Модели вычислений

В.А. Захаров, Р.И. Подловченко

Лекция 10.

1. Конечные автоматы-преобразователи
2. Рациональные отношения и их свойства
3. Алгоритмические проблемы для автоматов-преобразователей
4. Детерминированные автоматы-преобразователи

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Автоматы могут не только распознавать языки, но и преобразовывать слова одного языка в слова другого.

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Автоматы могут не только распознавать языки, но и преобразовывать слова одного языка в слова другого.

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Автоматы могут не только распознавать языки, но и преобразовывать слова одного языка в слова другого.

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Автоматы могут не только распознавать языки, но и преобразовывать слова одного языка в слова другого.

Для моделирования управляющих систем вполне подходят автоматы Мура и Миля, схемы из функциональных элементов с задержкой и пр.

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

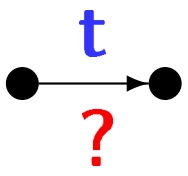
Представим себе работу воображаемого автомата.

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.

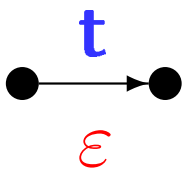


АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.

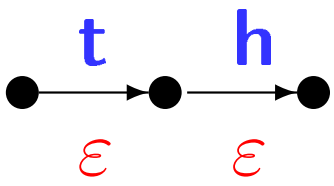


АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.

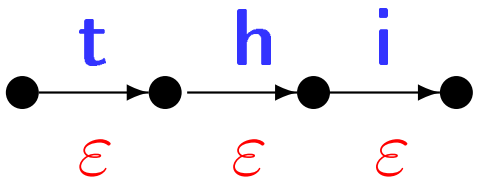


АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.

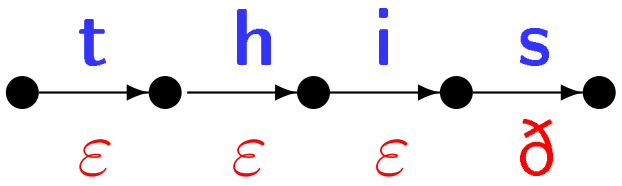


АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.

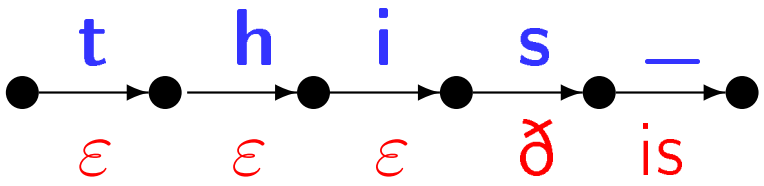


АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.

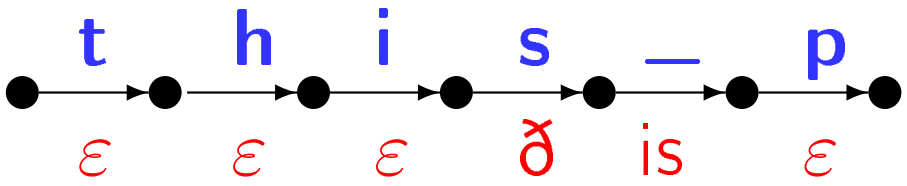


АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.

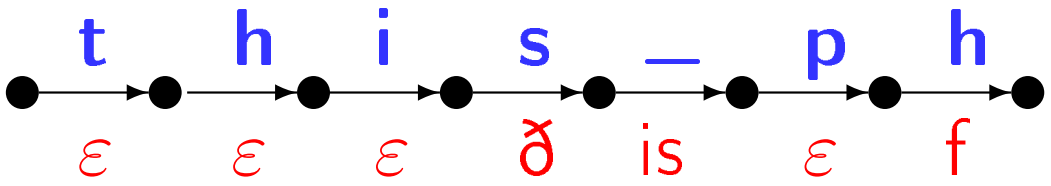


АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.

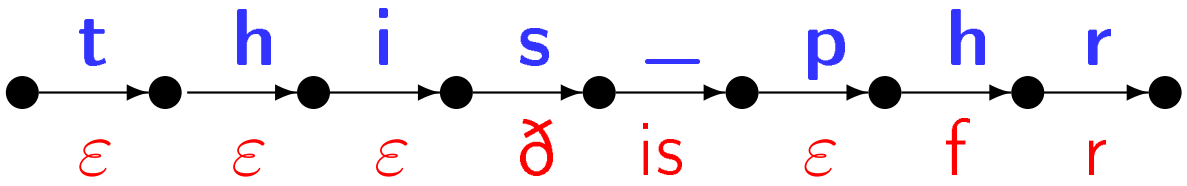


АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.

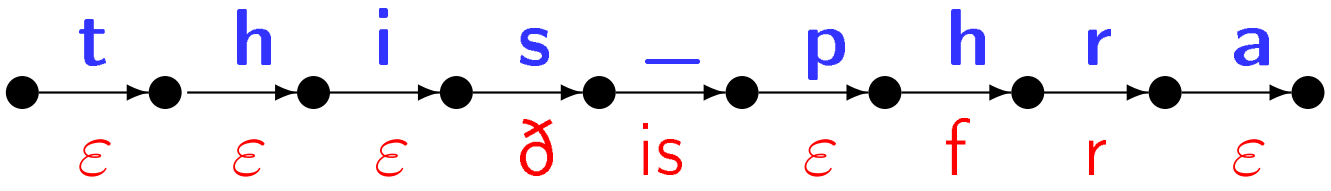


АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.

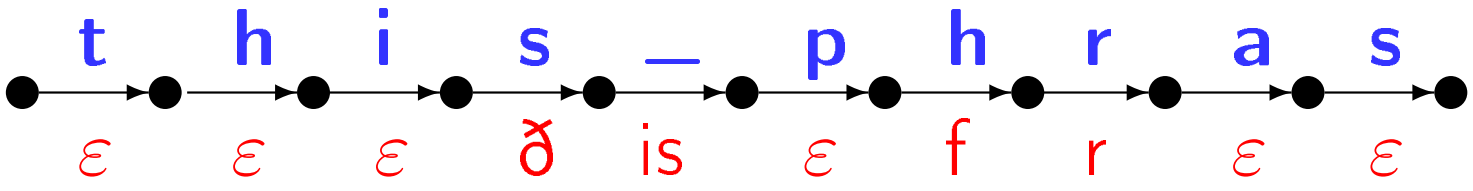


АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.

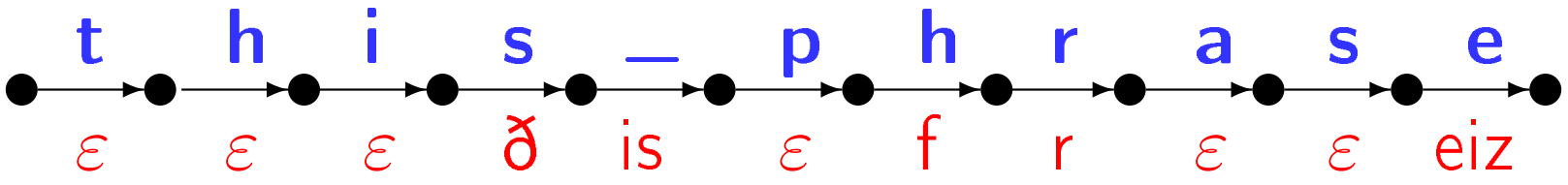


АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.

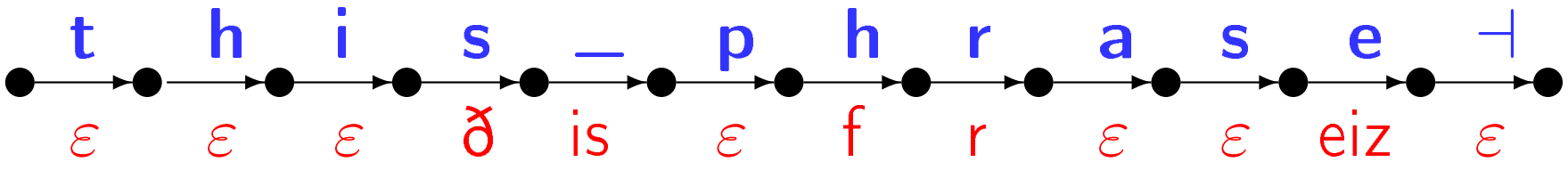


АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Но эти автоматные модели пригодны не для всех задач. Попробуем создать устройство озвучивания текста. Оно должно выполнять транскрипцию текста, т.е. преобразовывать слова в последовательности фонем.

Например, «this phrase» $\Longrightarrow$ ðisfreiz .

Представим себе работу воображаемого автомата.



АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Формально, конечный автомат-преобразователь — это система $\mathcal{A} = (\Sigma, \Delta, S, I, F, T)$, где

- Σ — конечный входной алфавит ;
- Δ — конечный выходной алфавит ;
- S — конечное множество состояний ;
- I — множество начальных состояний , $I \subseteq S$;
- F — множество финальных состояний , $F \subseteq S$;
- T — отношение переходов ,
  $T \subseteq S \times (\Sigma \cup \{\varepsilon\}) \times S \times \Delta^*$.

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Четверки (s', x, s'', β) из отношения переходов T будем называть переходами и изображать их записями вида $s' \xrightarrow{x,\beta} s''$.

Вычислением автомата $\mathcal{A}$ из состояния s_0 в состояние s_n называется любая конечная (в т.ч. пустая) последовательность переходов

$$run = s_0 \xrightarrow{x_1,\beta_1} s_1 \xrightarrow{x_2,\beta_2} \dots \xrightarrow{x_n,\beta_n} s_n \ .$$

Будем говорить, что вычисление run транслирует слово $w = x_1 x_2 \dots x_n$ в слово $\alpha = \beta_1 \beta_2 \dots \beta_n$, и условимся обозначать это вычисление записью $s_0 \xrightarrow{w,\alpha}_* s_n$.

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

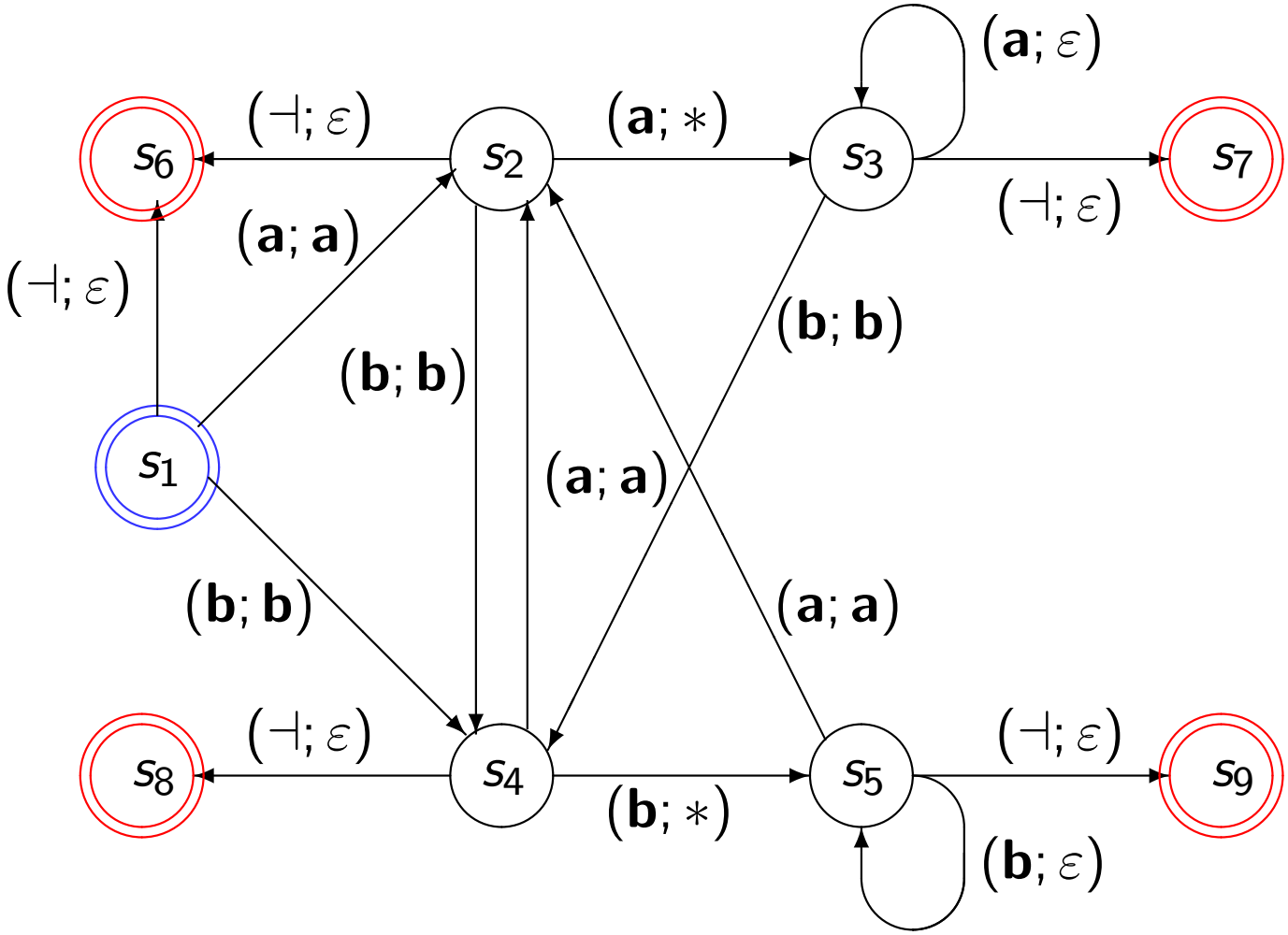
Конечный автомат-преобразователь $\mathcal{A} = (\Sigma, \Delta, S, I, F, T)$ реализует отношение $R(\mathcal{A}) \subseteq \Sigma^* \times \Delta^*$

$$R(\mathcal{A}) = \{(w, \alpha) : s_0 \xrightarrow{w,\alpha}_* s_n, s_0 \in I, s_n \in F\}$$

Отношение $R \subseteq \Sigma^* \times \Delta^*$ называется рациональным отношением над алфавитами Σ, Δ , если существует конечный автомат-преобразователь, реализующий это отношение, т.е. $R = R(\mathcal{A})$.

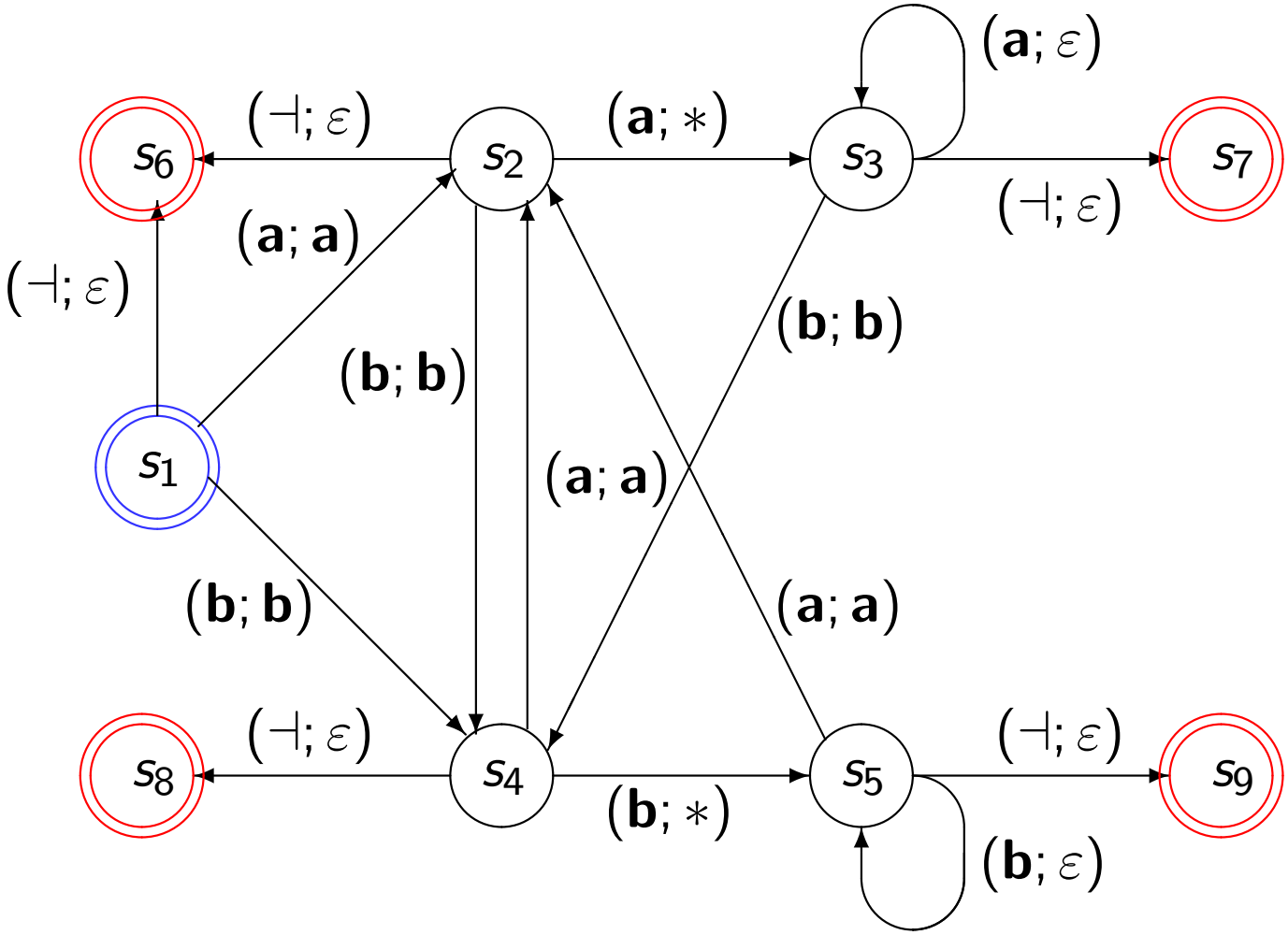
АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Автомат-преобразователь слов в шаблоны



АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

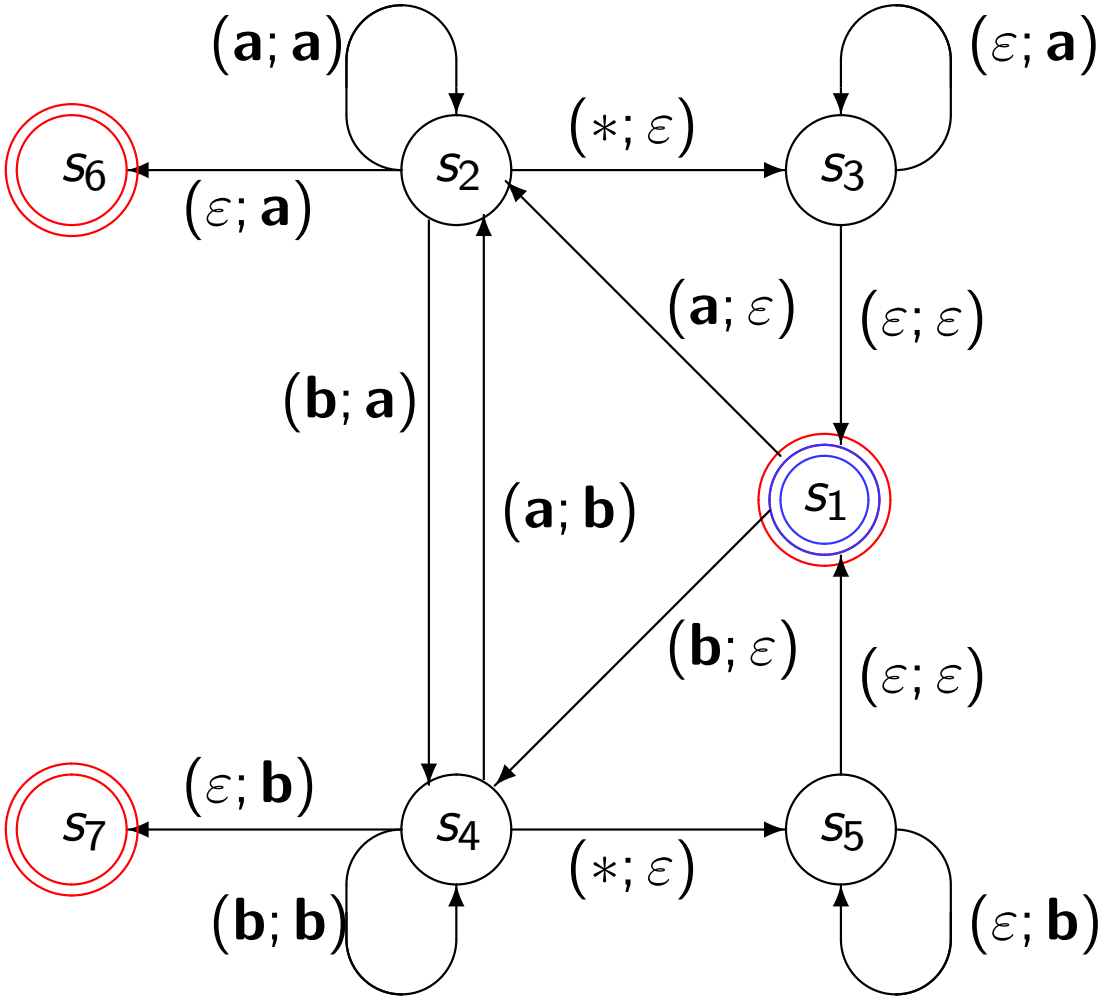
Автомат-преобразователь слов в шаблоны



$aaabbabaaabbba \dashv \Rightarrow \quad a * b * aba * b * a$

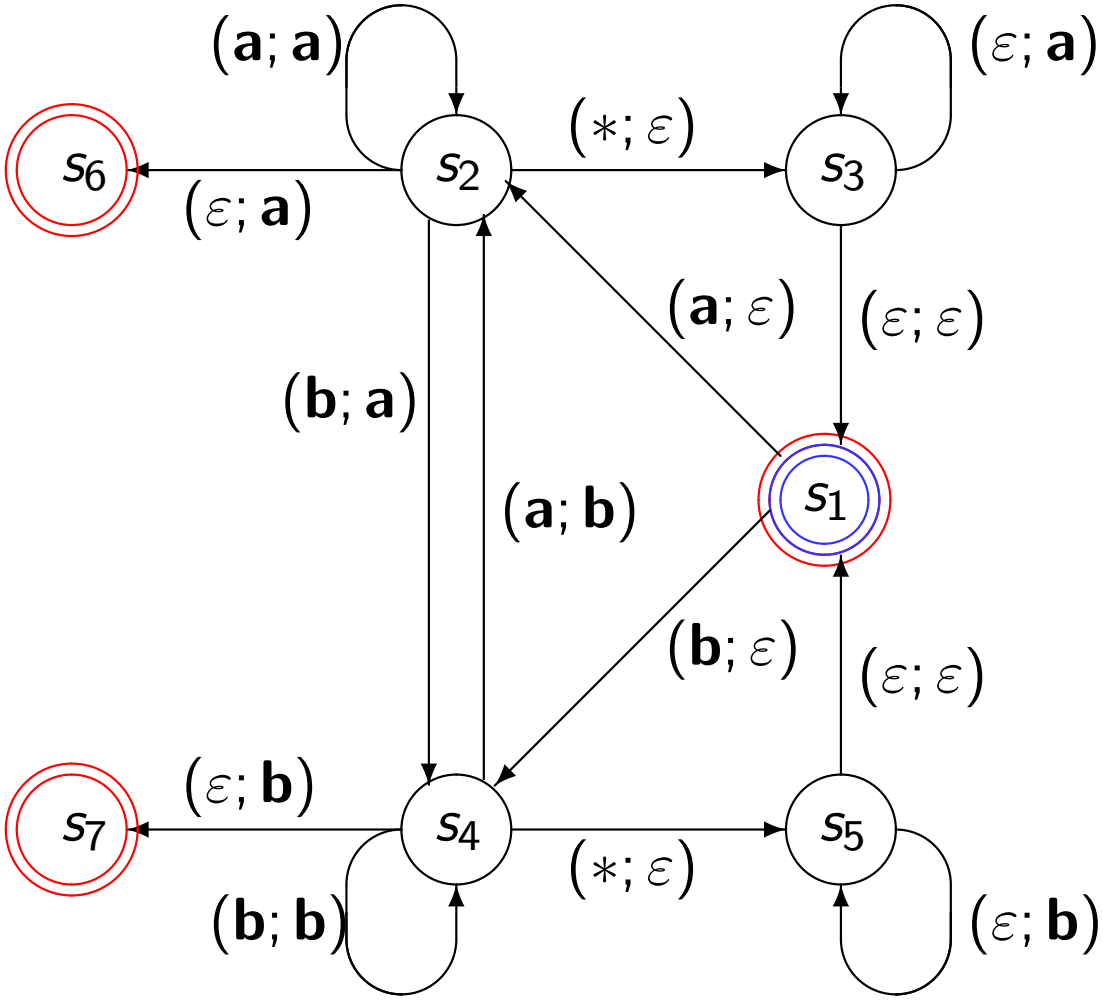
АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Автомат-преобразователь шаблонов в слова



АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Автомат-преобразователь шаблонов в слова



$aa * b * b * aa \Rightarrow aaaabaa$

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Пример. Рассмотрим примеры словарных отношений

- $R_= = \{(w, w) : w \in \Sigma^*\}$
- $R_\neq = \{(w, u) : w, u \in \Sigma^*, u \neq w\}$
- $R_{\times 2} = \{(w, ww) : w \in \Sigma^*\}$
- $R_{rev} = \{(w, w^{-1}) : w \in \Sigma^*\}$

Какие из этих отношений рациональные, а какие — нет?

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Пример. Рассмотрим примеры словарных отношений

- $R_= = \{(w, w) : w \in \Sigma^*\}$ — рациональное отношение;
- $R_\neq = \{(w, u) : w, u \in \Sigma^*, u \neq w\}$ — рациональное отношение;
- $R_{\times 2} = \{(w, ww) : w \in \Sigma^*\}$ — иррациональное отношение;
- $R_{rev} = \{(w, w^{-1}) : w \in \Sigma^*\}$ — иррациональное отношение.

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

Пример. Рассмотрим примеры словарных отношений

- $R_= = \{(w, w) : w \in \Sigma^*\}$ — рациональное отношение;
- $R_\neq = \{(w, u) : w, u \in \Sigma^*, u \neq w\}$ — рациональное отношение;
- $R_{\times 2} = \{(w, ww) : w \in \Sigma^*\}$ — иррациональное отношение;
- $R_{rev} = \{(w, w^{-1}) : w \in \Sigma^*\}$ — иррациональное отношение.

А как в этом убедиться?

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

У нас уже есть опыт и методика.

- Проверить свойства замкнутости класса рациональных отношений.
- Отыскать алгебраический способ описания рациональных отношений и алгоритм трансляции алгебраических описаний в автоматы- преобразователи.
- Обнаружить аналог теоремы о разрастании языков.
- Разработать алгоритмы проверки эквивалентности и минимизации автоматов-преобразователи.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Для всякого отношения $R, R \subseteq \Sigma^* \times \Delta^*$, его областью определения будем называть язык $Dom(R) = \{w : \exists \alpha (w, \alpha) \in R\}$, а множеством значений — язык $Range(R) = \{\alpha : \exists w (w, \alpha) \in R\}$.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Для всякого отношения $R, R \subseteq \Sigma^* \times \Delta^*$, его областью определения будем называть язык $Dom(R) = \{w : \exists \alpha (w, \alpha) \in R\}$, а множеством значений — язык $Range(R) = \{\alpha : \exists w (w, \alpha) \in R\}$.

Утверждение 10.1. Если R — рациональное отношение над алфавитами Σ, Δ , то его область определения $Dom(R)$ и множество значений $Range(R)$ являются регулярными языками.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Для всякого отношения $R, R \subseteq \Sigma^* \times \Delta^*$, его областью определения будем называть язык $Dom(R) = \{w : \exists\alpha(w, \alpha) \in R\}$, а множеством значений — язык $Range(R)=\{\alpha : \exists w(w, \alpha)\in R\}$.

Утверждение 10.1. Если R — рациональное отношение над алфавитами Σ, Δ , то его область определения $Dom(R)$ и множество значений $Range(R)$ являются регулярными языками.

Доказательство. Автомат-преобразователь, реализующий отношение R , легко преобразуется в конечные автоматы, распознающие языки $Dom(R)$ и $Range(R)$. QED

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Утверждение 10.2. Если R — рациональное отношение над алфавитами Σ, Δ , то язык $L_R = \{w\alpha^{-1} : (w, \alpha) \in R\}$ является КС-языком в алфавите $\Sigma \cup \Delta$.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Утверждение 10.2. Если R — рациональное отношение над алфавитами Σ, Δ , то язык $L_R = \{w\alpha^{-1} : (w, \alpha) \in R\}$ является КС-языком в алфавите $\Sigma \cup \Delta$.

Доказательство. Автомат-преобразователь, реализующий отношение R , легко преобразуется в магазинный автомат, распознающий слова языка L_R : на каждом переходе $s' \xrightarrow{(x,\beta)} s''$ он записывает слово β^{-1} в магазин, и после того, как будет прочтено слово w , продолжает распознавание слова α^{-1} , используя накопленное содержимое магазина. QED

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Таким образом,

отношение $R_{\times 2} = \{(w, ww) : w \in \Sigma^*\}$ не является рациональным, поскольку $Range(R) = \{ww : w \in \Sigma^*\}$ не является регулярным языком

отношение $R_{rev} = \{(w, w^{-1}) : w \in \Sigma^*\}$ не является рациональным, поскольку $L_{R_{rev}} = \{ww : w \in \Sigma^*\}$ не является КС-языком.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Теорема о накачке тоже имеет место.

Утверждение 10.3. Пусть R — рациональное отношение над алфавитами Σ, Δ . Тогда существует такое положительное целое число N , что для любой пары слов (w, α), $(w, \alpha) \in R$, суммарной длины не менее N существуют такие разбиения этих слов $w = xyz, \alpha = \eta\beta\theta$, где $|xy| \leq N, |\eta\beta| \leq N$ и $y\beta \neq \varepsilon$, для которых включение $(xy^iz, \eta\beta^i\theta) \in R$ верно для всех $i, i \geq 0$.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Теорема о накачке тоже имеет место.

Утверждение 10.3. Пусть R — рациональное отношение над алфавитами Σ, Δ . Тогда существует такое положительное целое число N , что для любой пары слов (w, α), $(w, \alpha) \in R$, суммарной длины не менее N существуют такие разбиения этих слов $w = xyz, \alpha = \eta\beta\theta$, где $|xy| \leq N, |\eta\beta| \leq N$ и $y\beta \neq \varepsilon$, для которых включение $(xy^iz, \eta\beta^i\theta) \in R$ верно для всех $i, i \geq 0$.

Доказательство. Совершенно аналогично доказательству теоремы о разрастании для автоматных языков

QED

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Для словарных отношений можно определять разнообразные операции.

Конкатенацией отношений P, Q над алфавитами Σ, Δ называется такое словарное отношение $R = PQ$ над теми же алфавитами, которое определяется соотношением

$$R = \{(w, \alpha) : \ w = w_1 w_2, \alpha = \alpha_1 \alpha_2, \\ (w_1, \alpha_1) \in P, (w_2, \alpha_2) \in Q\}.$$

Итерация отношения P определяется так:

$$P^* = \bigcup_{k=0}^{\infty} P^k \ .$$

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Композицией отношений P, Q над алфавитами Σ, Δ и Δ, Γ соответственно называется такое словарное отношение $R = P \circ Q$ над алфавитами Σ, Γ , которое определяется соотношением

$$R = \{(w, u) : \exists \alpha((w, \alpha) \in P \ \wedge \ (\alpha, u) \in Q)\}.$$

Объединение, пересечение и дополнение словарных отношений вводятся обычным образом.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Утверждение 10.4. Класс рациональных отношений замкнут относительно операций объединения, конкатенации, итерации и композиции.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Утверждение 10.4. Класс рациональных отношений замкнут относительно операций объединения, конкатенации, итерации и композиции.

Доказательство. Совершенно аналогично доказательству теоремы о замкнутости класса автоматных языков. Замкнутость относительно операции композиции доказать самостоятельно . QED

Рациональные отношения и их свойства

А есть ли замкнутость рациональных отношений относительно пересечения?

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

А есть ли замкнутость рациональных отношений относительно пересечения? Увы, ее нет!

Утверждение 10.5. Класс рациональных отношений не замкнут относительно операций пересечения и дополнения.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

А есть ли замкнутость рациональных отношений относительно пересечения? Увы, ее нет!

Утверждение 10.5. Класс рациональных отношений не замкнут относительно операций пересечения и дополнения.

Доказательство. Рассмотрим словарные отношения $P = \{(a^n b^m, c^n) : m, n \geq 0\}$ и $Q = \{(a^m b^n, c^n) : m, n \geq 0\}$. Нетрудно построить автоматы-распознаватели, реализующие эти отношения. Однако их пересечение $P \cap Q = \{(a^n b^n, c^n) : n \geq 0\}$ очевидно не является рациональным отношением. QED

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Рациональные отношения можно задавать при помощи регулярных выражений:

- константами объявляются все пары $(x, \varepsilon), x \in \Sigma \cup \{\varepsilon\}$, и $(\varepsilon, y), y \in \Delta \cup \{\varepsilon\}$;
- если P_1 и P_2 — регулярные выражения, то $(P_1 + P_2)$, $(P_1 \cdot P_2)$ и (P_1^*) — также регулярные выражения.

Значение $R(P)$ регулярного выражения P определяется на основании операций объединения, конкатенации и итерации для словарных отношений так же, как и для языков.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Для введенных таким образом регулярных выражений справедлив аналог теоремы Клини.

Утверждение 10.6. Словарное отношение является регулярным тогда и только тогда, когда оно является рациональным.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Для введенных таким образом регулярных выражений справедлив аналог теоремы Клини.

Утверждение 10.6. Словарное отношение является регулярным тогда и только тогда, когда оно является рациональным.

Доказательство. Проводится аналогично доказательству теоремы Клини для конечных автоматов-распознавателей. QED

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Пример. Словарное отношение $R_{\neq} = \{(w, u) : w, u \in \Sigma^*, u \neq w\}$ является рациональным.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Пример. Словарное отношение $R_{\neq} = \{(w, u) : w, u \in \Sigma^*, u \neq w\}$ является рациональным.

Доказательство. Для любой пары букв x, y будем использовать запись (x, y) для обозначения регулярного выражения $(x, \varepsilon) \cdot (\varepsilon, y)$.
Тогда $R_{\neq} = R(F_1 + F_2 + F_3)$, где

- $F_1 = (\sum\limits_{x \in \Sigma}(x, x))^* \cdot (\sum\limits_{x \neq y}(x, y)) \cdot (\sum\limits_{x, y \in \Sigma}(x, y))^*$,
- $F_2 = (\sum\limits_{x, y \in \Sigma}(x, y))^* \cdot (\sum\limits_{x \in \Sigma}(x, \varepsilon)) \cdot (\sum\limits_{x \in \Sigma}(x, \varepsilon))^*$,
- $F_3 = (\sum\limits_{x, y \in \Sigma}(x, y))^* \cdot (\sum\limits_{y \in \Sigma}(\varepsilon, y)) \cdot (\sum\limits_{y \in \Sigma}(\varepsilon, y))^*$.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Задача 1. Разработать алгоритм, который для автоматов-распознавателей, реализующих отношения R_1 и R_2 , строит автомат, реализующий их композицию $R_1 \circ R_2$. Какова оценка размера этого автомата?

Задача 2. Верно ли, что всякое словарное отношение P , которое удовлетворяет условиям

1. $Dom(P)$ и $Range(P)$ — регулярные языки,
2. $L = \{wu^{-1} : (w, u) \in P\}$ — КС-язык

является рациональным.

РАЦИОНАЛЬНЫЕ ОТНОШЕНИЯ И ИХ СВОЙСТВА

Задача 3. Верно ли, что для любого рационального отношения R , словарное отношение $R^{-1} = \{(u, w) : (w, u) \in R\}$ также является рациональным?

Задача 4. Верно ли, что для любого рационального отношения R , словарное отношение $R^{rev} = \{(w^{-1}, u^{-1}) : (w, u) \in R\}$ также является рациональным?

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

А как обстоит дело с алгоритмическими проблемами?

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

А как обстоит дело с алгоритмическими проблемами?

Очевидно, что проблема пустоты для рациональных отношений $R(\mathcal{A}) = \varnothing$? разрешима. Почему?

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

А как обстоит дело с алгоритмическими проблемами?

Очевидно, что проблема пустоты для рациональных отношений $R(\mathcal{A}) = \varnothing$? разрешима. Почему?

Иначе обстоит дело с другими задачами.

Утверждение 10.7. Проблема тотальности для рациональных отношений $R(\mathcal{A}) = \Sigma^* \times \Delta^*$? алгоритмически неразрешима.

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Доказательство. Покажем, что Проблема соответствий Поста (ПСП) сводима к проблеме тотальности рациональных отношений.

Пусть задана система пар слов $\mathcal{P} = \{(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k)\}$ в алфавите Δ . Для этой системы определим два отношения над алфавитами $\Sigma = \{1, 2, \ldots, k\}$ и Δ :

$$\begin{aligned} R^1_{\mathcal{P}} &= \{(i_1 i_2 \ldots i_n, u_{i_1} u_{i_2} \ldots u_{i_n}) : n \geq 1\}, \\ R^2_{\mathcal{P}} &= \{(j_1 j_2 \ldots j_m, v_{j_1} v_{j_2} \ldots v_{j_m}) : m \geq 1\}. \end{aligned}$$

Нетрудно видеть, что оба эти отношения — рациональные.

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Значит, рациональным является и отношение

$$R_{\mathcal{P}} = (R^1_{\mathcal{P}} \circ R_{\neq}) + (R^2_{\mathcal{P}} \circ R_{\neq}).$$

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Значит, рациональным является и отношение

$$R_{\mathcal{P}} = (R^1_{\mathcal{P}} \circ R_{\neq}) + (R^2_{\mathcal{P}} \circ R_{\neq}).$$

Но тогда ПСП для $\mathcal{P}$ имеет решение $i_1, i_2, \ldots, i_n$

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Значит, рациональным является и отношение

$$R_{\mathcal{P}} = (R^1_{\mathcal{P}} \circ R_{\neq}) + (R^2_{\mathcal{P}} \circ R_{\neq}).$$

Но тогда ПСП для $\mathcal{P}$ имеет решение $i_1, i_2, \ldots, i_n$

$$\Longleftrightarrow$$

$$u_{i_1} u_{i_2} \ldots u_{i_n} = v_{i_1} v_{i_1} \ldots v_{i_n} = w$$

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Значит, рациональным является и отношение

$$R_{\mathcal{P}} = (R^1_{\mathcal{P}} \circ R_{\neq}) + (R^2_{\mathcal{P}} \circ R_{\neq}).$$

Но тогда ПСП для $\mathcal{P}$ имеет решение $i_1, i_2, \ldots, i_n$

$$\Longleftrightarrow$$

$$u_{i_1} u_{i_2} \ldots u_{i_n} = v_{i_1} v_{i_1} \ldots v_{i_n} = w$$

$$\Longleftrightarrow$$

$$(i_1 i_2 \ldots i_n, w) \notin R^1_{\mathcal{P}} \circ R_{\neq},\ (i_1 i_2 \ldots i_n, w) \notin R^2_{\mathcal{P}} \circ R_{\neq}$$

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Значит, рациональным является и отношение

$$R_{\mathcal{P}} = (R^1_{\mathcal{P}} \circ R_{\neq}) + (R^2_{\mathcal{P}} \circ R_{\neq}).$$

Но тогда ПСП для $\mathcal{P}$ имеет решение $i_1, i_2, \ldots, i_n$

$$\Longleftrightarrow$$

$$u_{i_1} u_{i_2} \ldots u_{i_n} = v_{i_1} v_{i_1} \ldots v_{i_n} = w$$

$$\Longleftrightarrow$$

$$(i_1 i_2 \ldots i_n, w) \notin R^1_{\mathcal{P}} \circ R_{\neq},\ (i_1 i_2 \ldots i_n, w) \notin R^2_{\mathcal{P}} \circ R_{\neq}$$

$$\Longleftrightarrow$$

$$R_{\mathcal{P}} \neq \Sigma^* \times \Delta^* .$$

QED

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Следствие. Проблема эквивалентности для конечных автоматов-преобразователей алгоритмически неразрешима.

Это обстоятельство сильно затрудняет оптимизацию автоматов-преобразователей.

Задача 5. Является ли алгоритмически разрешимой проблема включения $(w, u) \in R(\mathcal{A})$ для конечных автоматов-преобразователей?

Задача 6. Является ли алгоритмически разрешимой проблема пустоты пересечения $R(\mathcal{A}) \cap R(\mathcal{B}) = \varnothing$ для конечных автоматов-преобразователей?

АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ

Следствие. Проблема эквивалентности для конечных автоматов-преобразователей алгоритмически неразрешима.

Это обстоятельство сильно затрудняет оптимизацию автоматов-преобразователей.

Задача 5. Является ли алгоритмически разрешимой проблема включения $(w, u) \in R(\mathcal{A})$ для конечных автоматов-преобразователей?

Задача 6. Является ли алгоритмически разрешимой проблема пустоты пересечения $R(\mathcal{A}) \cap R(\mathcal{B}) = \varnothing$ для конечных автоматов-преобразователей?

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

А как обстоит дело с алгоритмическими проблемами для **детерминированных** автоматов-преобразователей?

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

А как обстоит дело с алгоритмическими проблемами для **детерминированных** автоматов-преобразователей?

Конечный автомат-преобразователь $\mathcal{A} = (\Sigma, \Delta, S, I, F, T)$ называется детерминированным , если он удовлетворяет следующим требованиям:

- $|I| \leq 1$, т.е. имеет не более одного начального состояния;
- отношение переходов T является функцией $T : S \times \Sigma \to S \times \Delta^*$.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

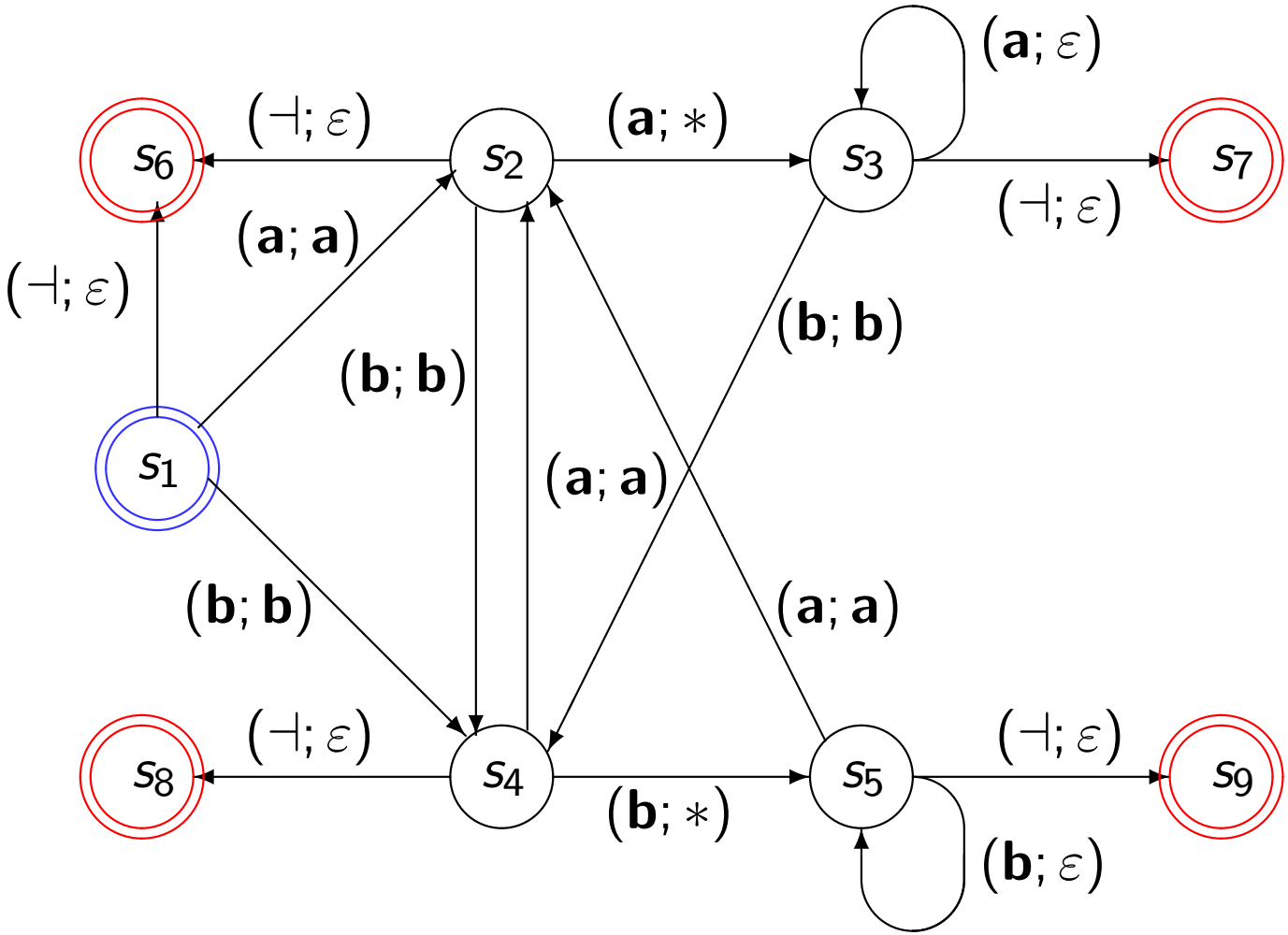
Детерминированный автомат должен знать, где оканчивается транслируемое слово. Поэтому к входному алфавиту Σ прилагается специальный маркер конца слова $\dashv$.

Детерминированный автомат-преобразователь $\mathcal{A} = (\Sigma, \Delta, S, I, F, T)$ реализует частичную функцию $R_{\mathcal{A}} : \Sigma^* \to \Delta^*$, значение которой для каждого входного слова w определяется соотношением

$$R_{\mathcal{A}}(w) = \alpha \quad \Leftrightarrow \quad s_0 \xrightarrow{(w\dashv,\alpha)}{}_* s_n, s_0 \in I, s_n \in F.$$

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Детерминированный автомат-преобразователь



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Теорема 10.8. Проблема эквивалентности $R_{\mathcal{A}_1} = R_{\mathcal{A}_2}$? для детерминированных конечных автоматов-преобразователей разрешима.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Теорема 10.8. Проблема эквивалентности $R_{\mathcal{A}_1} = R_{\mathcal{A}_2}$? для детерминированных конечных автоматов-преобразователей разрешима.

Доказательство. Рассмотрим автоматы-преобразователи $\mathcal{A}_1, \mathcal{A}_2$, удовлетворяющие условиям

1. $Dom(R_{\mathcal{A}_1}) = Dom(R_{\mathcal{A}_2})$
   (иначе $\mathcal{A}_1$ и $\mathcal{A}_2$ очевидно неэквивалентны);
2. $\mathcal{A}_1$ и $\mathcal{A}_2$ не имеют бесполезных состояний
   (эти состояния можно обнаружить и удалить).

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Определим на множестве $\Delta^* \times \Delta^*$ функцию

$$Dif(\alpha_1, \alpha_2) = \begin{cases} [\beta, \varepsilon], \text{если } \alpha_1 = \alpha_2\beta, \\ [\varepsilon, \beta], \text{если } \alpha_2 = \alpha_1\beta, \\ \mathit{fail}, \text{в остальных случаях.} \end{cases}$$

Эта функция «вычитает» из одной строки другую как префикс и сообщает о неудаче, если ни одна из строк α_1, α_2 не является префиксом другой.

Например,
$Dif(abbba, ab) = (bba, \varepsilon)$, $Dif(abbba, ba) = \text{fail}$.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Из определения функции Dif следуют

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Из определения функции *Dif* следуют

Свойство 1. Для любой пары строк α_1, α_2

$$Dif(\alpha_1, \alpha_2) = [\varepsilon, \varepsilon] \quad \Leftrightarrow \quad \alpha_1 = \alpha_2.$$

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Из определения функции Dif следуют

Свойство 1. Для любой пары строк α_1, α_2

$$Dif(\alpha_1, \alpha_2) = [\varepsilon, \varepsilon] \quad \Leftrightarrow \quad \alpha_1 = \alpha_2.$$

Свойство 2. Для любых пар строк α_1', α_2' и α_1'', α_2''

$$Dif(\alpha_1', \alpha_2') = [\beta_1, \beta_2] \Rightarrow Dif(\alpha_1'\alpha_1'', \alpha_2'\alpha_2'') = Dif(\beta_1\alpha_1'', \beta_2\alpha_2'')$$

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Из определения функции Dif следуют

Свойство 1. Для любой пары строк α_1, α_2

$$Dif(\alpha_1, \alpha_2) = [\varepsilon, \varepsilon] \quad \Leftrightarrow \quad \alpha_1 = \alpha_2.$$

Свойство 2. Для любых пар строк α_1', α_2' и α_1'', α_2''

$$Dif(\alpha_1', \alpha_2') = [\beta_1, \beta_2] \Rightarrow Dif(\alpha_1'\alpha_1'', \alpha_2'\alpha_2'') = Dif(\beta_1\alpha_1'', \beta_2\alpha_2'')$$

Свойство 3. Если $Dif(\alpha_1, \alpha_2) =$ fail, то для любой пары строк β_1, β_2 верно $\alpha_1\beta_1 \neq \alpha_2\beta_2$.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Эквивалентность преобразователей $\mathcal{A}_1$ и $\mathcal{A}_2$ будем проверять так же, как и эквивалентность конечных автоматов — путем построения декартова произведения этих преобразователей $\mathcal{A}_1 \times \mathcal{A}_2$.

Но для автоматов-преобразователей это декартово произведение будем определять иначе.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Пусть $\mathcal{A}_i = (\Sigma, \Delta, S_i, \{s_0^i\}, F_i, T_i), i = 1, 2$. Тогда $\mathcal{A}_1 \times \mathcal{A}_2$ — это детерминированный автомат-распознаватель с **бесконечным** числом состояний $\mathcal{A}_1 \times \mathcal{A}_2 = (\Sigma, Q, q_0, H, \varphi)$, где

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Пусть $\mathcal{A}_i = (\Sigma, \Delta, S_i, \{s_0^i\}, F_i, T_i), i = 1, 2$. Тогда $\mathcal{A}_1 \times \mathcal{A}_2$ — это детерминированный автомат-распознаватель с **бесконечным** числом состояний $\mathcal{A}_1 \times \mathcal{A}_2 = (\Sigma, Q, q_0, H, \varphi)$, где

- множество состояний:
  $Q = S_1 \times S_2 \times ((\Delta^* \times \Delta^*) \cup \{\text{ fail}\})$,

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Пусть $\mathcal{A}_i = (\Sigma, \Delta, S_i, \{s_0^i\}, F_i, T_i), i = 1, 2$. Тогда $\mathcal{A}_1 \times \mathcal{A}_2$ — это детерминированный автомат-распознаватель с **бесконечным** числом состояний $\mathcal{A}_1 \times \mathcal{A}_2 = (\Sigma, Q, q_0, H, \varphi)$, где

- множество состояний:
  $Q = S_1 \times S_2 \times ((\Delta^* \times \Delta^*) \cup \{\text{ fail}\})$,
- подмножество начальных состояний:
  $q_0 = (s_0^1, s_0^2, [\varepsilon, \varepsilon])$,

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Пусть $\mathcal{A}_i = (\Sigma, \Delta, S_i, \{s_0^i\}, F_i, T_i), i = 1, 2$. Тогда $\mathcal{A}_1 \times \mathcal{A}_2$ — это детерминированный автомат-распознаватель с **бесконечным** числом состояний $\mathcal{A}_1 \times \mathcal{A}_2 = (\Sigma, Q, q_0, H, \varphi)$, где

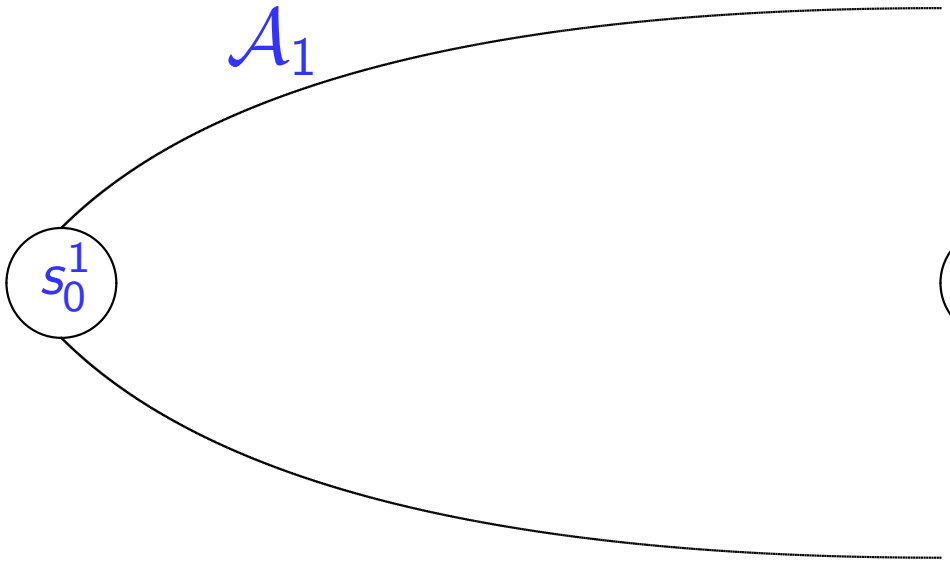
- множество состояний:
  $Q = S_1 \times S_2 \times ((\Delta^* \times \Delta^*) \cup \{\text{ fail}\})$,
- подмножество начальных состояний:
  $q_0 = (s_0^1, s_0^2, [\varepsilon, \varepsilon])$,
- функция переходов:
  если $T_1(s_1', a) = (\beta_1, s_1'')$ и $T_2(s_2', a) = (\beta_2, s_2'')$, то
  $\varphi((s_1', s_2', [\alpha_1, \alpha_2]), a) = (s_1'', s_2'', Dif(\alpha_1\beta_1, \alpha_2\beta_2))$,

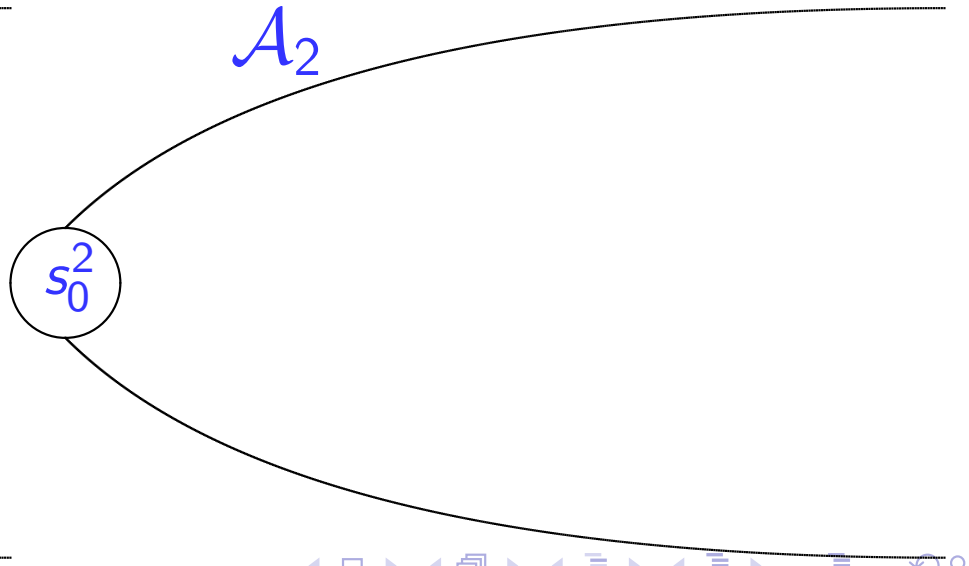
ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Пусть $\mathcal{A}_i = (\Sigma, \Delta, S_i, \{s_0^i\}, F_i, T_i), i = 1, 2$. Тогда $\mathcal{A}_1 \times \mathcal{A}_2$ — это детерминированный автомат-распознаватель с **бесконечным** числом состояний $\mathcal{A}_1 \times \mathcal{A}_2 = (\Sigma, Q, q_0, H, \varphi)$, где

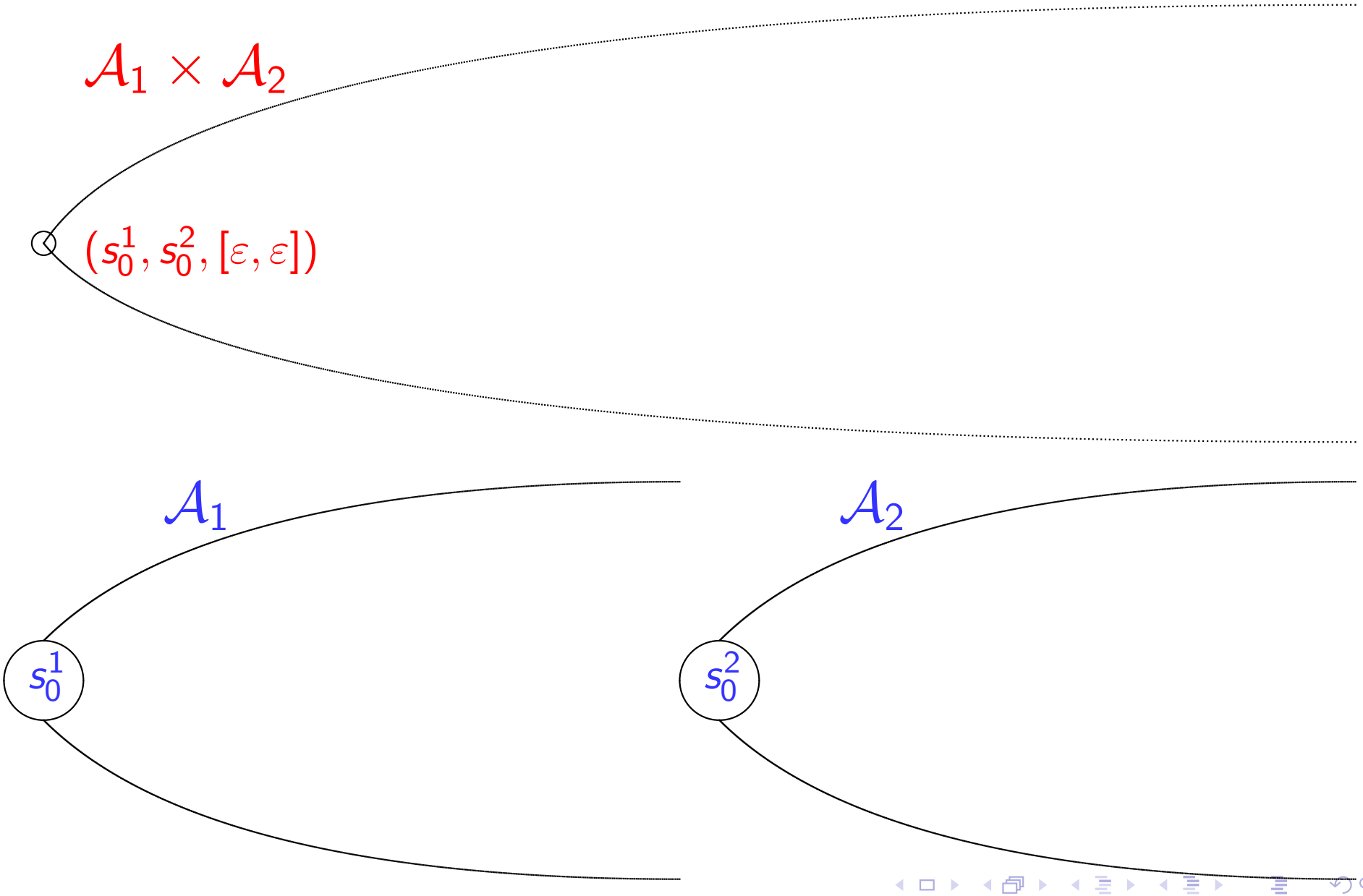
- множество состояний:
  $Q = S_1 \times S_2 \times ((\Delta^* \times \Delta^*) \cup \{\text{ fail}\})$,
- подмножество начальных состояний:
  $q_0 = (s_0^1, s_0^2, [\varepsilon, \varepsilon])$,
- функция переходов:
  если $T_1(s_1', a) = (\beta_1, s_1'')$ и $T_2(s_2', a) = (\beta_2, s_2'')$, то
  $\varphi((s_1', s_2', [\alpha_1, \alpha_2]), a) = (s_1'', s_2'', Dif(\alpha_1\beta_1, \alpha_2\beta_2))$,
- подмножество допускающих состояний:
  $H = H_1 \cup H_2$, где
  $H_1 = \{(s', s'', [\alpha_1, \alpha_2]) : s' \in F_1, s'' \in F_2, [\alpha_1, \alpha_2] \neq [\varepsilon, \varepsilon]\}$,
  $H_2 = \{(s', s'', \text{ fail}) : s' \in S_1, s'' \in S_2\}$.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

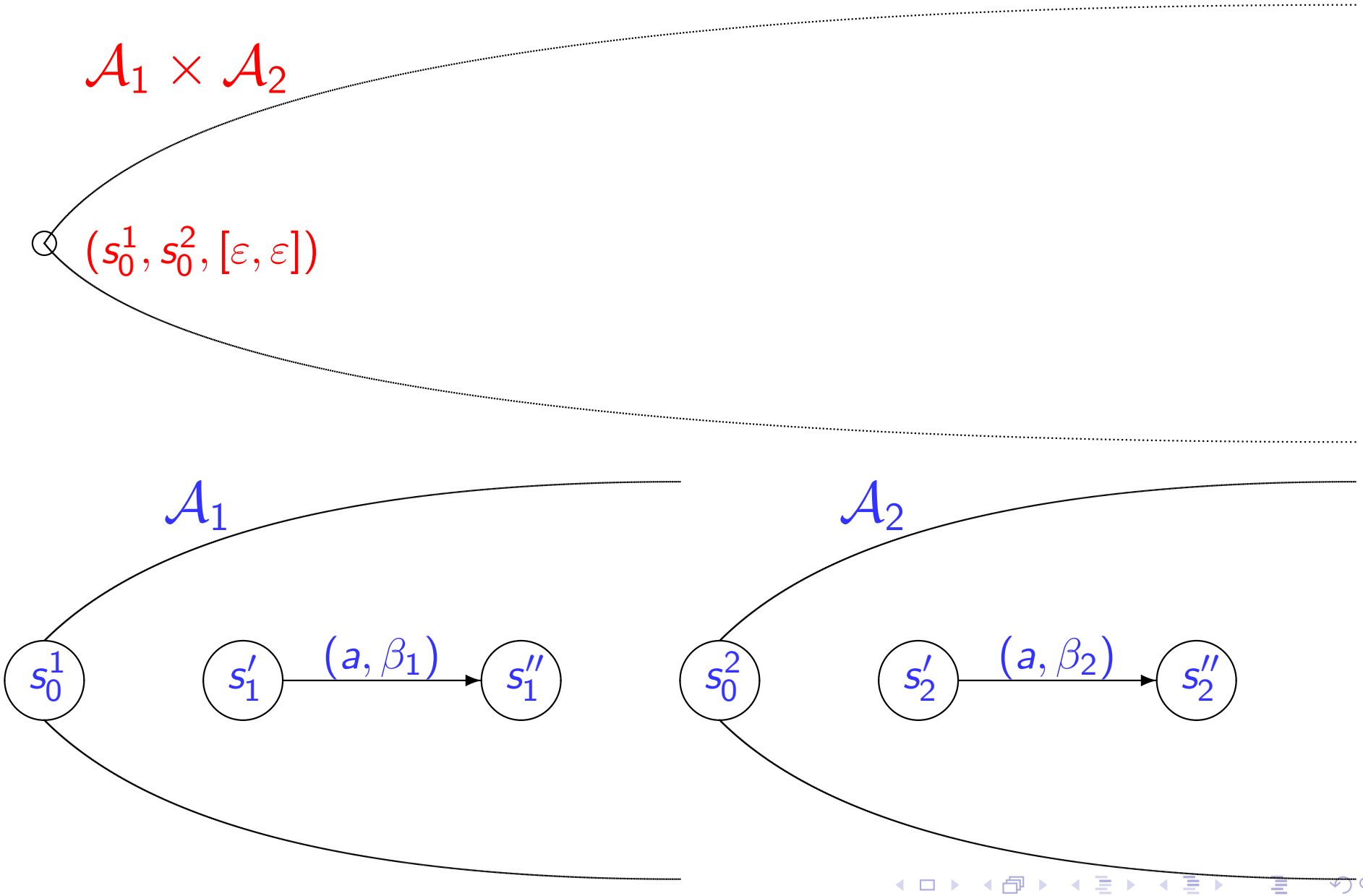




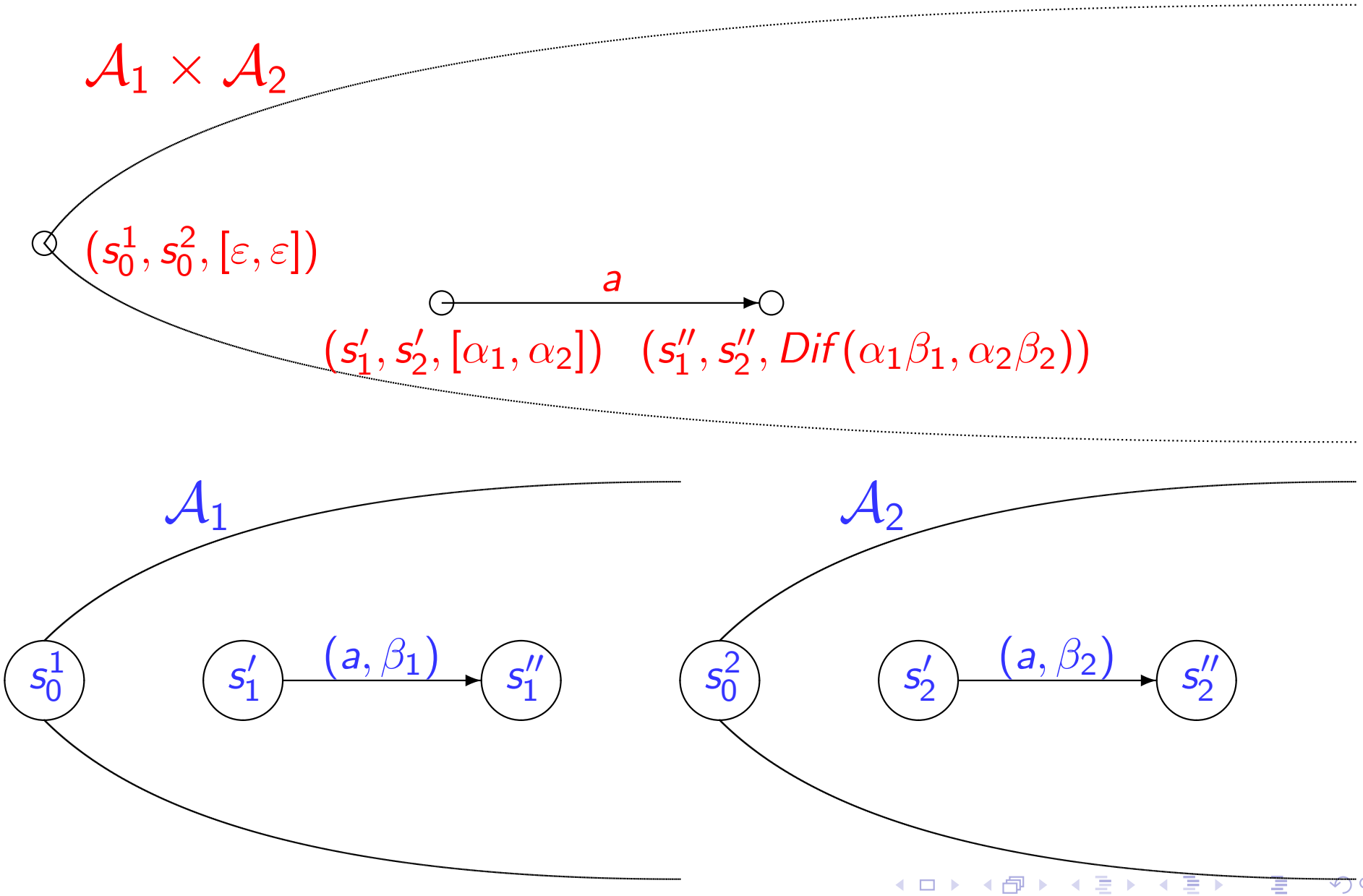
ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



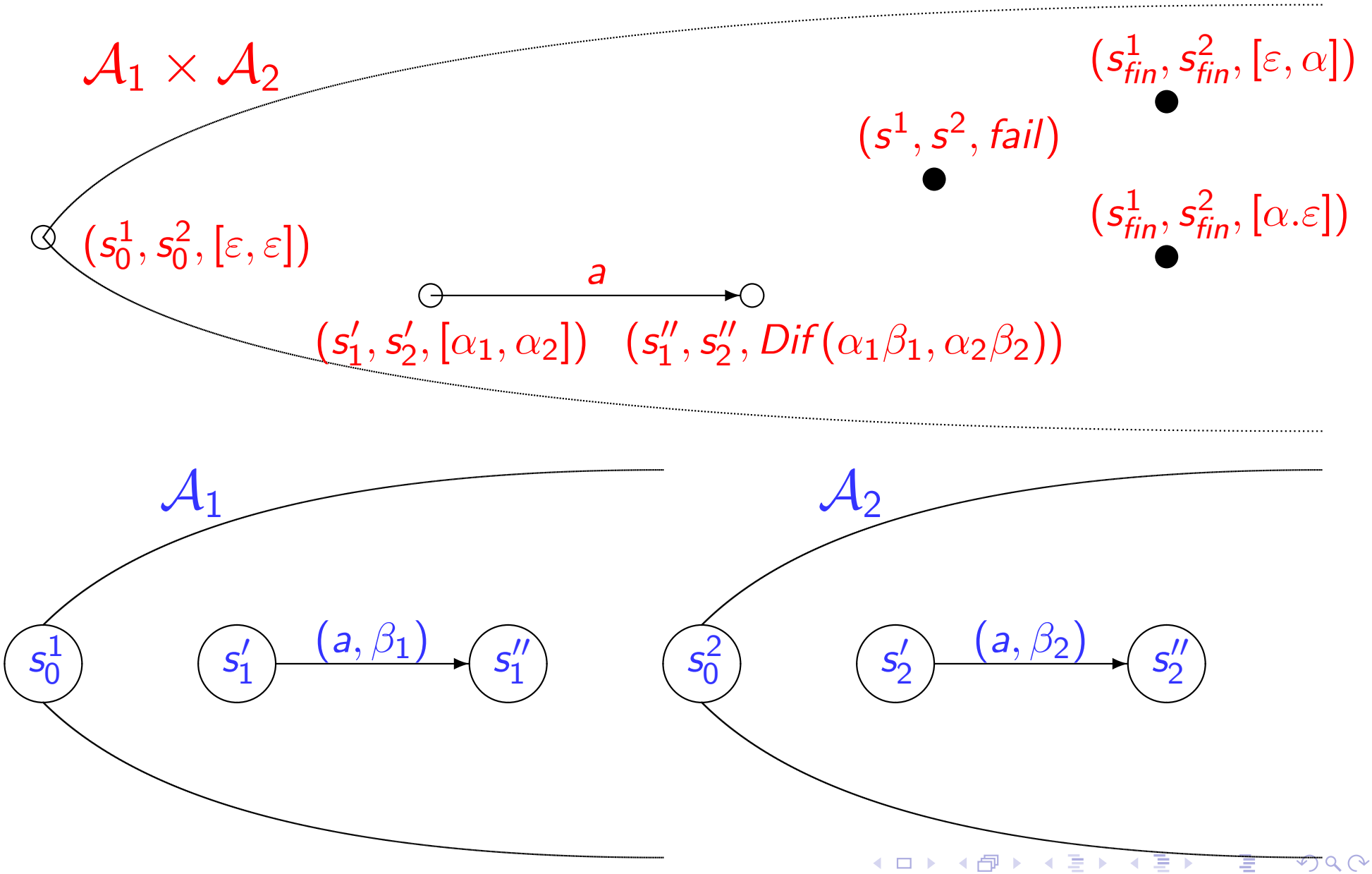
ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Автомат $\mathcal{A}_1 \times \mathcal{A}_2$ моделирует синхронную работу автоматов-преобразователей $\mathcal{A}_1$ и $\mathcal{A}_1$.

Лемма 1. Для любого слова $w, w \in \Sigma^*$, верно

1. $(s_0^1, s_0^2, [\varepsilon, \varepsilon]) \xrightarrow{w}_* (s', s'', [\beta_1, \beta_2])$ тогда и только тогда, когда $s_0^1 \xrightarrow{w,\alpha_1}_* s'$, $s_0^2 \xrightarrow{w,\alpha_2}_* s''$ и $Dif(\alpha_1, \alpha_2) = [\beta_1, \beta_2]$;

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Автомат $\mathcal{A}_1 \times \mathcal{A}_2$ моделирует синхронную работу автоматов-преобразователей $\mathcal{A}_1$ и $\mathcal{A}_1$.

Лемма 1. Для любого слова w, $w \in \Sigma^*$, верно

1. $(s_0^1, s_0^2, [\varepsilon, \varepsilon]) \xrightarrow{w}_* (s', s'', [\beta_1, \beta_2])$ тогда и только тогда, когда $s_0^1 \xrightarrow{w,\alpha_1}_* s'$, $s_0^2 \xrightarrow{w,\alpha_2}_* s''$ и $Dif(\alpha_1, \alpha_2) = [\beta_1, \beta_2]$;
2. если $(s_0^1, s_0^2, [\varepsilon, \varepsilon]) \xrightarrow{w}_* (s', s''$,fail), то $s_0^1 \xrightarrow{w,\alpha_1}_* s'$, $s_0^2 \xrightarrow{w,\alpha_2}_* s''$ и $Dif(\alpha_1, \alpha_2) =$fail;

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

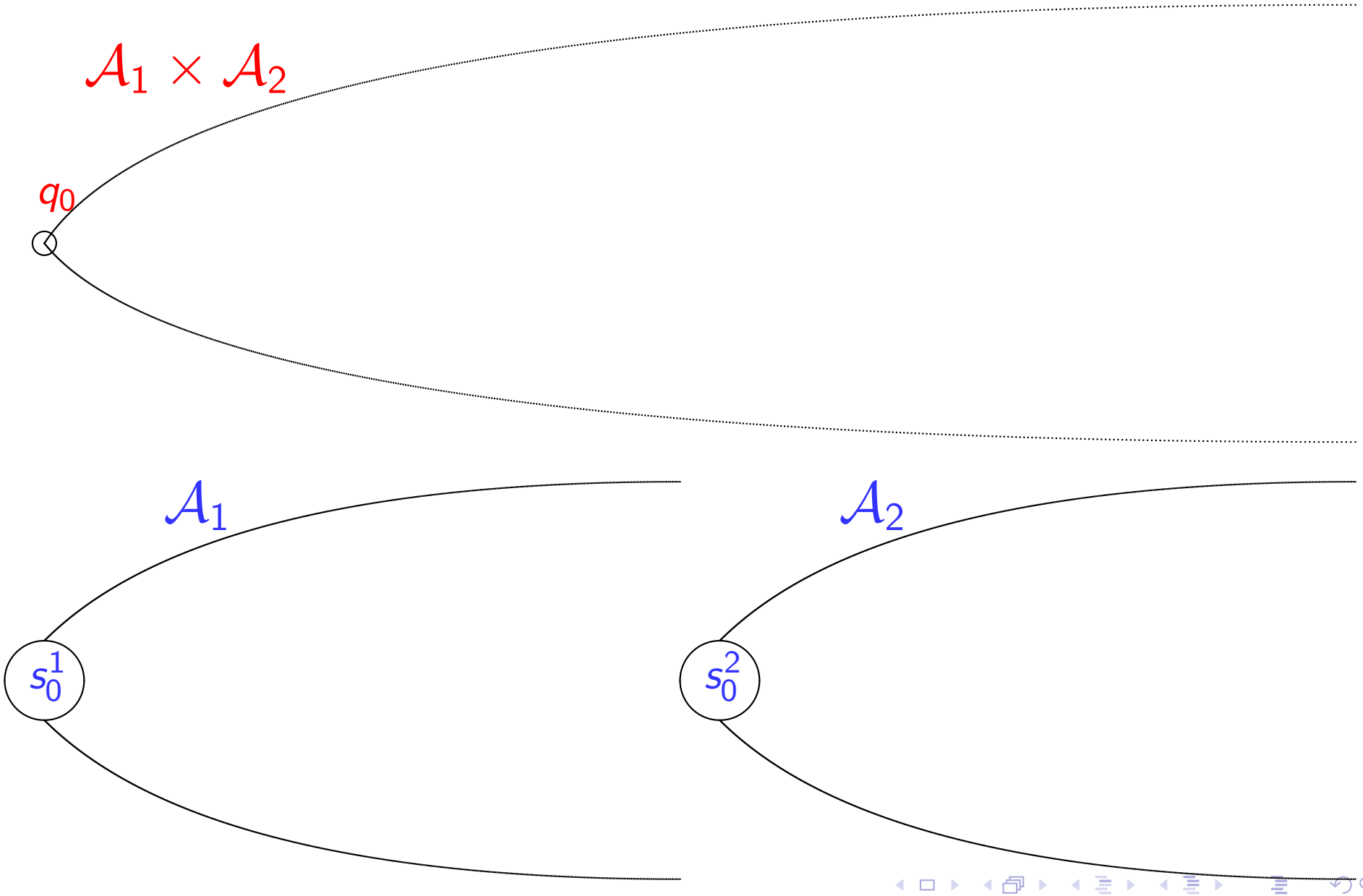
Автомат $\mathcal{A}_1 \times \mathcal{A}_2$ моделирует синхронную работу автоматов-преобразователей $\mathcal{A}_1$ и $\mathcal{A}_1$.

Лемма 1. Для любого слова w, $w \in \Sigma^*$, верно

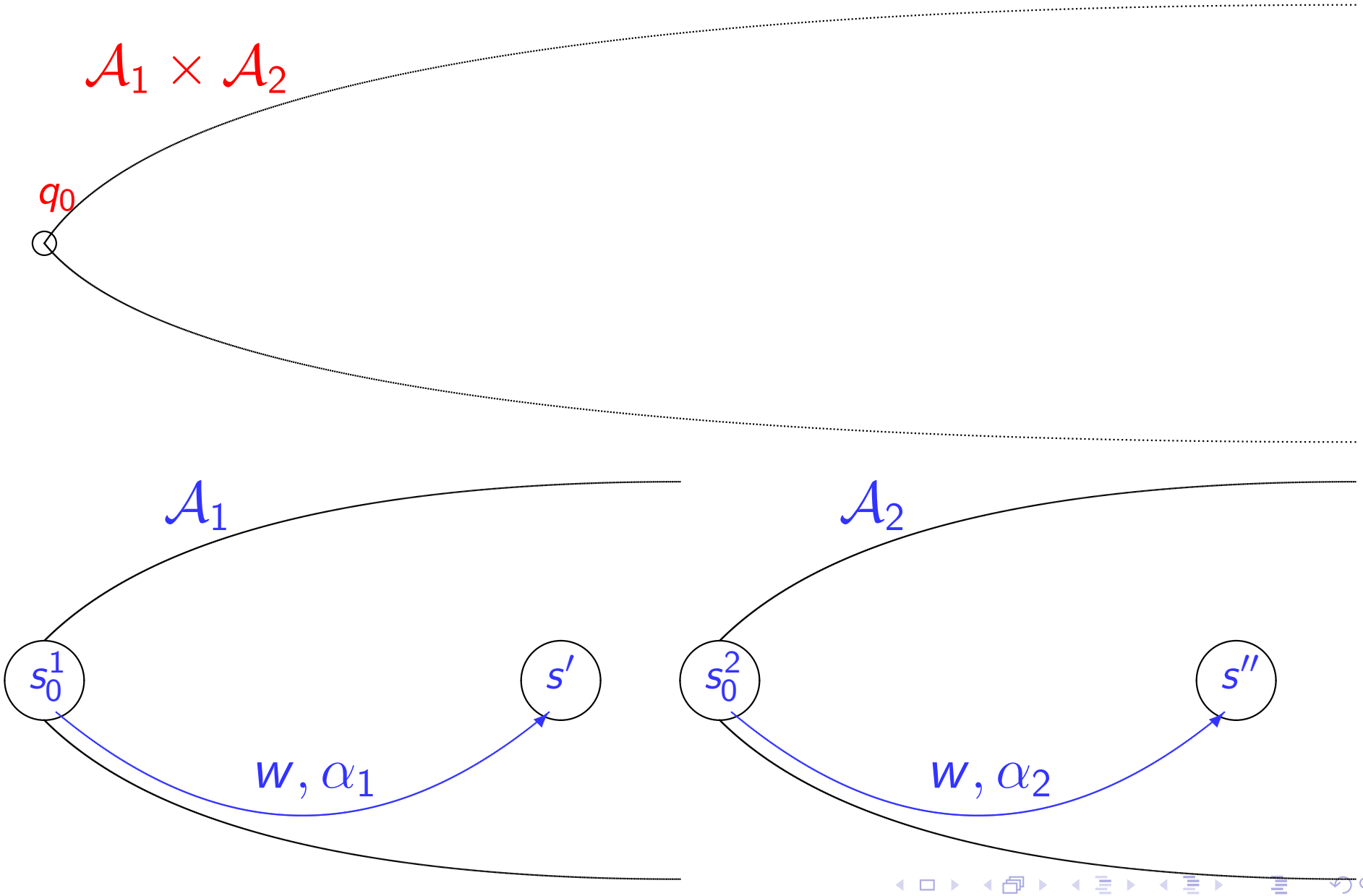
1. $(s_0^1, s_0^2, [\varepsilon, \varepsilon]) \xrightarrow{w}_* (s', s'', [\beta_1, \beta_2])$ тогда и только тогда, когда $s_0^1 \xrightarrow{w,\alpha_1}_* s'$, $s_0^2 \xrightarrow{w,\alpha_2}_* s''$ и $Dif(\alpha_1, \alpha_2) = [\beta_1, \beta_2]$;
2. если $(s_0^1, s_0^2, [\varepsilon, \varepsilon]) \xrightarrow{w}_* (s', s''$,fail$)$, то $s_0^1 \xrightarrow{w,\alpha_1}_* s'$, $s_0^2 \xrightarrow{w,\alpha_2}_* s''$ и $Dif(\alpha_1, \alpha_2) =$fail;

Доказательство. Индукцией по длине слова w с использованием Свойств 2 и 3.

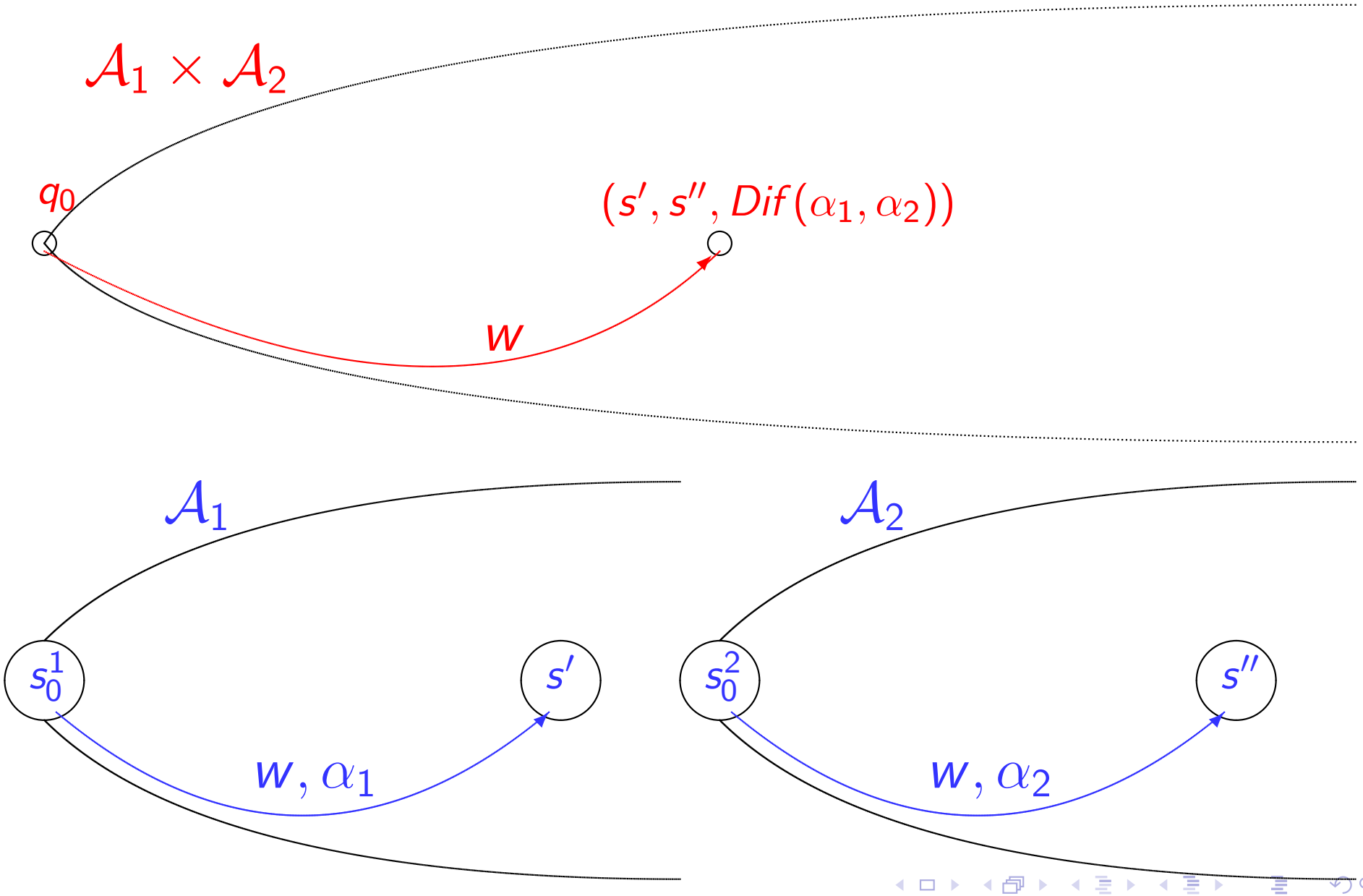
ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Таким образом, $R_{\mathcal{A}_1} \neq R_{\mathcal{A}_2}$

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Таким образом, $R_{\mathcal{A}_1} \neq R_{\mathcal{A}_2}$

$$\Longleftrightarrow$$

существует такое слово $w \in \Sigma^*$, что $s_0^1 \xrightarrow{w,\alpha_1}_* s'$, $s_0^2 \xrightarrow{w,\alpha_2}_* s''$, где $s' \in F_1, s'' \in F_2$ и $\alpha_1 \neq \alpha_2$

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Таким образом, $R_{\mathcal{A}_1} \neq R_{\mathcal{A}_2}$

$$\Longleftrightarrow$$

существует такое слово $w \in \Sigma^*$, что $s_0^1 \xrightarrow{w,\alpha_1}_* s'$, $s_0^2 \xrightarrow{w,\alpha_2}_* s''$, где $s' \in F_1, s'' \in F_2$ и $\alpha_1 \neq \alpha_2$

$$\Longleftrightarrow$$

согласно Лемме 1, верно одно из двух

1) существует такое слово $w \in \Sigma^*$, что $q_0 \xrightarrow{w}_* (s', s'', [\beta, \varepsilon])$ (в случае $\alpha_1{=}\alpha_2\beta$) или $q_0 \xrightarrow{w}_* (s', s'', [\varepsilon, \beta])$ (в случае $\alpha_1\beta{=}\alpha_2$), и при этом $s' \in F_1, s'' \in F_2$ и $\beta \neq \varepsilon$;

2) существует такое слово $w' \in \Sigma^*$, что $q_0 \xrightarrow{w'}_* (s', s'', \mathrm{fail})$ (в случае, если ни одна из строк α_1, α_2 является собственным префиксом другой.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Таким образом, $R_{\mathcal{A}_1} \neq R_{\mathcal{A}_2}$

$$\Longleftrightarrow$$

существует такое слово $w \in \Sigma^*$, что $s_0^1 \xrightarrow{w,\alpha_1}_* s'$, $s_0^2 \xrightarrow{w,\alpha_2}_* s''$, где $s' \in F_1, s'' \in F_2$ и $\alpha_1 \neq \alpha_2$

$$\Longleftrightarrow$$

согласно Лемме 1, верно одно из двух

1) существует такое слово $w \in \Sigma^*$, что $q_0 \xrightarrow{w}_* (s', s'', [\beta, \varepsilon])$ (в случае $\alpha_1{=}\alpha_2\beta$) или $q_0 \xrightarrow{w}_* (s', s'', [\varepsilon, \beta])$ (в случае $\alpha_1\beta{=}\alpha_2$), и при этом $s' \in F_1, s'' \in F_2$ и $\beta \neq \varepsilon$;

2) существует такое слово $w' \in \Sigma^*$, что $q_0 \xrightarrow{w'}_* (s', s'', \mathrm{fail})$ (в случае, если ни одна из строк α_1, α_2 является собственным префиксом другой.

$$\Longleftrightarrow$$

$L(\mathcal{A}_1 \times \mathcal{A}_2) \neq \varnothing$.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Соотношение $R_{\mathcal{A}_1} \neq R_{\mathcal{A}_2} \Longleftrightarrow L(\mathcal{A}_1 \times \mathcal{A}_2) \neq \varnothing$ еще не дает алгоритма решения задачи: ведь автомат $\mathcal{A}_1 \times \mathcal{A}_2$ имеет бесконечно много состояний.

А это означает, что достижимость множества финальных состояний H из начального состояния q_0 придется искать неограниченным перебором.

Как же сократить этот перебор?

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Соотношение $R_{\mathcal{A}_1} \neq R_{\mathcal{A}_2} \Longleftrightarrow L(\mathcal{A}_1 \times \mathcal{A}_2) \neq \varnothing$ еще не дает алгоритма решения задачи: ведь автомат $\mathcal{A}_1 \times \mathcal{A}_2$ имеет бесконечно много состояний.

А это означает, что достижимость множества финальных состояний H из начального состояния q_0 придется искать неограниченным перебором.

Как же сократить этот перебор?

Для этого понадобится еще одна лемма.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Лемма 2. Пусть $L(\mathcal{A}_1 \times \mathcal{A}_2) = \varnothing$. Предположим, что для некоторой пары входных слов w', w'' в автомате $\mathcal{A}_1 \times \mathcal{A}_2$ существуют вычисления

$$q_0 \xrightarrow{w'}_* (s_1, s_2, [\beta_1', \beta_2']),$$
$$q_0 \xrightarrow{w''}_* (s_1, s_2, [\beta_1'', \beta_2'']).$$

Тогда $[\beta_1', \beta_2'] = [\beta_1'', \beta_2'']$.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

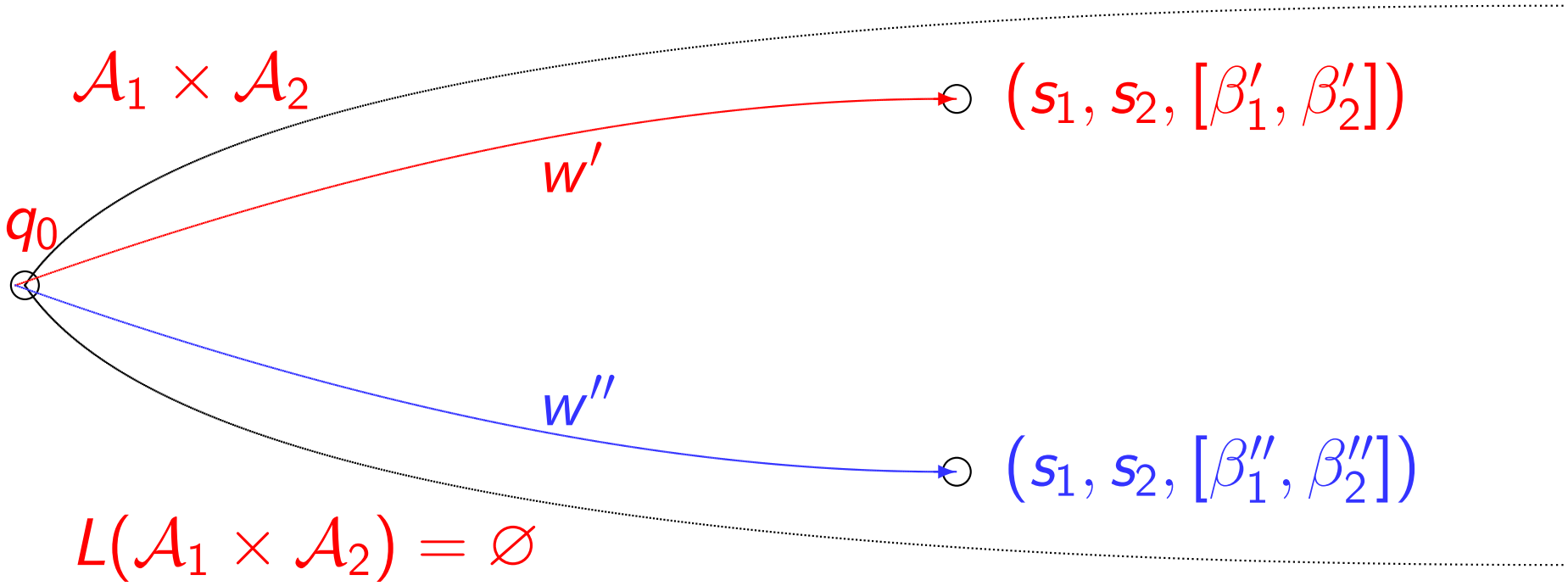
Лемма 2. Пусть $L(\mathcal{A}_1 \times \mathcal{A}_2) = \varnothing$. Предположим, что для некоторой пары входных слов w', w'' в автомате $\mathcal{A}_1 \times \mathcal{A}_2$ существуют вычисления

$$q_0 \xrightarrow{w'}_* (s_1, s_2, [\beta_1', \beta_2']),$$
$$q_0 \xrightarrow{w''}_* (s_1, s_2, [\beta_1'', \beta_2'']).$$

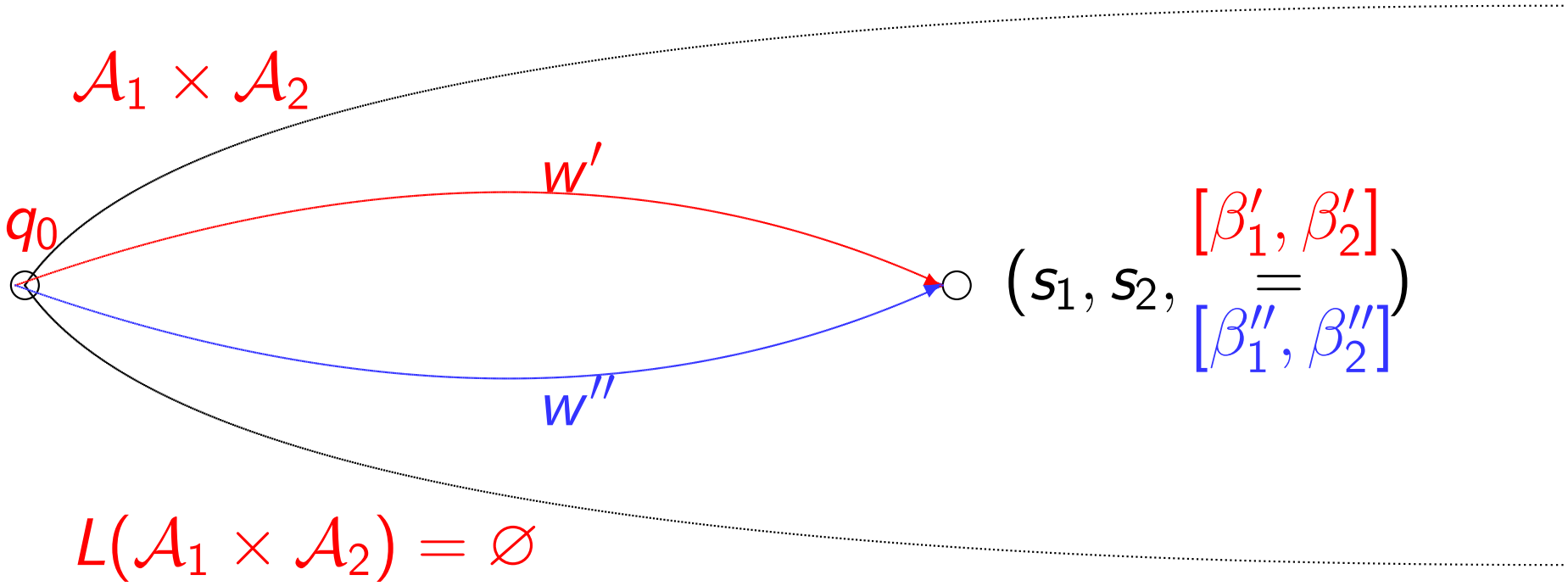
Тогда $[\beta_1', \beta_2'] = [\beta_1'', \beta_2'']$.

Из Леммы 2 следует, что в случае $L(\mathcal{A}_1 \times \mathcal{A}_2) = \varnothing$ из начального состояния q_0 автомата $\mathcal{A}_1 \times \mathcal{A}_2$ достижимо не более $|S_1| \times |S_2|$ различных состояний. И поэтому пространство поиска становится ограниченным.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Доказательство леммы 2. Без ограничения общности будем полагать $\beta_1' = \varepsilon$ (в случае $\beta_2' = \varepsilon$ доказательство проводится аналогичным образом).

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

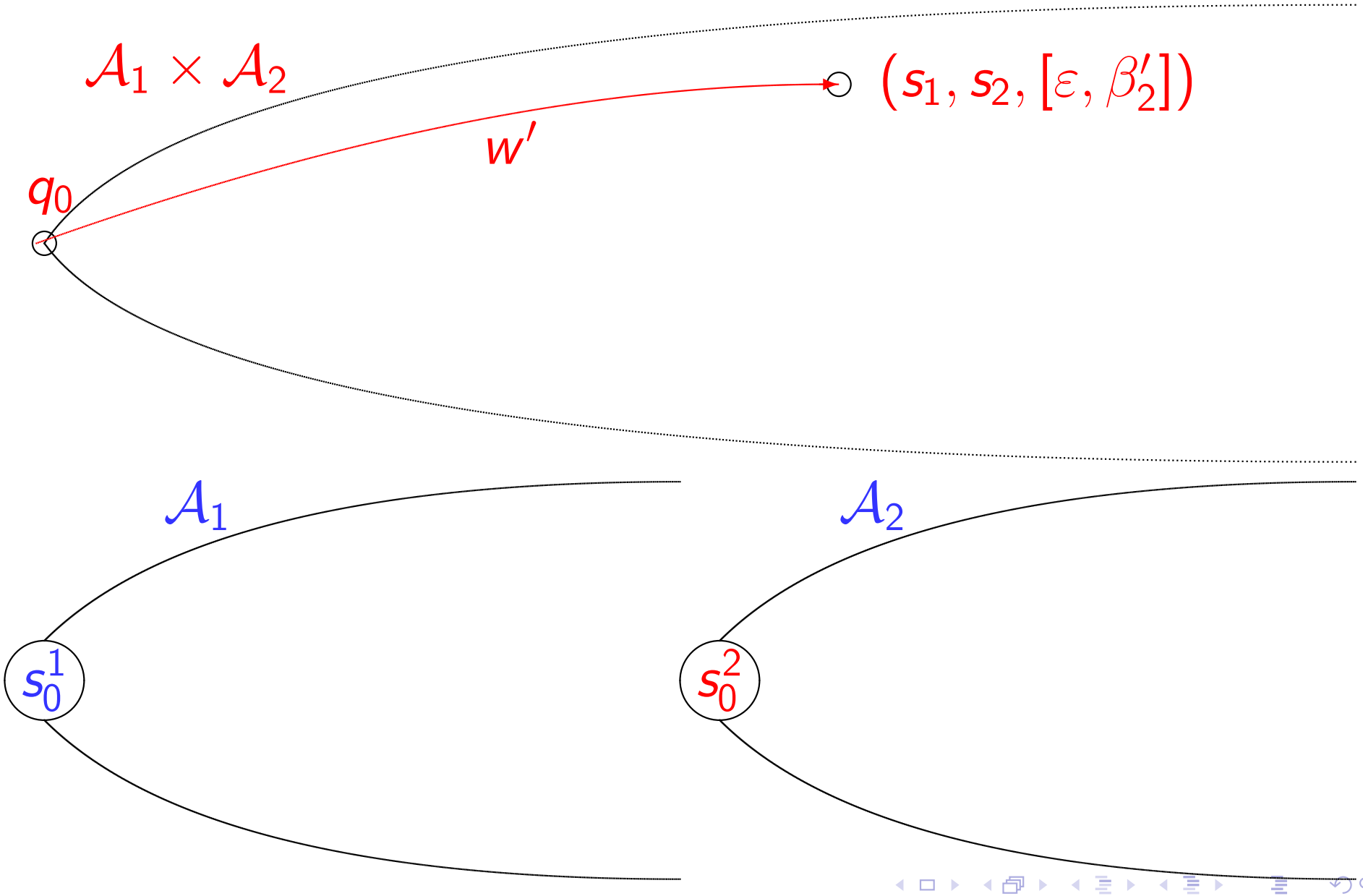
Доказательство леммы 2. Без ограничения общности будем полагать $\beta_1' = \varepsilon$ (в случае $\beta_2' = \varepsilon$ доказательство проводится аналогичным образом).

Согласно Лемме 1 если $q_0 \xrightarrow{w'}_* (s_1, s_2, [\varepsilon, \beta_2'])$, то в автоматах-преобразователях $\mathcal{A}_1$ и $\mathcal{A}_2$ есть вычисления

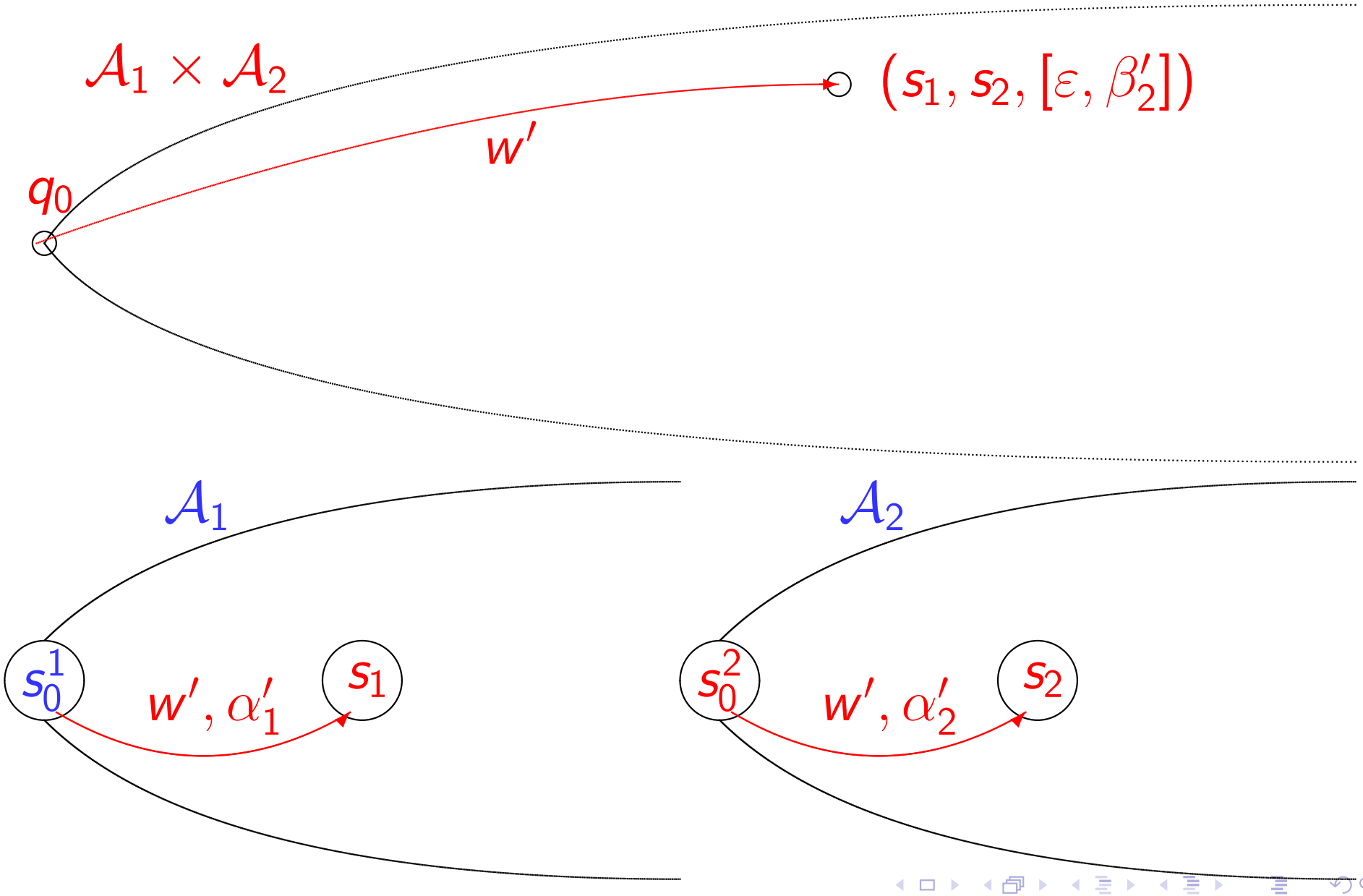
$$s_0^1 \xrightarrow{w', \alpha_1'}_* s_1,$$
$$s_0^2 \xrightarrow{w', \alpha_2'}_* s_2.$$

и при этом $Dif(\alpha_1', \alpha_2') = (\varepsilon, \beta_2')$.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

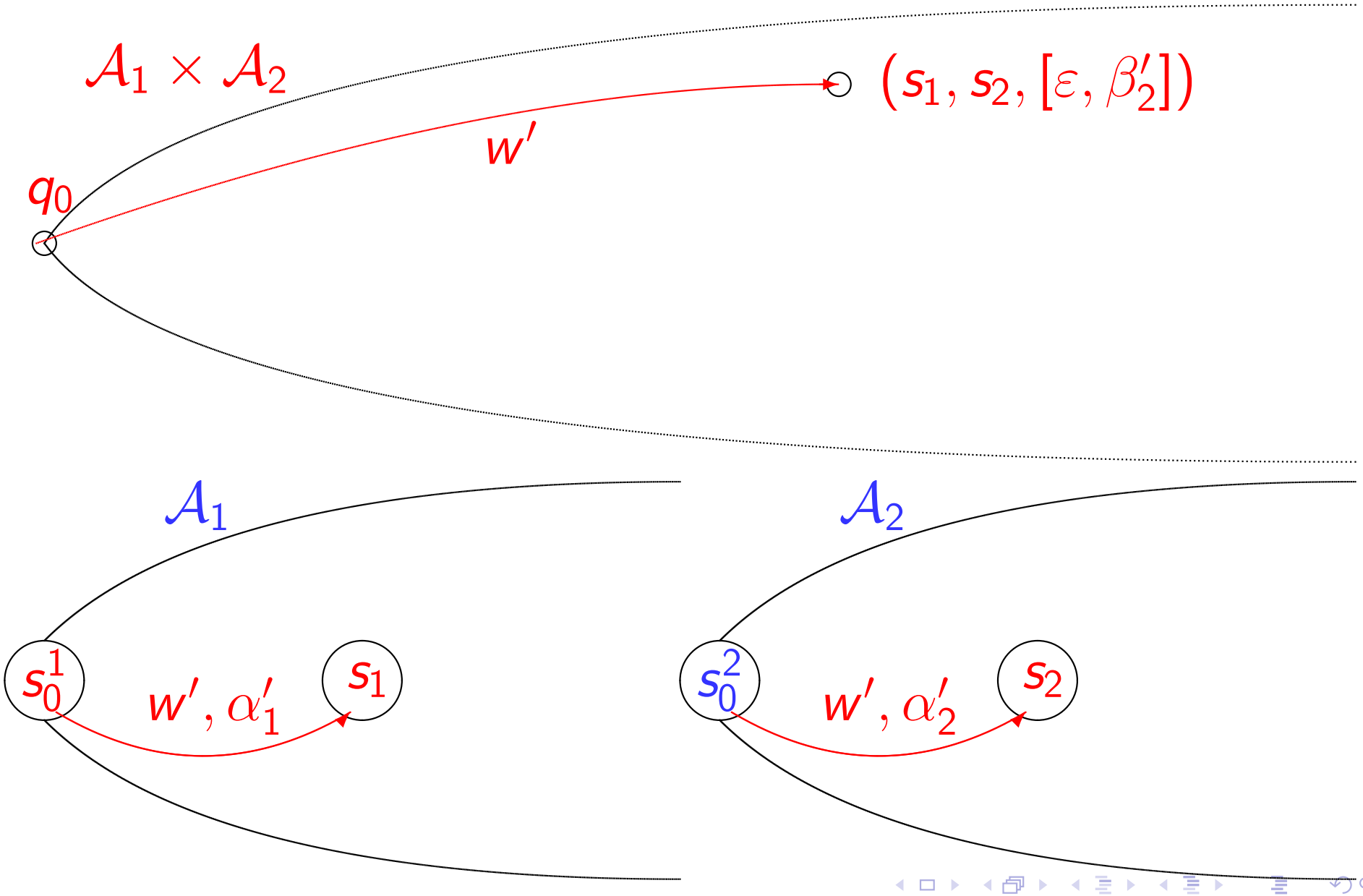


ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

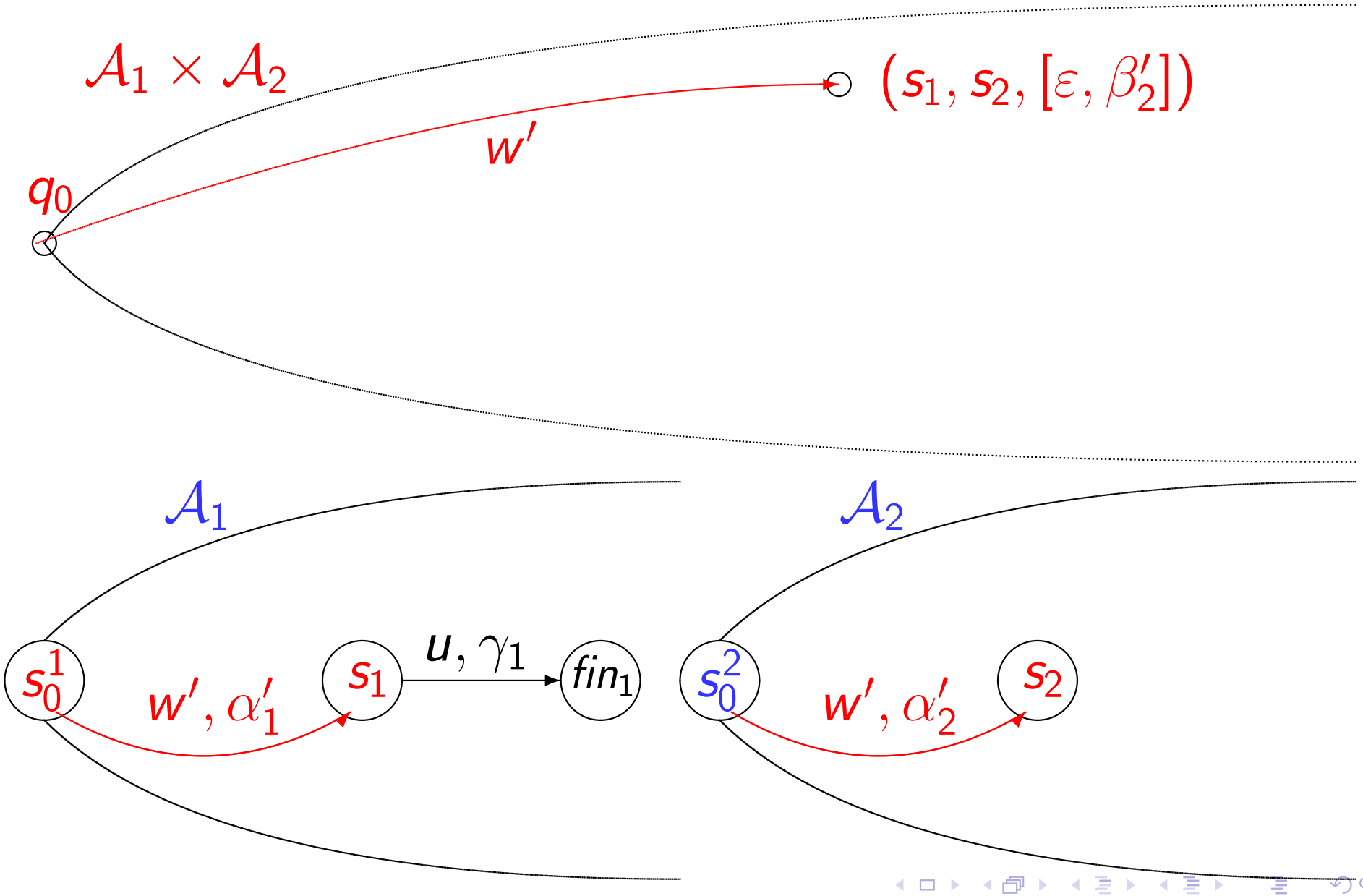
Доказательство леммы 2. Поскольку в автомате $\mathcal{A}_1$ нет бесполезных состояний, для некоторого входного слова u в этом автомате есть финальное вычисление из состояния s_1

$$s_1 \xrightarrow{u,\gamma_1} fin_1, \;\; fin_1 \in F_1$$

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Доказательство леммы 2. Поскольку в автомате $\mathcal{A}_1$ нет бесполезных состояний, для некоторого входного слова u в этом автомате есть финальное вычисление из состояния s_1

$$s_1 \xrightarrow{u,\gamma_1} fin_1, \;\; fin_1 \in F_1$$

Значит, $R_{\mathcal{A}_1}(w'u) = \alpha_1'\gamma_1$.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Доказательство леммы 2. Поскольку в автомате $\mathcal{A}_1$ нет бесполезных состояний, для некоторого входного слова u в этом автомате есть финальное вычисление из состояния s_1

$$s_1 \xrightarrow{u,\gamma_1} fin_1, \;\; fin_1 \in F_1$$

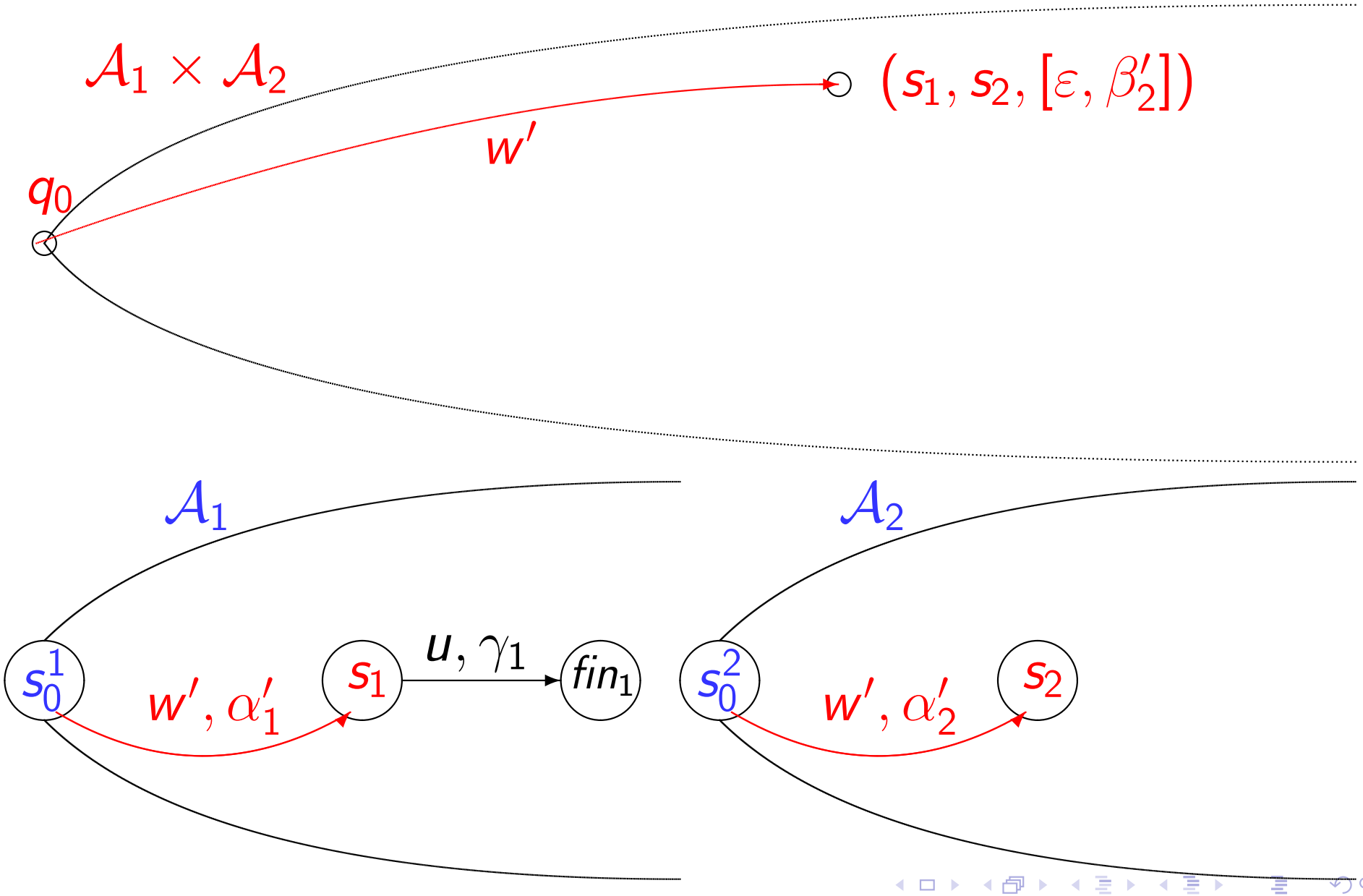
Значит, $R_{\mathcal{A}_1}(w'u) = \alpha'_1\gamma_1$.

Поскольку $L(\mathcal{A}_1 \times \mathcal{A}_2) = \varnothing$, автоматы $\mathcal{A}_1$ и $\mathcal{A}_2$ эквивалентны. Поэтому $R_{\mathcal{A}_1}(w'u) = R_{\mathcal{A}_1}(w'u)$. Значит, автомат $\mathcal{A}_1$ имеет успешное вычисление

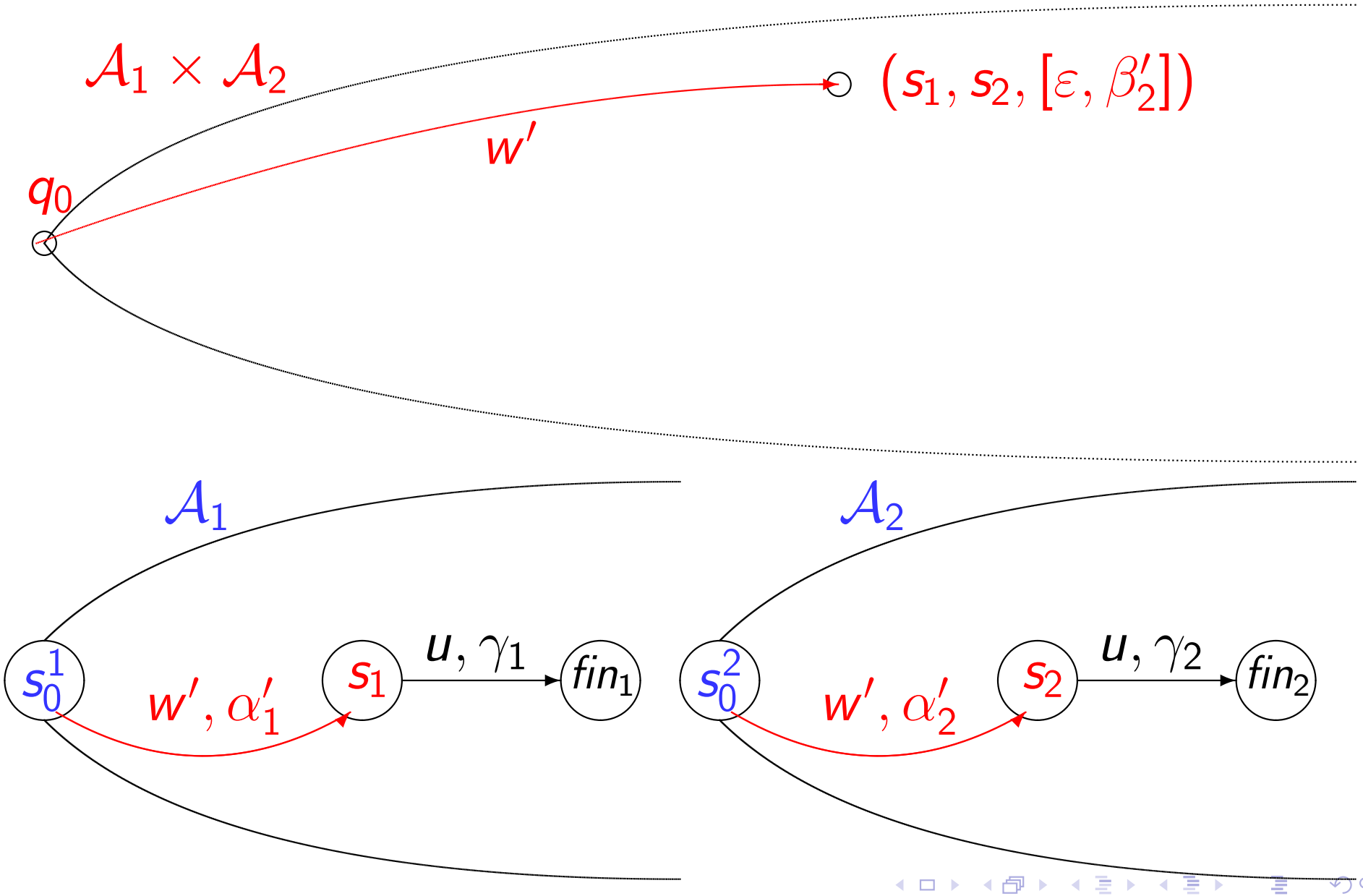
$$s_0^2 \xrightarrow{w',\alpha'_2} s_2 \xrightarrow{u,\gamma_2} fin_2, \;\; fin_2 \in F_2,$$

для которого имеет место равенство $\alpha'_1\gamma_1 = \alpha'_2\gamma_2$.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Доказательство леммы 2. Равенства $\alpha_1'\gamma_1 = \alpha_2'\gamma_2$ и $Dif(\alpha_1'.\alpha_2') = [\varepsilon, \beta_2']$ влекут

$$\begin{aligned} Dif(\alpha_1'\gamma_1, \alpha_2'\gamma_2) &= [\varepsilon, \varepsilon] \\ Dif(\alpha_1'\gamma_1, \alpha_2'\gamma_2) &= Dif(\gamma_1, \beta_2'\gamma_2). \end{aligned}$$

Таким образом, $\gamma_1 = \beta_2'\gamma_2$.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

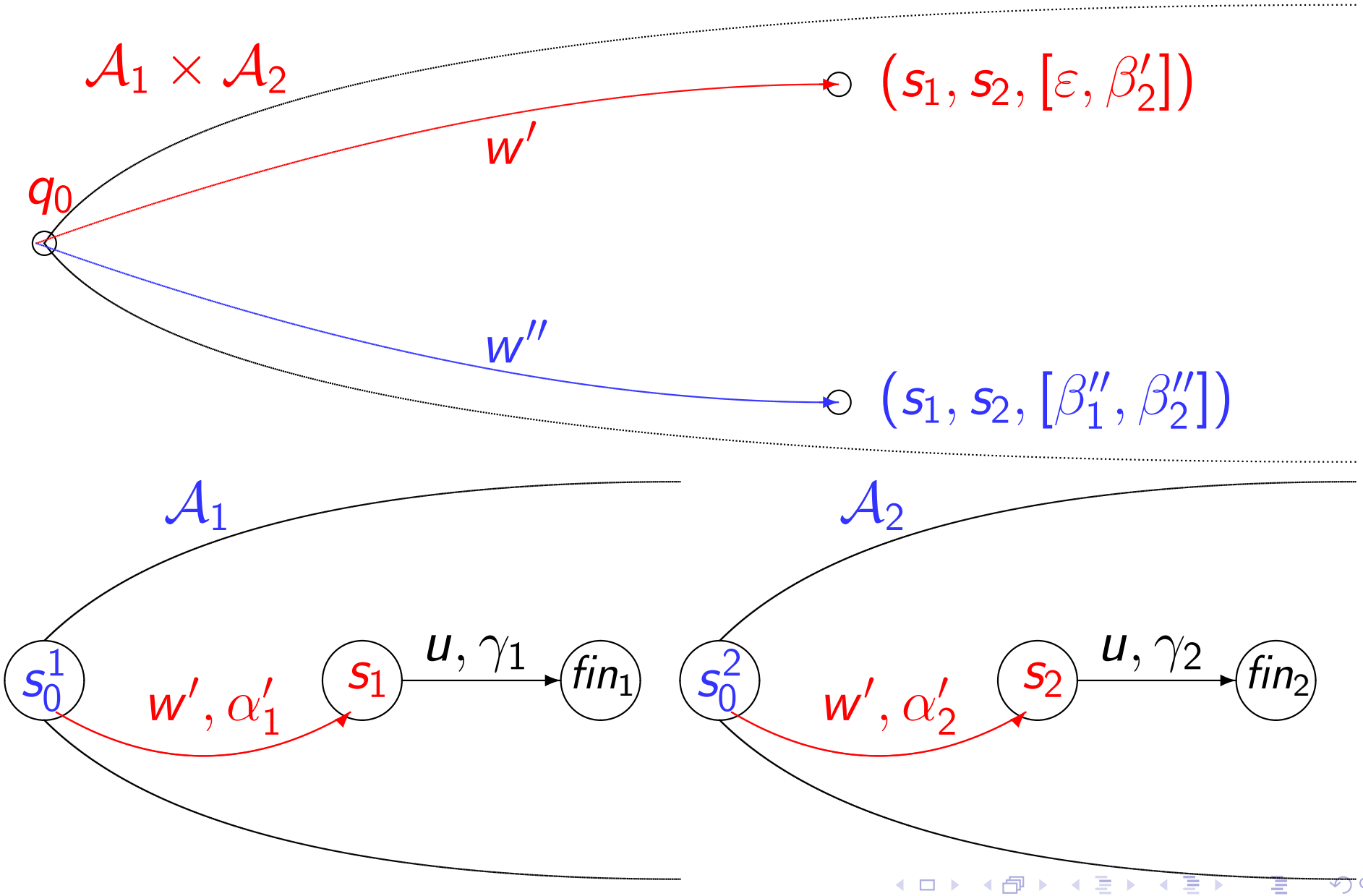
Доказательство леммы 2. Равенства $\alpha_1'\gamma_1 = \alpha_2'\gamma_2$ и $Dif(\alpha_1'.\alpha_2') = [\varepsilon, \beta_2']$ влекут

$$\begin{aligned} Dif(\alpha_1'\gamma_1, \alpha_2'\gamma_2) &= [\varepsilon, \varepsilon] \\ Dif(\alpha_1'\gamma_1, \alpha_2'\gamma_2) &= Dif(\gamma_1, \beta_2'\gamma_2). \end{aligned}$$

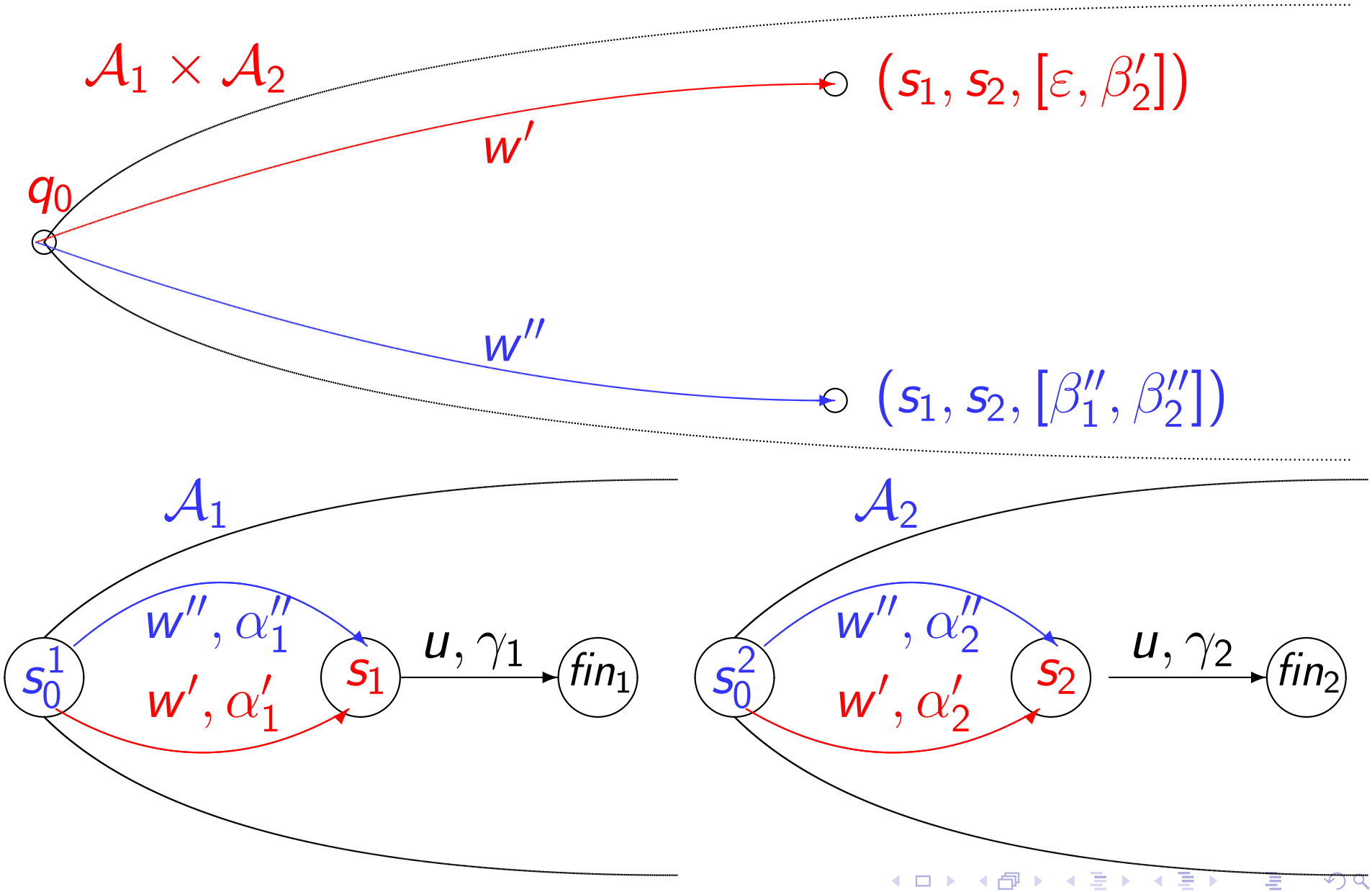
Таким образом, $\gamma_1 = \beta_2'\gamma_2$.

Точно такие же рассуждения можно провести и для вычисления $q_0 \xrightarrow{w''}_* (s_1, s_2, [\beta_1'', \beta_2''])$ в автомате $\mathcal{A}_1 \times \mathcal{A}_2$.

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ



ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Доказательство леммы 2. Таким образом, мы приходим к следующим равенствам

$$\begin{array}{lcl} \gamma_1 & = & \beta_2'\gamma_2 \\ \beta_1''\gamma_1 & = & \beta_2''\gamma_2 \\ \beta_1'' = \varepsilon & \vee & \beta_2'' = \varepsilon. \end{array}$$

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Доказательство леммы 2. Таким образом, мы приходим к следующим равенствам

$$\begin{array}{lcl} \gamma_1 & = & \beta_2'\gamma_2 \\ \beta_1''\gamma_1 & = & \beta_2''\gamma_2 \\ \beta_1'' = \varepsilon & \vee & \beta_2'' = \varepsilon. \end{array}$$

Отсюда следует $\beta_1'' = \varepsilon = \beta_1'$ и $\beta_2'' = \beta_1''$. QED

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Доказательство Теоремы 10.8. Для проверки эквивалентности автоматов-преобразователей $\mathcal{A}_1$ и $\mathcal{A}_2$ рассмотрим автомат $\mathcal{A}_1 \times \mathcal{A}_2$, в котором проведем поиск финальных состояний, достижимых из инициального состояния.

Если такое финальное состояние будет найдено или в процессе поиска удастся достичь более $|S_1| \times |S_2|$ состояний, то автоматы преобразователи $\mathcal{A}_1$ и $\mathcal{A}_2$ неэквивалентны.

В противном случае автоматы $\mathcal{A}_1$ и $\mathcal{A}_2$ эквивалентны

QED

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Задача 7. Какова сложность алгоритма проверки эквивалентности конечных детерминированных автоматов-преобразователей?

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Задача 7. Какова сложность алгоритма проверки эквивалентности конечных детерминированных автоматов-преобразователей?

Недетерминированный автомат-преобразователь $\mathcal{A}$ называется фуннкциональным , если реализуемое им отношение $R_{\mathcal{A}}$ является функцией, т.е. для любого входного слова w существует не более одной такой строки α , что $(w, \alpha) \in R_{\mathcal{A}}$.

Задача 8 [Трудная]. Существует ли алгоритм проверки свойства функциональности для конечных автоматов-преобразователей?

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

ИТОГИ.

Конечные автоматы-преобразователи занимают промежуточное положение между автоматами-распознавателями и магазинными автоматами.

Для разрешимых задач анализа поведения автоматов-преобразователей можно использовать такие же методы, как и для анализа конечных автоматов-распознавателей.

А для доказательства неразрешимости можно применять те же приемы, которые были использованы для магазинных автоматов.

АВТОМАТЫ-ПРЕОБРАЗОВАТЕЛИ

ИТОГИ.

Конечные автоматы-преобразователи занимают промежуточное положение между автоматами-распознавателями и магазинными автоматами.

Для разрешимых задач анализа поведения автоматов-преобразователей можно использовать такие же методы, как и для анализа конечных автоматов-распознавателей.

А для доказательства неразрешимости можно применять те же приемы, которые были использованы для магазинных автоматов.

Все эти методы являются многоразовыми!

ДЕТЕРМИНИРОВАННЫЕ ПРЕОБРАЗОВАТЕЛИ

Задача 9 [Трудная]. А как построить алгоритм минимизации конечных детерминированных автоматов-преобразователей?

КОНЕЦ ЛЕКЦИИ 10