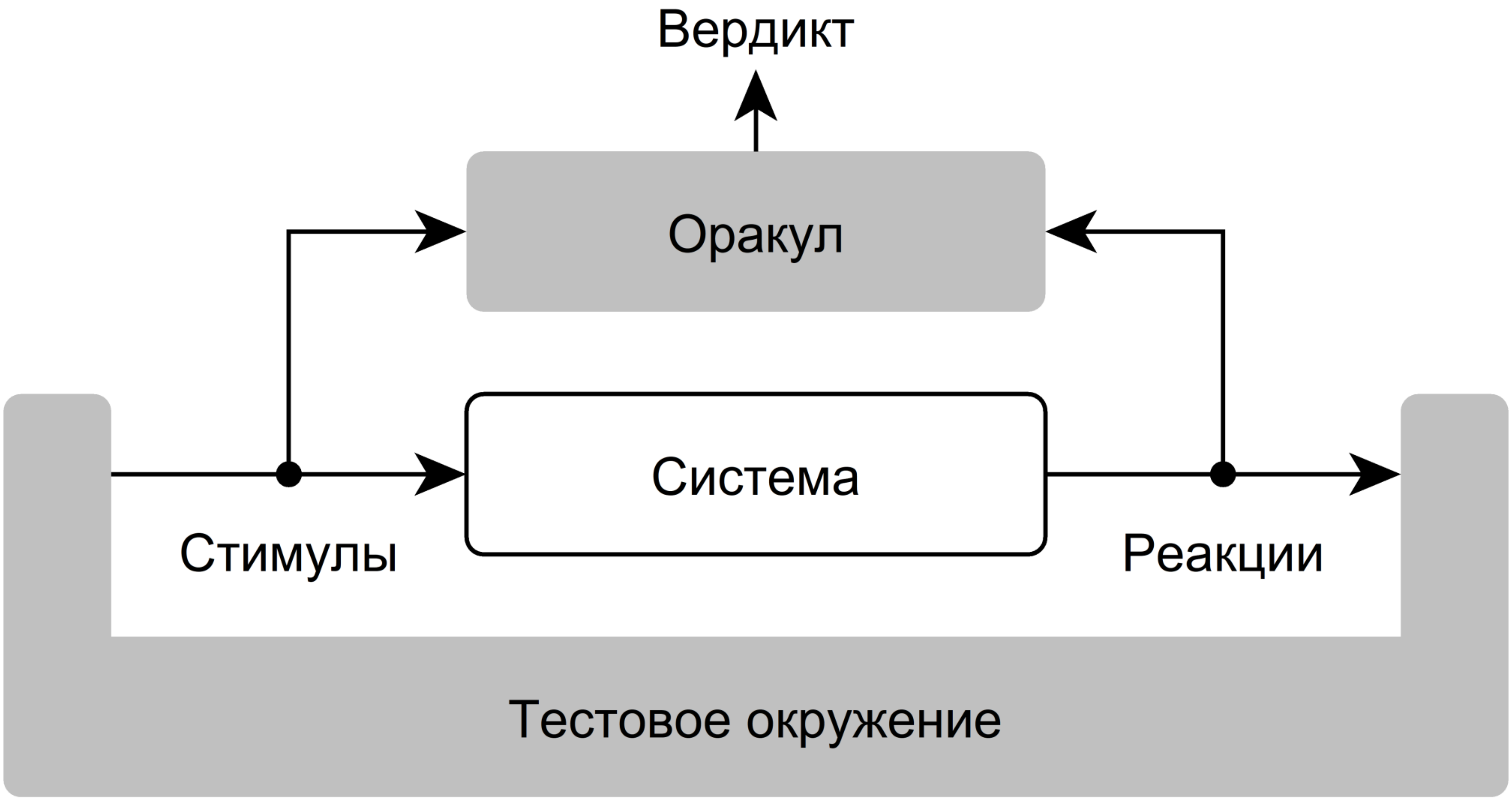
Александр Сергеевич Камкин

kamkin@ispras.ru

Еще раз о постановке задачи

- **Дано**
  - M – структура Крипке (модель программы)
  - φ – формула LTL (модель требований)
- **Требуется**
  - Проверить $M \models \varphi$ (истинность формулы на модели)
  - Если ответ отрицательный, предъявить *контрпример*

Проверка требований при тестировании



Формулировка в терминах теории языков

- **Пусть**
  - $\mathcal{L}_M$– язык модели (все возможные траектории)
  - $\mathcal{L}_{\neg\varphi}$– язык оракула (все ошибочные траектории)
- **Требуется**
  - Проверить $\mathcal{L}_M \cap \mathcal{L}_{\neg\varphi} = \emptyset$
  - Если $\mathcal{L}_M \cap \mathcal{L}_{\neg\varphi} \neq \emptyset$, предъявить контрпример

Конечные автоматы (распознаватели)

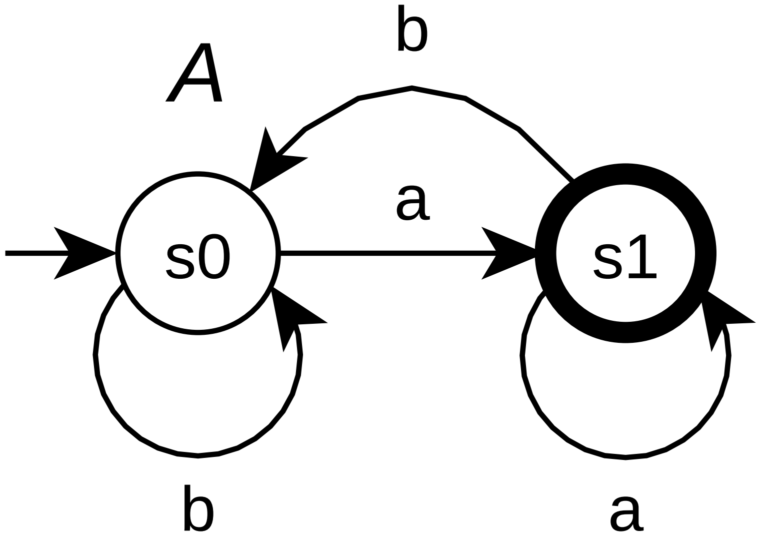
Недетерминированный конечный автомат – $\langle S, \Sigma, S_0, \delta, \mathcal{F} \rangle$

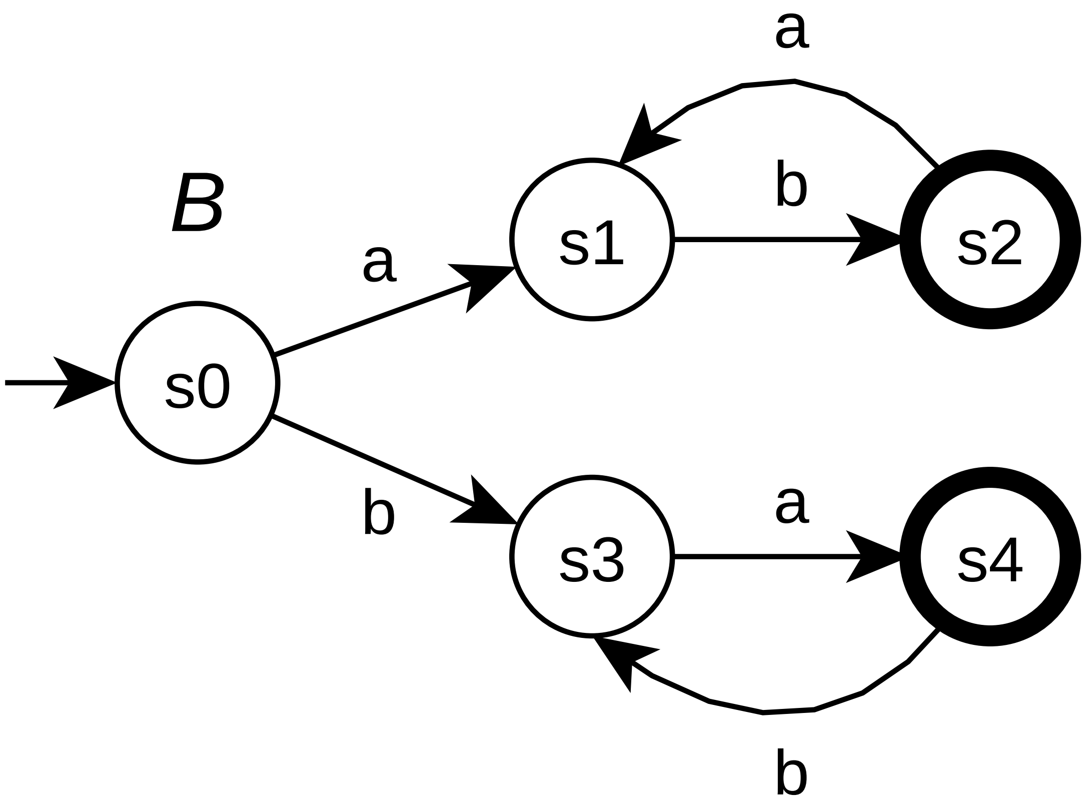
- S — множество состояний
- Σ — входной алфавит
- $S_0 \subseteq S$ — непустое множество начальных состояний
- $\delta: S \times \Sigma \to 2^S$ — функция переходов
- $\mathcal{F} \subseteq S$ — множество допускающих состояний

Автомат *допускает* слово $w = x_0 x_1 \dots x_{n-1} \in \Sigma^*$, если существует вычисление $\pi = \{s_i\}_{i=0}^{n}$, такое что

- $s_0 \in S_0$
- $s_i \in \delta(s_{i-1}, x_{i-1})$ для всех $1 \leq i \leq n$
- $s_n \in \mathcal{F}$

Примеры конечных автоматов

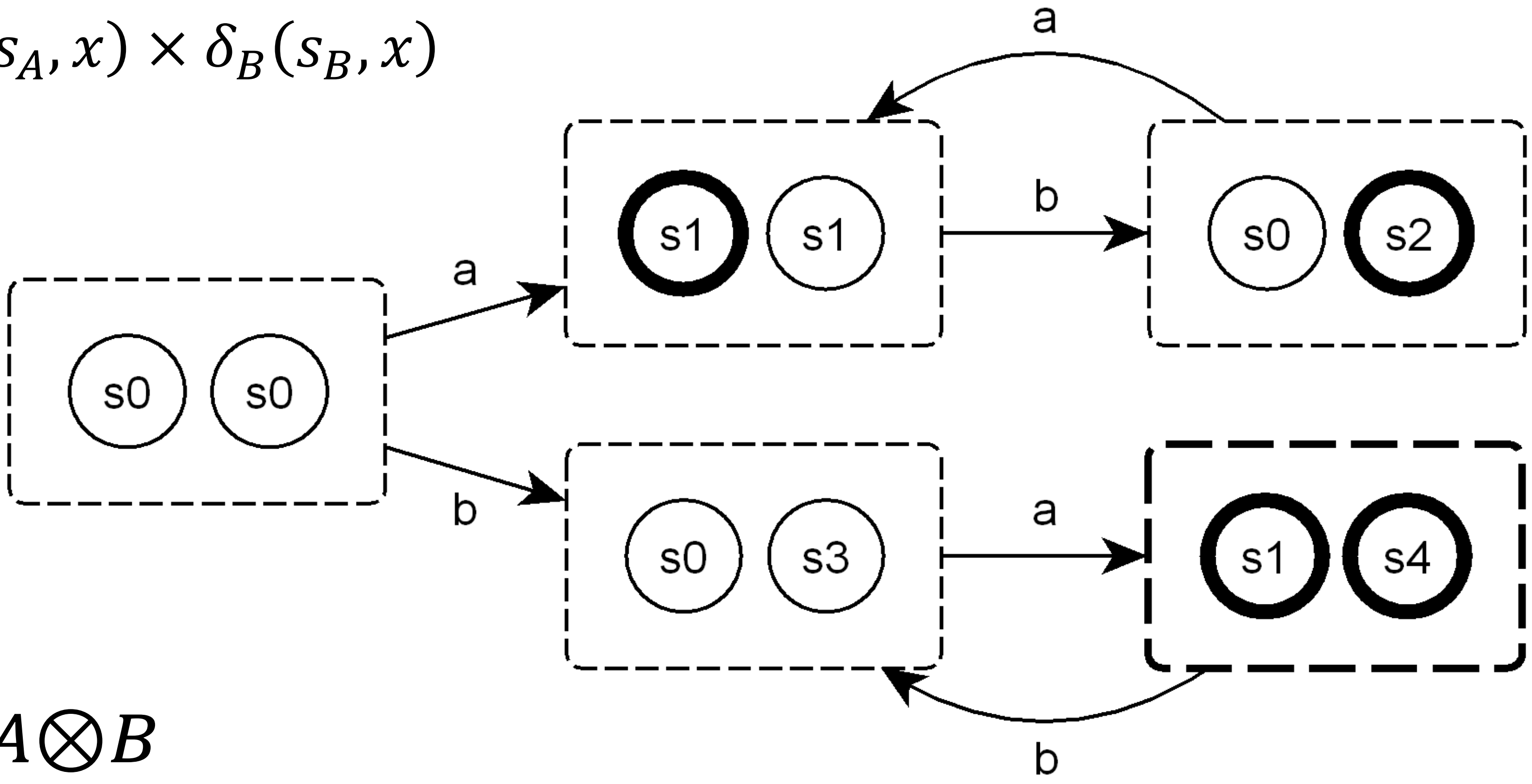




Пересечение регулярных языков

$$A \otimes B = \langle S_A \times S_B, \Sigma, S_{A0} \times S_{B0}, \delta_{AB}, \mathcal{F}_A \times \mathcal{F}_B \rangle$$

$$\delta_{AB}\big((s_A, s_B), x\big) = \delta_A(s_A, x) \times \delta_B(s_B, x)$$

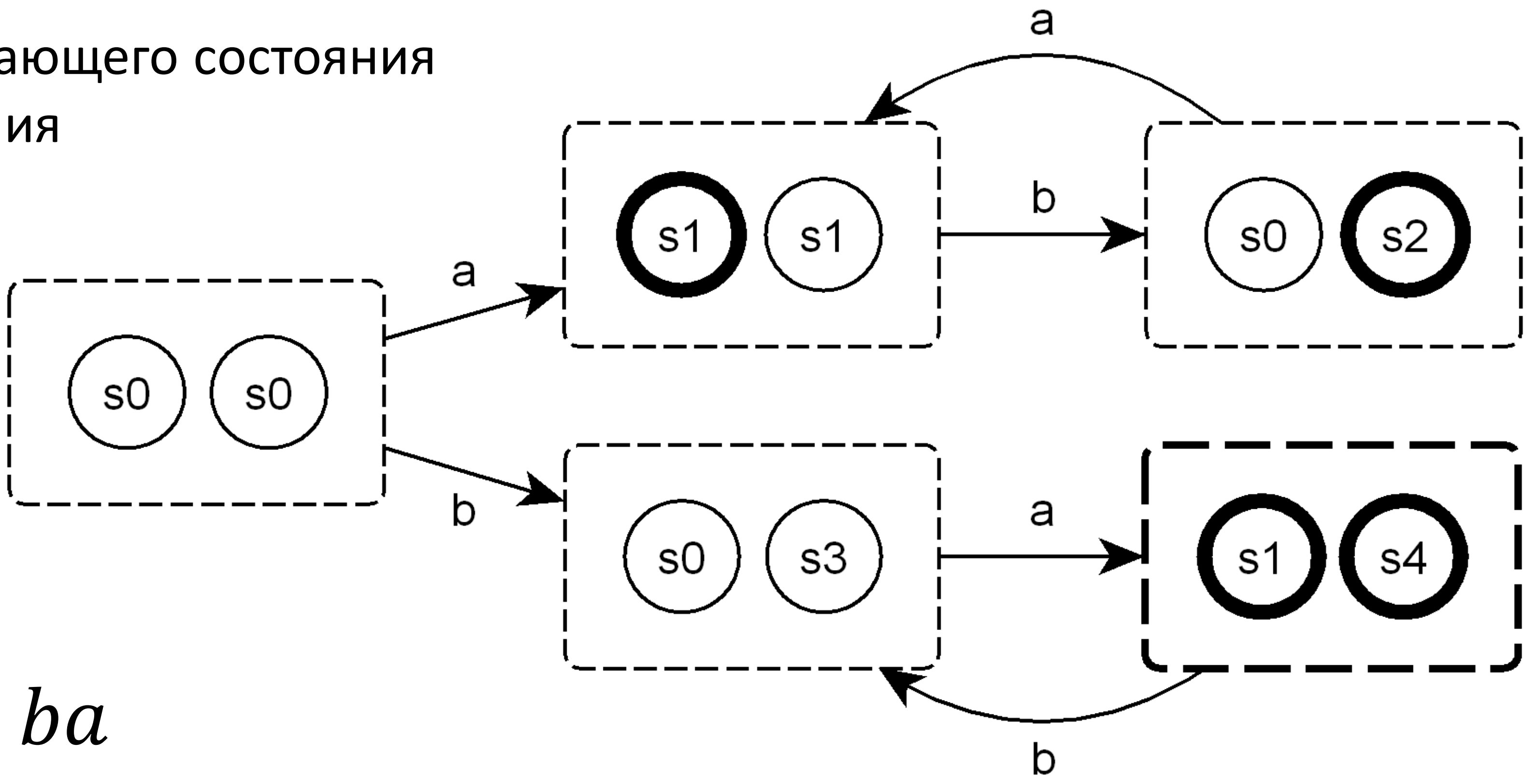


$$\mathcal{L}_A \cap \mathcal{L}_B = \mathcal{L}_{A \otimes B}$$

Проверка автоматного языка на пустоту

$\mathcal{L}_A \cap \mathcal{L}_B \neq \emptyset$

Достижимость допускающего состояния из начального состояния



Контрпример – ba

Автоматы Бюхи

Недетерминированный автомат Бюхи – $\langle S, \Sigma, S_0, \delta, \mathcal{F} \rangle$

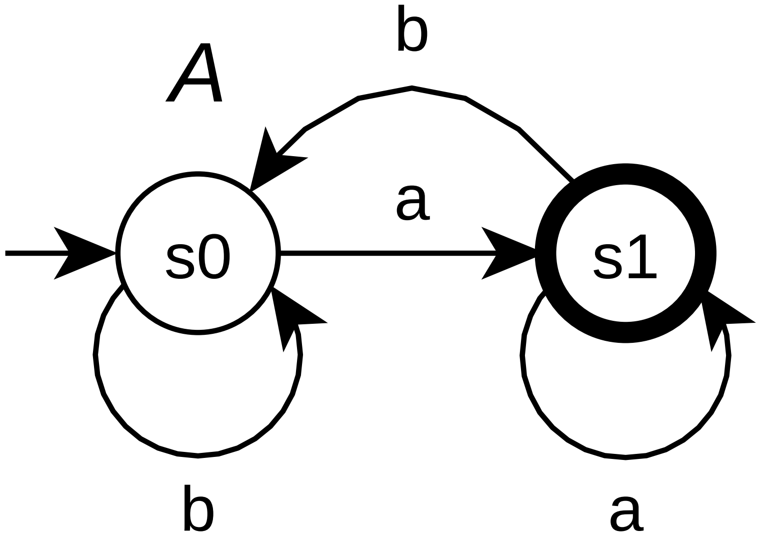
- S — множество состояний
- Σ — входной алфавит
- $S_0 \subseteq S$ — непустое множество начальных состояний
- $\delta: S \times \Sigma \to 2^S$ — функция переходов
- $\mathcal{F} \subseteq S$ — множество допускающих состояний

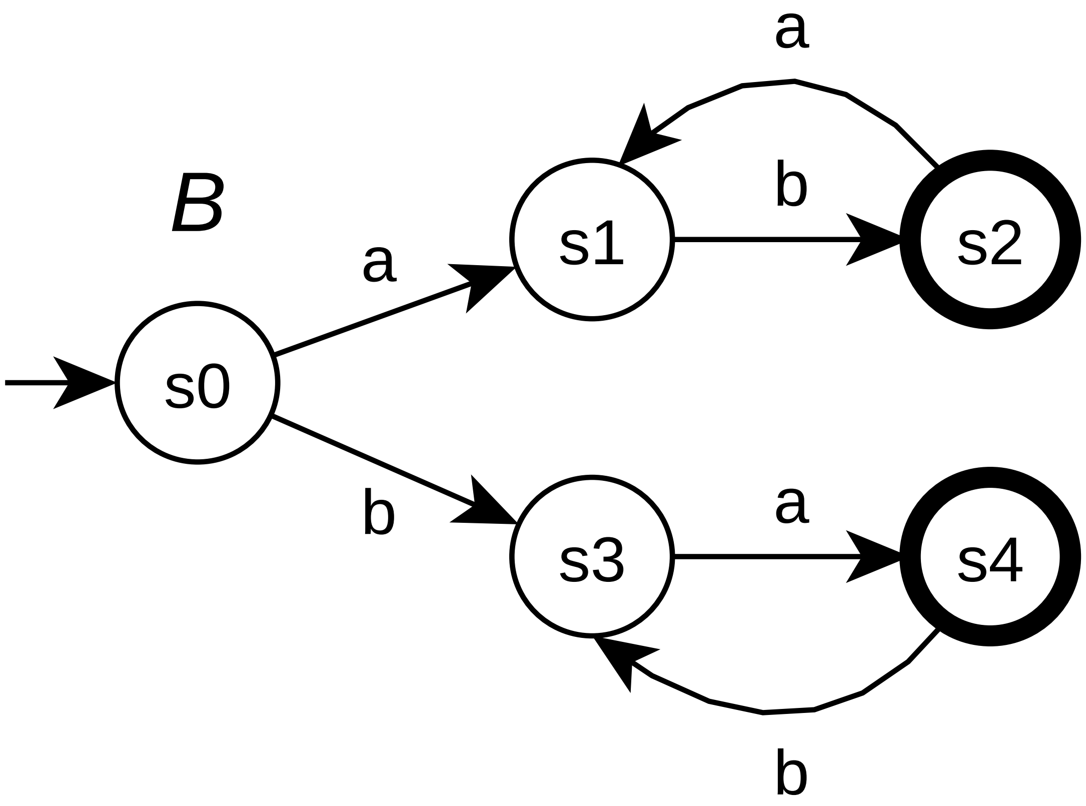
Julius Richard Büchi
(1924-1984)

Автомат *допускает* ω-слово $w = x_0 x_1 \ldots \in \Sigma^\omega$, если существует вычисление $\pi = \{s_i\}_{i=0}^{\infty}$, такое что

- $s_0 \in S_0$
- $s_i \in \delta(s_{i-1}, x_{i-1})$ для всех $i \geq 1$
- $\mathrm{inf}(\pi) \cap \mathcal{F} \neq \emptyset$ — хотя бы одно допускающее состояние встречается ∞ число раз

Примеры автоматов Бюхи





Обобщенный автомат Бюхи

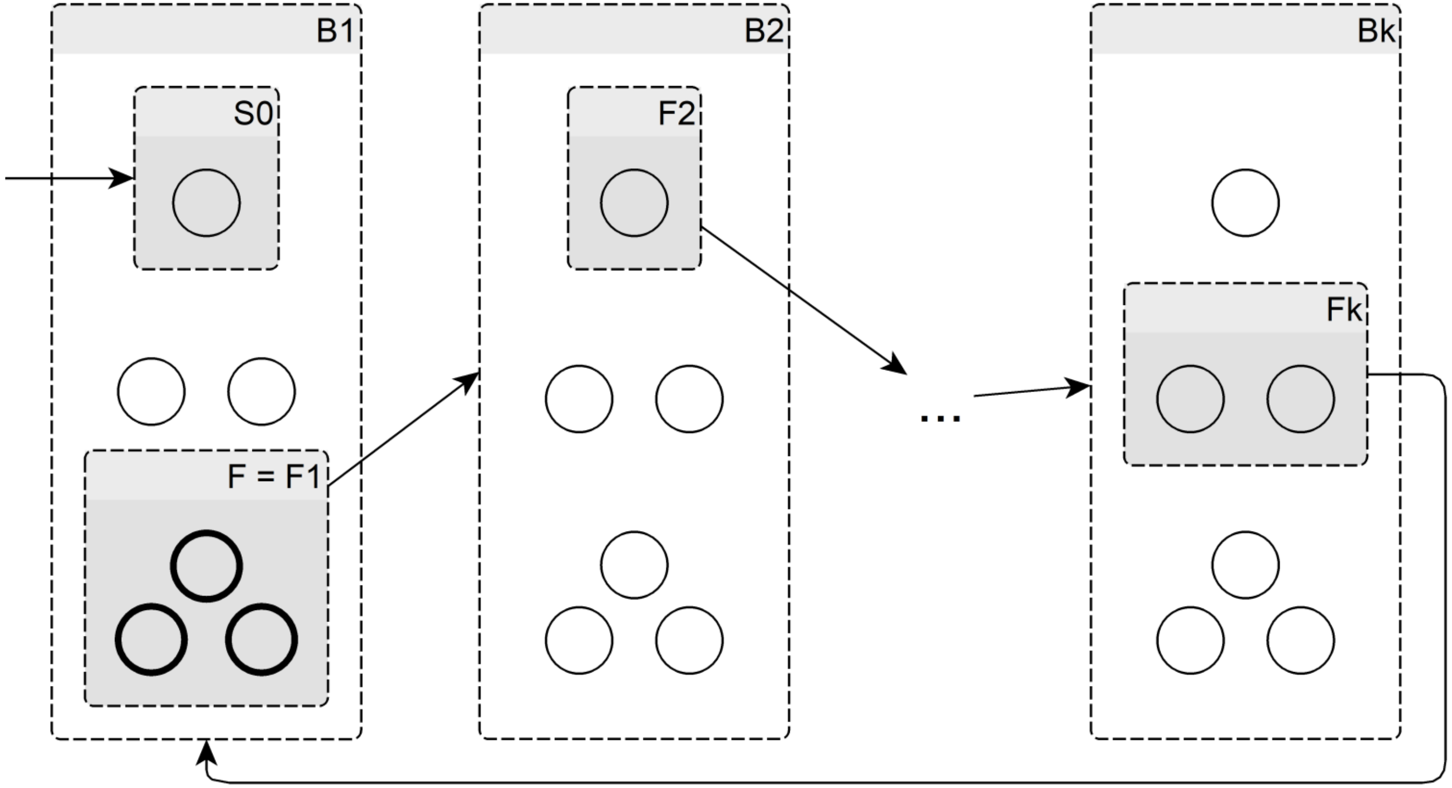
Обобщенный автомат Бюхи – пятерка $\langle S, \Sigma, S_0, \delta, \{\mathcal{F}_1, \dots, \mathcal{F}_k\}\rangle$

- S — множество состояний
- Σ — входной алфавит
- $S_0 \subseteq S$ — непустое множество начальных состояний
- $\delta: S \times \Sigma \to 2^S$ — функция переходов
- $\{\mathcal{F}_1, \dots, \mathcal{F}_k\}$ — семейство множеств допускающих состояний

Автомат *допускает* ω-слово $w = x_0 x_1 \dots \in \Sigma^\omega$, если существует вычисление $\pi = \{s_i\}_{i=0}^n$, такое что

- $s_0 \in S_0$
- $s_i \in \delta(s_{i-1}, x_{i-1})$ для всех $i \geq 1$
- $\mathrm{inf}(\pi) \cap \mathcal{F}_j \neq \emptyset$ для всех $1 \leq j \leq k$

Построение автомата Бюхи по обобщенному



Обобщенный автомат Бюхи

$G = \langle S, \Sigma, S_0, \delta, \boldsymbol{\mathcal{F}} = \{\mathcal{F}_1, \dots, \mathcal{F}_k\} \rangle$

Эквивалентный ему автомат Бюхи

$B = \langle S \times \{1, \dots, k\}, \Sigma, S_0 \times \{1\}, \delta_B, \mathcal{F}_1 \times \{1\} \rangle$

k копий автомата G

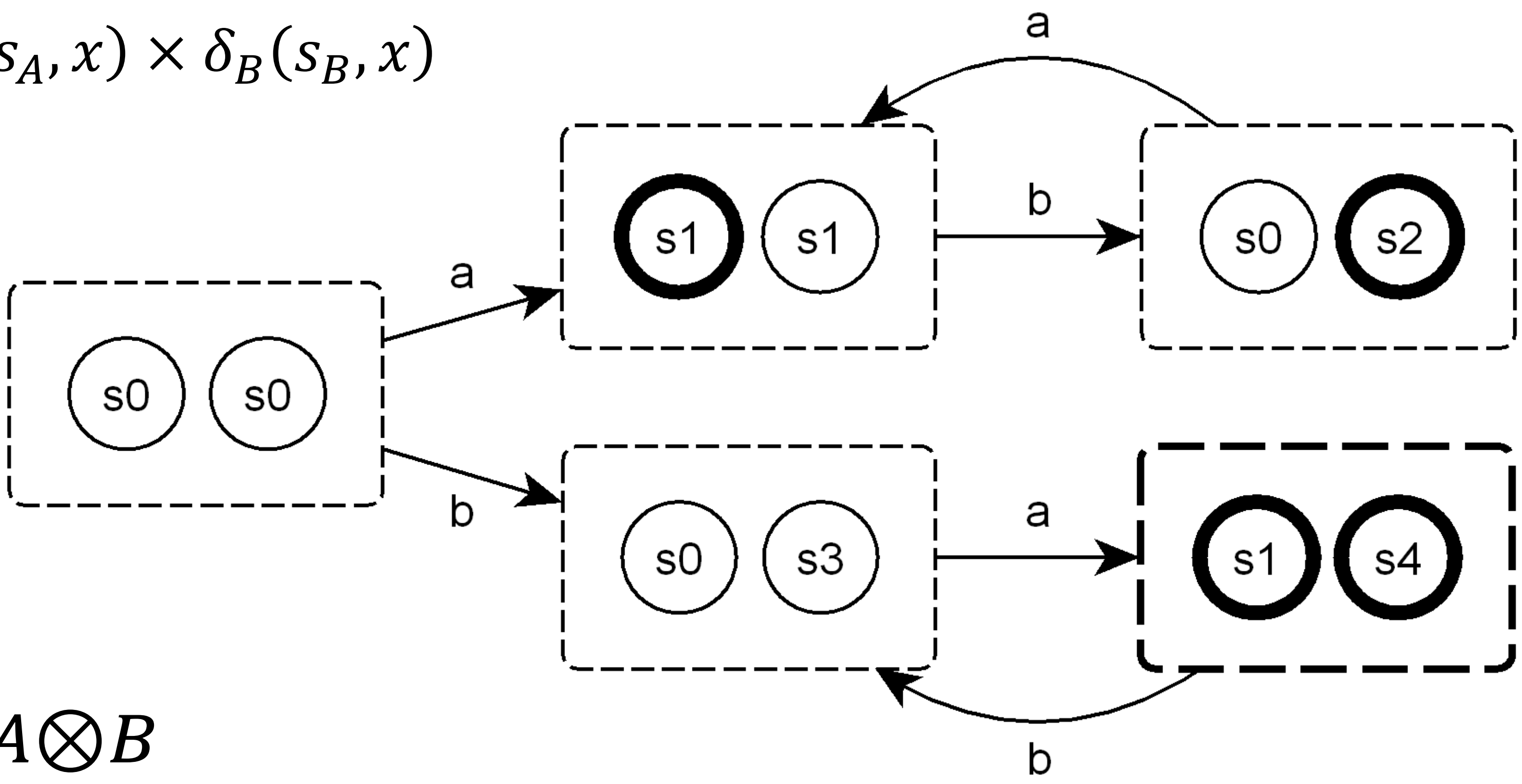
$$\delta_B\big((s, i), x\big) = \{(s', i) \mid s' \in \delta(s, x)\}, \text{ если } s \notin \mathcal{F}_i$$

$$\delta_B\big((s, i), x\big) = \{(s', (i \,\%\, k) + 1) \mid s' \in \delta(s, x)\}, \text{ иначе}$$

Пересечение регулярных ω-языков

$$A \otimes B = \langle S_A \times S_B, \Sigma, S_{A0} \times S_{B0}, \delta_{AB}, \{\mathcal{F}_A \times S_B, S_A \times \mathcal{F}_B\}\rangle$$

$$\delta_{AB}\big((s_A, s_B), x\big) = \delta_A(s_A, x) \times \delta_B(s_B, x)$$

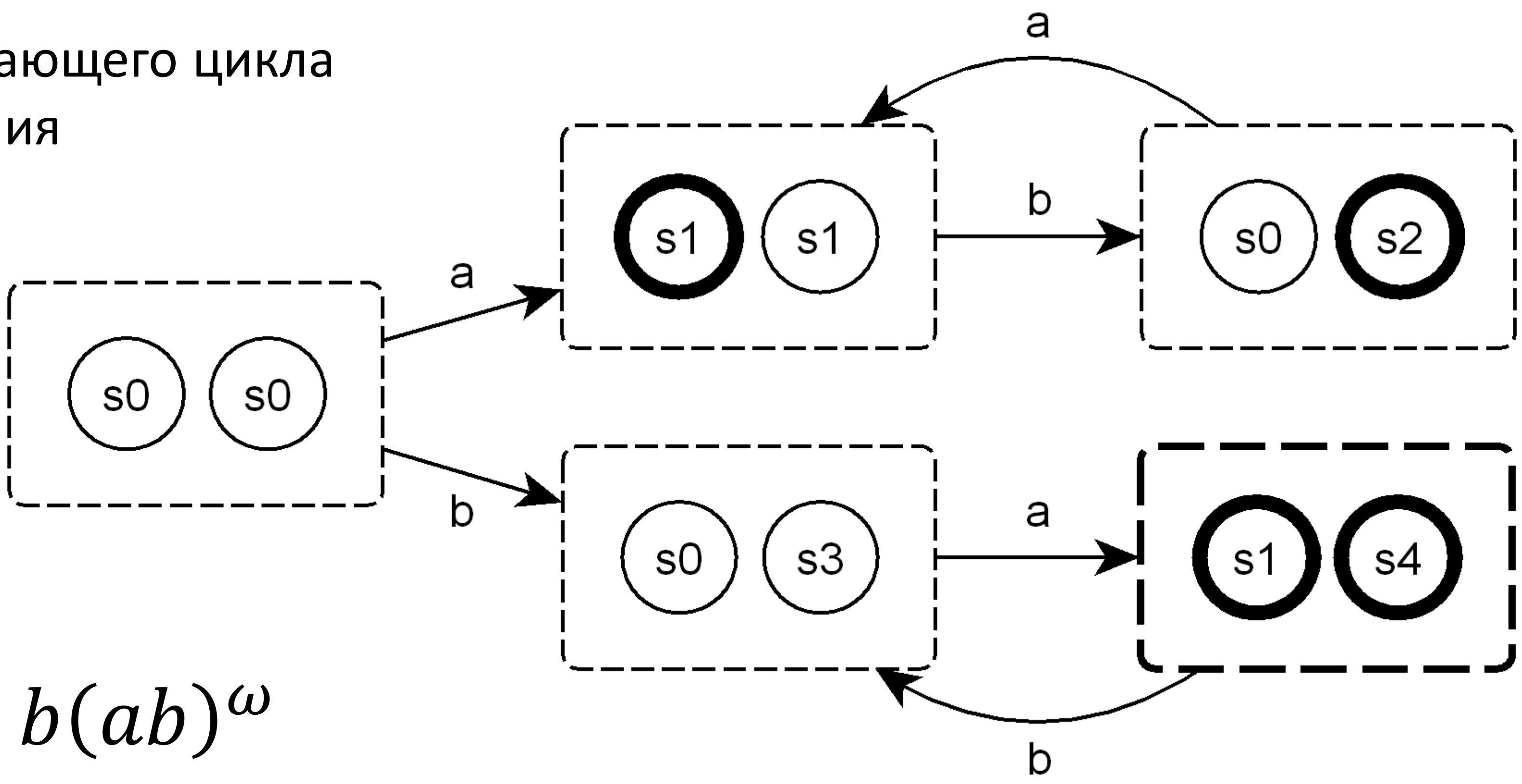


$$\mathcal{L}_A \cap \mathcal{L}_B = \mathcal{L}_{A\otimes B}$$

Проверка автоматного языка на пустоту

$\mathcal{L}_A \cap \mathcal{L}_B \neq \emptyset$

Достижимость допускающего цикла
из начального состояния



Контрпример – $b(ab)^{\omega}$

Методы поиска допускающего цикла

- **Построение компонент сильной связности (SCC)**
  - Алгоритм Тарьяна (1972)
  - Допускающее состояние $\in$ нетривиальной SCC
- **Вложенный поиск в глубину (nested DFS)**
  - Используется в инструменте SPIN
  - Возможность работы “на лету” в процессе обхода графа

Robert Endre Tarjan
(1948)

Вложенный поиск в глубину

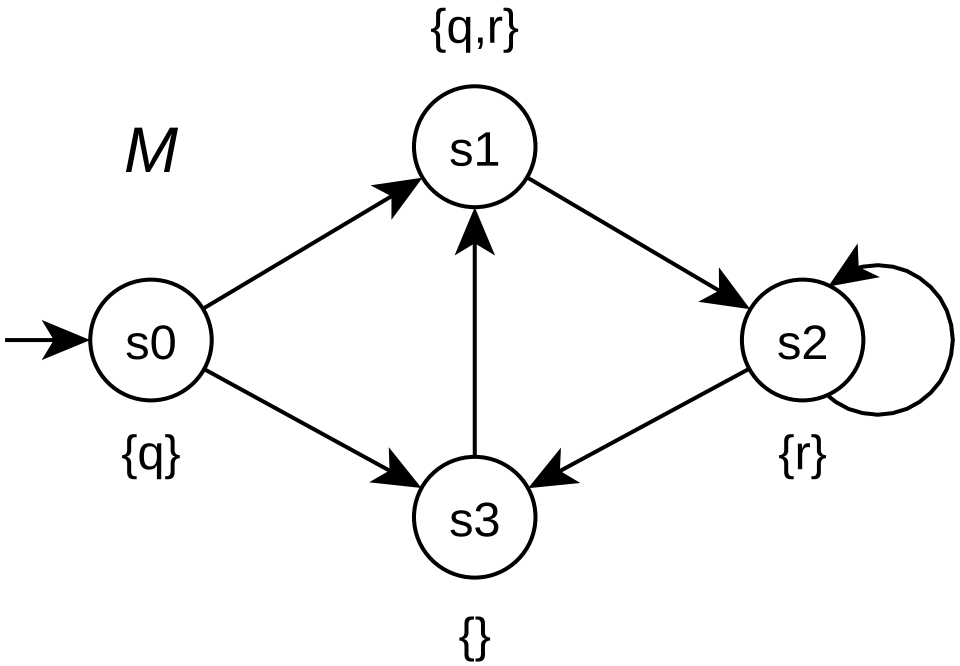
Основной поиск	Вложенный поиск: вызов $ndfs(s_F)$

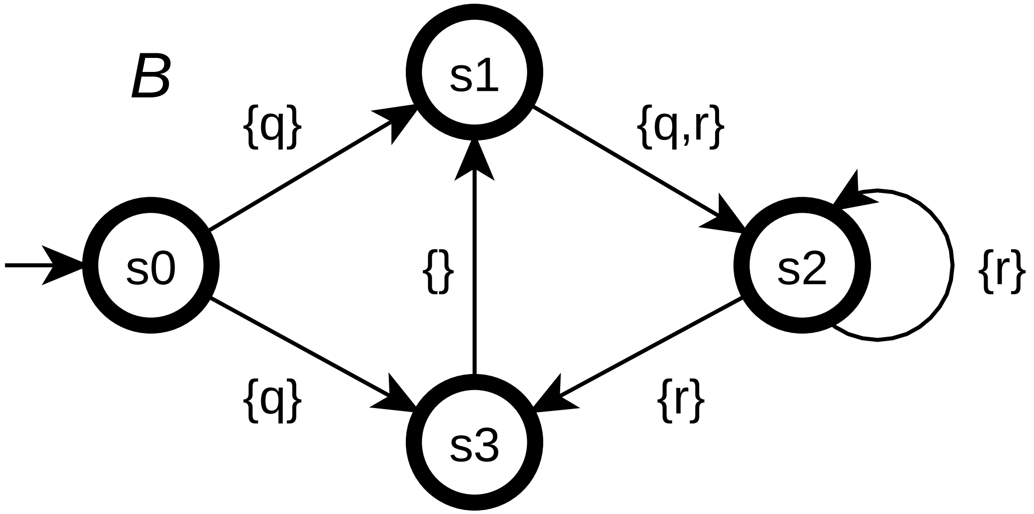
<pre>S := {(s₀,0)};
dfs_stack := {s₀};
while dfs_stack ≠ ∅ **do**
  s := dfs_stack.top();
  if (succ(s,0) \ S) = ∅ **then**
    if accepting(s) **then**
      if ndfs(s) **then**
        return true // цикл найден
      end
    end;
    dfs_stack.pop()
  else
    (s',0) := choose(succ(s,0) \ S);
    S := S ∪ {(s',0)};
    dfs_stack.push(s')
  end
end;
return false // цикл не найден</pre>
	<pre>S := S ∪ {(s_F,1)};
	ndfs_stack := {s_F};
	while ndfs_stack ≠ ∅ **do**
	  s := ndfs_stack.top();
	  if succ(s) ∩ dfs_stack ≠ ∅ **then**
	    return true // цикл найден
	  end;
	  if (succ(s,1) \ S) = ∅ **then**
	    ndfs_stack.pop()
	  else
	    (s',1) := choose(succ(s,1) \ S);
	    S := S ∪ {(s',1)};
	    ndfs_stack.push(s')
	  end
	end;
	return false // цикл не найден</pre>

Преобразование структуры Крипке в автомат

$$M = \langle S, S_0, R, L \rangle$$

$$B_M = \langle S, 2^{AP}, S_0, \delta, S \rangle$$

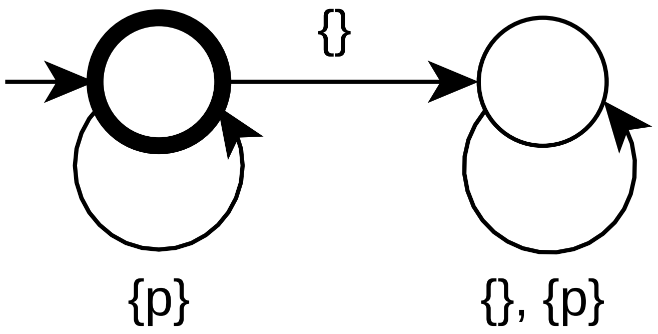




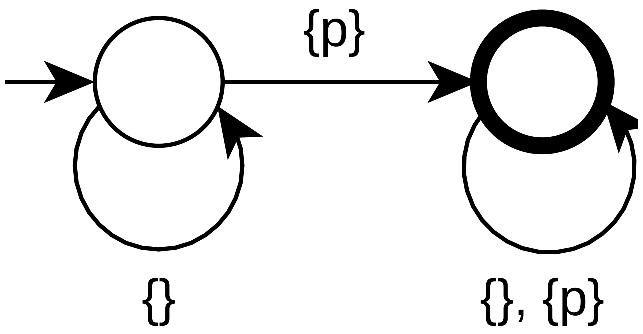
$$\delta\big(s, L(s)\big) = \{s' \mid (s, s') \in R\}$$

Примеры автоматов Бюхи для формул LTL

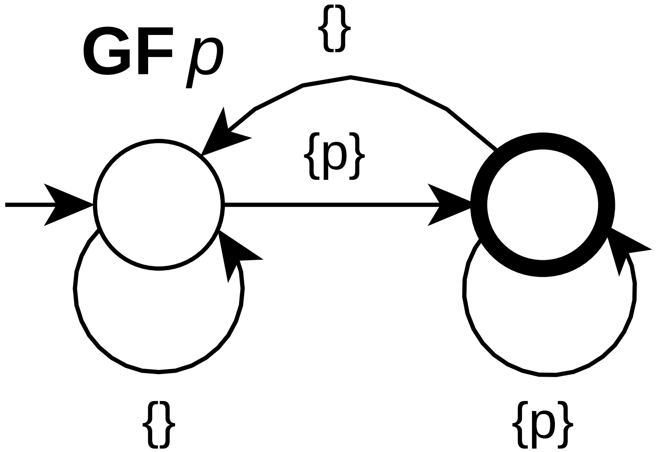
G *p*



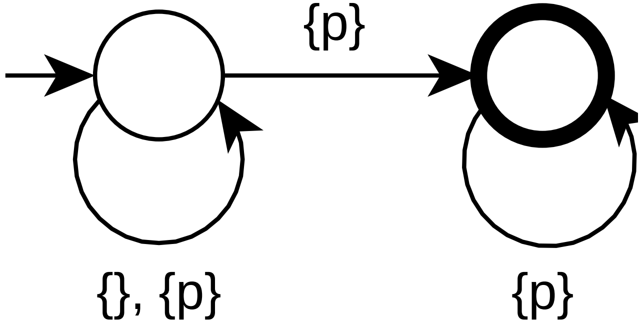
F *p*



GF *p*



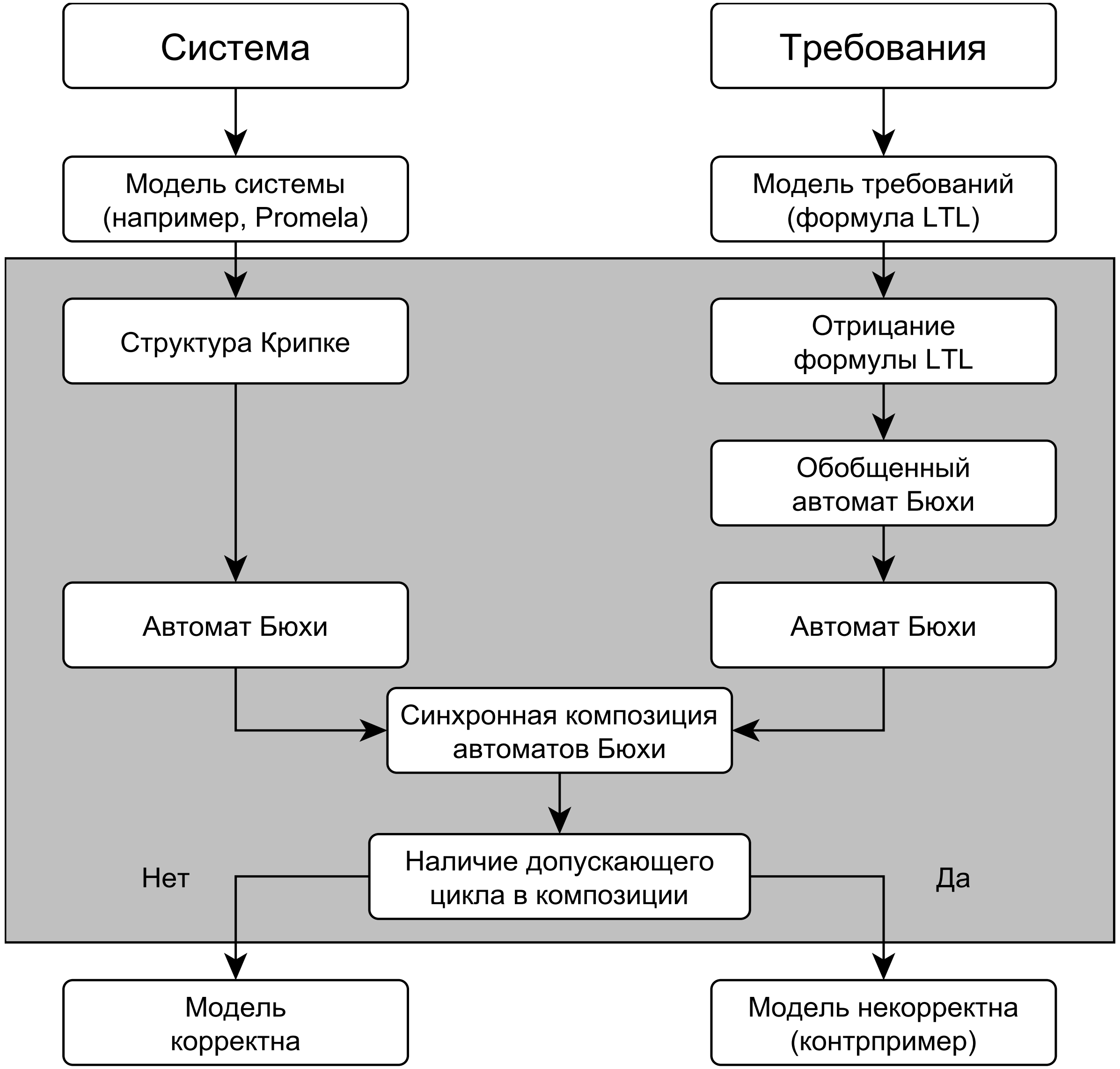
FG *p*



Общая схема теоретико-автоматного подхода

Moshe Y. Vardi
(1954)

Pierre Wolper
(1955)



Входные данные

Инструмент
проверки моделей
(например, Spin)

Преобразование обобщенного автомата Бюхи к обычному избыточно: проще искать несколько допускающих циклов

Результат