Модели вычислений

В.А. Захаров, Р.И. Подловченко

Лекция 12.

1. Логический способ описания языков
2. Монадическая логика предикатов второго порядка S1S
3. Логика предикатов S1S и ω-автоматы
4. Другие логики предикатов второго порядка

Способы описания языков

Нам уже известны разные способы описания формальных языков:

Способы описания языков

Нам уже известны разные способы описания формальных языков:

- **автоматный**: язык состоит из слов, распознаваемых некоторым вычислительным устройством (автоматом);

Способы описания языков

Нам уже известны разные способы описания формальных языков:

- **автоматный**: язык состоит из слов, распознаваемых некоторым вычислительным устройством (автоматом);
- **грамматический** : язык состоит из слов, конструируемых при помощи грамматических правил;

Способы описания языков

Нам уже известны разные способы описания формальных языков:

- **автоматный**: язык состоит из слов, распознаваемых некоторым вычислительным устройством (автоматом);
- **грамматический** : язык состоит из слов, конструируемых при помощи грамматических правил;
- **алгебраический** : язык состоит из слов, вычисляемых при помощи алгебраических операций.

Способы описания языков

Нам уже известны разные способы описания формальных языков:

- **автоматный**: язык состоит из слов, распознаваемых некоторым вычислительным устройством (автоматом);
- **грамматический** : язык состоит из слов, конструируемых при помощи грамматических правил;
- **алгебраический** : язык состоит из слов, вычисляемых при помощи алгебраических операций.

Но бывают случаи, когда ни один из этих способов не подходит для описания нужного языка.

Способы описания языков

Как, например, формально описать язык, состоящий из всех слов, в которых между любыми двумя вхождениями букв a и b имеется хотя бы одно вхождение буквы c , а между любыми двумя вхождениями букв c и b имеется хотя бы одно вхождение буквы a ?

Способы описания языков

Как, например, формально описать язык, состоящий из всех слов, в которых между любыми двумя вхождениями букв a и b имеется хотя бы одно вхождение буквы c, а между любыми двумя вхождениями букв c и b имеется хотя бы одно вхождение буквы a?

Здесь, вероятно, был бы уместен не **операционный** способ задания, который инструктирует, как построить нужные слова, а **декларативный** способ задания, который описывает, каким требованиям должны удовлетворять интересующие нас слова.

Логический способ описания языков

В качестве средства декларативного задания языков можно использовать логические формулы.

Логический способ описания языков

В качестве средства декларативного задания языков можно использовать логические формулы.

Но как приспособить логику для описания устройства слов?

Какое множество нужно выбрать предметной областью?

Какие предикаты целесообразно использовать?

ЛОГИЧЕСКИЙ СПОСОБ ОПИСАНИЯ ЯЗЫКОВ

Попробуем рассуждать так.

ω-слово — это функция $\alpha : \mathbb{N} \to \Sigma$.

Поэтому в качестве предметной области можно использовать натуральный ряд $\mathbb{N}$, рассматривая числа как номера позиций в слове.

Каждая буква алфавита Σ может восприниматься как определенное свойство позиции в слове.

А сами позиции можно сравнивать по отношению порядка как натуральные числа.

Логический способ описания языков

Тогда требование «Между любыми двумя вхождениями букв a и b имеется хотя бы одно вхождение буквы c » можно формально задать следующей формулой:

$$\forall x \forall y (A(x) \wedge B(y) \wedge (x < y) \rightarrow \exists z ((x < z) \wedge (z < y) \wedge C(z))).$$

Логический способ описания языков

Тогда требование «Между любыми двумя вхождениями букв a и b имеется хотя бы одно вхождение буквы c » можно формально задать следующей формулой:

$$\forall x \forall y (A(x) \wedge B(y) \wedge (x < y) \rightarrow \exists z ((x < z) \wedge (z < y) \wedge C(z))).$$

А как поступить, имея дело с таким требованием: «Буква a может быть только в позициях с четными номерами, а в каждой седьмой позиции должна быть буква b »?

Значит, в логике должны быть счетчик позиций.

Логический способ описания языков

А как поступать, когда нужно задать требование «Буква a располагается в четном числе позиций»?

Логический способ описания языков

А как поступать, когда нужно задать требование «Буква a располагается в четном числе позиций»?

Оказывается, что выразительных средств логики первого порядка для описания этого требования уже недостаточно.

Логический способ описания языков

Выяснилось, что для описания рассмотренных утверждений подходит **монадическая логика предикатов 2-го порядка с одной функцией следования** (S1S — S econd-order logic with 1 S uccessor).

Термин **монадическая** означает, что в формулах этой логике используются только одноместные предикаты.
Такие предикаты можно истолковывать как множества позиций, и запись $P(x)$ обозначает отношение включения $x \in P$.

ЛОГИЧЕСКИЙ СПОСОБ ОПИСАНИЯ ЯЗЫКОВ

Термин **2-го порядка** означает, что в формулах этой логике разрешается использовать переменные двух типов:

1. предметные переменные , значениями которых являются номера позиций в слове;
2. предикатные переменные , значениями которых являются отношения на номерах позиций в слове.

При этом кванторы разрешается применять к переменным обоих типов.

Логический способ описания языков

Термин **1 функция следования** означает, что в формулах этой логике разрешается использовать функцию, которая для каждой позиции в слове вычисляет номер следующей позиции.

Существуют также монадические логики второго порядка с несколькими функциями следования (SnS), которые можно использовать для описания древесных языков — деревьев, вершины которых помечены буквами.

ЛОГИКА S1S

Синтаксис.

Алфавит логики S1S состоит из

- Счетно-бесконечного множества предметных переменных $Var = \{x_1, x_2, \dots,\}$;
- Счетно-бесконечного множества одноместных переменных-предикатов $Set = \{X_1, X_2, \dots,\}$;
- Одноместного функционального символа s .

ЛОГИКА S1S

Синтаксис.

Алфавит логики S1S состоит из

- Счетно-бесконечного множества предметных переменных $Var = \{x_1, x_2, \dots, \}$;
- Счетно-бесконечного множества одноместных переменных-предикатов $Set = \{X_1, X_2, \dots, \}$;
- Одноместного функционального символа s .

Термы логики S1S — это выражения, устройство которых определяется следующими правилами:

- всякая предметная переменная $x, x \in Var$, является термом;
- Если t — терм, то выражение $s(t)$ — это терм.

Множество всех термов обозначим записью $Term$.

Синтаксис.

Формулы логики S1S — это выражения, устройство которых определяется следующими правилами

- Если $X \in Set$ и $t \in Term$, то $X(t)$ — формула (атомарная формула);
- Если φ_1 и φ_2 — формулы, то выражения $(\neg\varphi_1)$ и $(\varphi_1 \wedge \varphi_2)$ — формулы;
- Если φ — формула и $x \in Var$, то выражение $(\exists x \varphi)$ — формула;
- Если φ — формула и $X \in Set$, то выражение $(\exists X \varphi)$ — формула.

Чтобы избавиться от избытка скобок, воспользуемся соглашением о приоритете связок.

ЛОГИКА S1S

Семантика.

Значения термов и формул логики S1S определяются на интерпретациях I , предметной областью которых служит множество натуральных чисел $\mathbb{N}$.

Каждая интерпретация I определяется парой оценок (σ_1, σ_2) , где

$\sigma_1 : Var \to \mathbb{N}$ (оценка предметных переменных);

$\sigma_2 : Set \to 2^{\mathbb{N}}$ (оценка переменных-множеств).

ЛОГИКА S1S

Семантика.

Значения термов и формул логики S1S определяются на интерпретациях I , предметной областью которых служит множество натуральных чисел $\mathbb{N}$.
Каждая интерпретация I определяется парой оценок (σ_1, σ_2) , где
$\sigma_1 : Var \to \mathbb{N}$ (оценка предметных переменных);
$\sigma_2 : Set \to 2^{\mathbb{N}}$ (оценка переменных-множеств).
Значением каждой предметной переменной является натуральное число (позиция в слове).
Значением каждой переменной-множества является множество натуральных чисел (множество позиций в слове).

Семантика.
Для каждого терма t его значение $[t]_I$ в интерпретации $I = (\sigma_1, \sigma_2)$ определяется по следующим правилам:

1. $[x]_I = \sigma_1(x)$ для всякой предметной переменной x ;
2. $[s(t)]_I = [t]_I + 1$.

ЛОГИКА S1S

Семантика.

Условимся использовать запись $I[n/y]$, где $I = (\sigma_1, \sigma_2), y \in Var, n \in \mathbb{N}$, для обозначения интерпретации $I' = (\sigma_1', \sigma_2)$, в которой

$$\sigma_1'(x) = \begin{cases} \sigma_1(x), \text{если } x \neq y, \\ n, \text{если } x = y. \end{cases}$$

Логика S1S

Семантика.
Условимся использовать запись $I[n/y]$, где $I = (\sigma_1, \sigma_2), y \in Var, n \in \mathbb{N}$, для обозначения интерпретации $I' = (\sigma_1', \sigma_2)$, в которой

$$\sigma_1'(x) = \begin{cases} \sigma_1(x), \text{если } x \neq y, \\ n, \text{если } x = y. \end{cases}$$

Аналогичным образом определяется и смысл записи $I[N/Y]$, где $Y \in Set, N \subseteq \mathbb{N}$.

ЛОГИКА S1S

Семантика.
Для каждой формулы φ ее выполнимость в интерпретации $I = (\sigma_1, \sigma_2)$ определяется по следующим правилам:

1. $I \models X(t) \Leftrightarrow [t]_I \in \sigma_2(X)$;
2. $I \models \neg\varphi \Leftrightarrow I \not\models \varphi$;
3. $I \models \varphi_1 \wedge \varphi_2 \Leftrightarrow I \models \varphi_1$ и $I \models \varphi_2$;
4. $I \models \exists x \varphi \Leftrightarrow$ существует такое n, что $I[n/x] \models \varphi$;
5. $I \models \exists X \varphi \Leftrightarrow$
   существует такое N, что $I[N/X] \models \varphi$.

Логика S1S

Семантика.

Можно ввести и другие логические операции и связки:

$$
\begin{array}{lcl}
\varphi \vee \psi & \Leftrightarrow & \neg(\neg\varphi \wedge \neg\psi); \\
\varphi \to \psi & \Leftrightarrow & \neg\varphi \vee \psi; \\
\varphi \equiv \psi & \Leftrightarrow & (\varphi \to \psi) \wedge (\psi \to \varphi); \\
\forall x \varphi & \Leftrightarrow & \neg\exists x \neg\varphi; \\
\forall X \varphi & \Leftrightarrow & \neg\exists X \neg\varphi.
\end{array}
$$

Логика S1S

Формулами логики S1S можно выразить многие общеупотребительные свойства и отношения на множестве натуральных чисел:

$$x = y \quad :$$

$$x = 0 \quad :$$

$$x = n \quad :$$

$$Up(X) \quad :$$

$$Down(X) \quad :$$

$$x \leq y \quad :$$

$$X \subseteq Y \quad :$$

$$X = Y \quad :$$

$$X = \varnothing \quad :$$

Логика S1S

Формулами логики S1S можно выразить многие общеупотребительные свойства и отношения на множестве натуральных чисел:

$x = y$: $\forall X (X(x) \equiv X(y))$;

$x = 0$:

$x = n$:

$Up(X)$:

$Down(X)$:

$x \leq y$:

$X \subseteq Y$:

$X = Y$:

$X = \varnothing$:

Логика S1S

Формулами логики S1S можно выразить многие общеупотребительные свойства и отношения на множестве натуральных чисел:

$$
\begin{array}{lcl}
x = y & : & \forall X(X(x) \equiv X(y)); \\
x = 0 & : & \forall y \neg(s(y) = x); \\
x = n & : & \\
 & & \\
Up(X) & : & \\
Down(X) & : & \\
x \leq y & : & \\
X \subseteq Y & : & \\
X = Y & : & \\
X = \varnothing & : &
\end{array}
$$

Логика S1S

Формулами логики S1S можно выразить многие общеупотребительные свойства и отношения на множестве натуральных чисел:

$$
\begin{array}{lcl}
x = y & : & \forall X(X(x) \equiv X(y)); \\
x = 0 & : & \forall y \neg (s(y) = x); \\
x = n & : & \exists y(y = 0 \wedge x = \underbrace{ss\ldots s}_{n \text{ раз}}(y)); \\
 & & \\
Up(X) & : & \\
Down(X) & : & \\
x \leq y & : & \\
X \subseteq Y & : & \\
X = Y & : & \\
X = \varnothing & : &
\end{array}
$$

Логика S1S

Формулами логики S1S можно выразить многие общеупотребительные свойства и отношения на множестве натуральных чисел:

$$
\begin{array}{lcl}
x = y & : & \forall X (X(x) \equiv X(y)); \\
x = 0 & : & \forall y \neg (s(y) = x); \\
x = n & : & \exists y (y = 0 \wedge x = \underbrace{ss\ldots s}_{n \text{ раз}}(y)); \\
Up(X) & : & \forall z (X(z) \to X(s(z))); \\
Down(X) & : & \\
x \leq y & : & \\
X \subseteq Y & : & \\
X = Y & : & \\
X = \varnothing & : &
\end{array}
$$

Логика S1S

Формулами логики S1S можно выразить многие общеупотребительные свойства и отношения на множестве натуральных чисел:

$$
\begin{array}{lcl}
x = y & : & \forall X(X(x) \equiv X(y)); \\
x = 0 & : & \forall y \neg(s(y) = x); \\
x = n & : & \exists y(y = 0 \wedge x = \underbrace{ss\ldots s}_{n \text{ раз}}(y)); \\
Up(X) & : & \forall z(X(z) \to X(s(z))); \\
Down(X) & : & \forall z(X(s(z)) \to X(z)); \\
x \le y & : & \\
X \subseteq Y & : & \\
X = Y & : & \\
X = \varnothing & : &
\end{array}
$$

Логика S1S

Формулами логики S1S можно выразить многие общеупотребительные свойства и отношения на множестве натуральных чисел:

$$
\begin{array}{lcl}
x = y & : & \forall X(X(x) \equiv X(y)); \\
x = 0 & : & \forall y \neg(s(y) = x); \\
x = n & : & \exists y(y = 0 \wedge x = \underbrace{ss\ldots s}_{n \text{ раз}}(y)); \\
Up(X) & : & \forall z(X(z) \rightarrow X(s(z))); \\
Down(X) & : & \forall z(X(s(z)) \rightarrow X(z)); \\
x \leq y & : & \forall X(X(x) \wedge Up(X) \rightarrow X(y)); \\
X \subseteq Y & : & \\
X = Y & : & \\
X = \varnothing & : &
\end{array}
$$

Логика S1S

Формулами логики S1S можно выразить многие общеупотребительные свойства и отношения на множестве натуральных чисел:

$$
\begin{array}{lcl}
x = y & : & \forall X (X(x) \equiv X(y)); \\
x = 0 & : & \forall y \neg (s(y) = x); \\
x = n & : & \exists y (y = 0 \wedge x = \underbrace{ss \ldots s}_{n \text{ раз}}(y)); \\
Up(X) & : & \forall z (X(z) \to X(s(z))); \\
Down(X) & : & \forall z (X(s(z)) \to X(z)); \\
x \leq y & : & \forall X (X(x) \wedge Up(X) \to X(y)); \\
X \subseteq Y & : & \forall x (X(x) \to Y(x)); \\
X = Y & : & \\
X = \varnothing & : &
\end{array}
$$

Логика S1S

Формулами логики S1S можно выразить многие общеупотребительные свойства и отношения на множестве натуральных чисел:

$$
\begin{array}{lcl}
x = y & : & \forall X(X(x) \equiv X(y)); \\
x = 0 & : & \forall y \neg(s(y) = x); \\
x = n & : & \exists y(y = 0 \wedge x = \underbrace{ss\ldots s}_{n \text{ раз}}(y)); \\
Up(X) & : & \forall z(X(z) \rightarrow X(s(z))); \\
Down(X) & : & \forall z(X(s(z)) \rightarrow X(z)); \\
x \leq y & : & \forall X(X(x) \wedge Up(X) \rightarrow X(y)); \\
X \subseteq Y & : & \forall x(X(x) \rightarrow Y(x)); \\
X = Y & : & X \subseteq Y \wedge Y \subseteq X; \\
X = \varnothing & : &
\end{array}
$$

Логика S1S

Формулами логики S1S можно выразить многие общеупотребительные свойства и отношения на множестве натуральных чисел:

$$
\begin{array}{lcl}
x = y & : & \forall X(X(x) \equiv X(y)); \\
x = 0 & : & \forall y \neg(s(y) = x); \\
x = n & : & \exists y(y = 0 \wedge x = \underbrace{ss\ldots s}_{n \text{ раз}}(y)); \\
Up(X) & : & \forall z(X(z) \to X(s(z))); \\
Down(X) & : & \forall z(X(s(z)) \to X(z)); \\
x \leq y & : & \forall X(X(x) \wedge Up(X) \to X(y)); \\
X \subseteq Y & : & \forall x(X(x) \to Y(x)); \\
X = Y & : & X \subseteq Y \wedge Y \subseteq X; \\
X = \varnothing & : & \forall x \neg X(x).
\end{array}
$$

Логика S1S

А также и такие отношения:

$$\begin{aligned}
first(x, X) &: \\
last(x, X) &: \\
Finite(X) &: \\
Single(X) &: \\
Even(X) &:
\end{aligned}$$

Логика S1S

А также и такие отношения:

$$
\begin{aligned}
\mathit{first}(x, X) &: \quad X(x) \wedge \forall y (X(y) \to x \leq y); \\
\mathit{last}(x, X) &: \\
\mathit{Finite}(X) &: \\
\mathit{Single}(X) &: \\
\mathit{Even}(X) &:
\end{aligned}
$$

Логика S1S

А также и такие отношения:

$$
\begin{aligned}
\mathit{first}(x, X) &: \quad X(x) \wedge \forall y(X(y) \to x \leq y);\\
\mathit{last}(x, X) &: \quad X(x) \wedge \forall y(X(y) \to y \leq x);\\
\mathit{Finite}(X) &: \\
\mathit{Single}(X) &: \\
\mathit{Even}(X) &:
\end{aligned}
$$

Логика S1S

А также и такие отношения:

$$
\begin{aligned}
\mathit{first}(x, X) &: \quad X(x) \wedge \forall y (X(y) \to x \leq y); \\
\mathit{last}(x, X) &: \quad X(x) \wedge \forall y (X(y) \to y \leq x); \\
\mathit{Finite}(X) &: \quad \exists y \forall x (X(x) \to x \leq y); \\
\mathit{Single}(X) &: \\
\mathit{Even}(X) &:
\end{aligned}
$$

Логика S1S

А также и такие отношения:

$$\begin{array}{lcl}
\mathit{first}(x, X) & : & X(x) \wedge \forall y(X(y) \to x \leq y); \\
\mathit{last}(x, X) & : & X(x) \wedge \forall y(X(y) \to y \leq x); \\
\mathit{Finite}(X) & : & \exists y \forall x(X(x) \to x \leq y); \\
\mathit{Single}(X) & : & \exists x(X(x) \wedge \forall y(X(y) \to x = y)); \\
\mathit{Even}(X) & : &
\end{array}$$

Логика S1S

А также и такие отношения:

$$
\begin{array}{lcl}
\mathit{first}(x, X) & : & X(x) \wedge \forall y(X(y) \rightarrow x \leq y); \\
\mathit{last}(x, X) & : & X(x) \wedge \forall y(X(y) \rightarrow y \leq x); \\
\mathit{Finite}(X) & : & \exists y \forall x(X(x) \rightarrow x \leq y); \\
\mathit{Single}(X) & : & \exists x(X(x) \wedge \forall y(X(y) \rightarrow x = y)); \\
\mathit{Even}(X) & : & \exists y(y = 0 \wedge X(y) \wedge \neg X(s(y))) \wedge \\
 & & \forall x(X(x) \equiv X(ss(x))).
\end{array}
$$

Логика S1S

Натуральные числа и множества натуральных чисел можно представлять бесконечными двоичными строками — ω-словами в алфавите $\{0, 1\}$ — следующим образом:

числу n соответствует строка

$$string(n) = \delta_0\delta_1\delta_2 \ldots \delta_i \ldots,$$

в которой $\delta_i = 1 \Leftrightarrow i = n$;

множеству чисел N соответствует строка

$$string(N) = \kappa_0\kappa_1\kappa_2 \ldots \kappa_i \ldots,$$

в которой $\kappa_i = 1 \Leftrightarrow i \in N$.

Логика S1S

Например,

$$string(4) = 00001000000000\ldots,$$

$$string(Prime) = 0011010100010100010100010000010\ldots$$

Логика S1S

Каждый набор $\;= \langle n_1, n_2, \ldots, n_k, N_1, N_2, \ldots, N_m \rangle$, состоящий из натуральных чисел $n_1, n_2, \ldots, n_k$ и множеств натуральных чисел $N_1, N_2, \ldots, N_m$ можно представить ω-словом в алфавите $\{0, 1\}^{k+m}$ (бесконечной строкой двоичных наборов) $string(C)$ вида

$$\begin{pmatrix} \delta_0^1 \\ \vdots \\ \delta_0^k \\ \kappa_0^1 \\ \vdots \\ \kappa_0^m \end{pmatrix} \begin{pmatrix} \delta_1^1 \\ \vdots \\ \delta_1^k \\ \kappa_1^1 \\ \vdots \\ \kappa_1^m \end{pmatrix} \ldots \begin{pmatrix} \delta_i^1 \\ \vdots \\ \delta_i^k \\ \kappa_i^1 \\ \vdots \\ \kappa_i^m \end{pmatrix} \ldots$$

Логика S1S

Каждый двоичный набор можно рассматривать как двоичный код некоторой буквы алфавита Σ .

Например, строку двоичных наборов

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \dots$$

можно воспринимать как ω-слово

$$abbcddc \dots$$

Логика S1S

Для каждой интерпретации $I = (\sigma_1, \sigma_2)$ и для любого набора $(\vec{x}, \vec{X}) = (x_1, \ldots, x_k, X_1, \ldots, X_m)$, состоящего из предметных переменных $x_1, \ldots, x_k$ и переменных множеств $X_1, \ldots, X_m$, обозначим записью $[(\vec{x}, \vec{X})]_I$ бесконечную последовательность (ω-слово) двоичных наборов

$$string(\langle \sigma_1(x_1), \ldots, \sigma_1(x_k), \sigma_2(X_1), \ldots, \sigma_2(X_m) \rangle).$$

Логика S1S

Таким образом, каждая формула логики S1S $\varphi(x_1, \ldots, x_k, X_1, \ldots, X_m)$ задает ω-язык

$$L(\varphi) = \{[(\vec{x}, \vec{X})]_I : I \models \varphi(x_1, \ldots, x_k, X_1, \ldots, X_m)\}.$$

Логика S1S

Таким образом, каждая формула логики S1S $\varphi(x_1, \dots, x_k, X_1, \dots, X_m)$ задает ω-язык

$$L(\varphi) = \{[(\vec{x}, \vec{X})]_I : I \models \varphi(x_1, \dots, x_k, X_1, \dots, X_m)\}.$$

Например, формула $\varphi(x, X) = X(x)$ задает ω-язык, состоящий из всевозможных бесконечных строк вида

$$\begin{pmatrix} 0 \\ \kappa_0 \end{pmatrix} \begin{pmatrix} 0 \\ \kappa_1 \end{pmatrix} \begin{pmatrix} 0 \\ \kappa_2 \end{pmatrix} \cdots \begin{pmatrix} 0 \\ \kappa_{i-1} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ \kappa_{i+1} \end{pmatrix} \cdots$$

Логика S1S

Язык формул S1S имеет и другое истолкование, непосредственно связанное с задачами информатики.

Логика S1S

Язык формул S1S имеет и другое истолкование, непосредственно связанное с задачами информатики.

Предметная область $\mathbb{N}$ — это шкала дискретного времени, и натуральные числа — это моменты времени (такты работы вычислительной системы).

Логика S1S

Язык формул S1S имеет и другое истолкование, непосредственно связанное с задачами информатики.

Предметная область $\mathbb{N}$ — это шкала дискретного времени, и натуральные числа — это моменты времени (такты работы вычислительной системы).

Каждая переменная-множество — это тип (разновидность) событий, которые могут произойти по ходу вычисления; включение $n \in X$ означает, что событие типа X происходит в момент времени n .

Логика S1S

Интерпретация $I = (\sigma_1, \sigma_2)$ — это модель вычисления информационной системы, в которой оценка предметных переменных σ_1 особо выделяет некоторые такты вычисления, а оценка переменных-множеств σ_2 указывает на каких тактах вычисления осуществляются те или иные события.

Логика S1S

Интерпретация $I = (\sigma_1, \sigma_2)$ — это модель вычисления информационной системы, в которой оценка предметных переменных σ_1 особо выделяет некоторые такты вычисления, а оценка переменных-множеств σ_2 указывает на каких тактах вычисления осуществляются те или иные события.

Таким образом, каждая формула логики S1S $\varphi(X_1, X_2, \ldots, X_m)$ — это формальное описание (спецификация) некоторого класса вычислений, а именно всех таких вычислений I , в которых выполняется данная формула:

$$Comp(\varphi) = \{I : I \models \varphi(X_1, X_2, \ldots, X_m)\}.$$

Логика S1S

Например, формула

$$\begin{aligned}\varphi(X,Y) \;=\; & \forall x \exists z (z > x \wedge X(z)) \wedge \\ & \forall x_1 \forall x_2 (x_1 < x_2 \wedge X(x_1) \wedge X(x_2) \rightarrow \\ & \qquad \exists y (x_1 < y \wedge y < x_2 \wedge Y(y)))\end{aligned}$$

описывает класс вычислений, в которых событие типа X (**запрос клиента**) случается бесконечно часто, и при этом между любыми двумя событиями этого типа происходит хотя бы одно событие типа Y (**отклик сервера**).

Логика S1S

Естественно, возникает вопрос:

Поведение каких вычислительных систем полностью описывается формулами логики S1S?

Логика S1S

Естественно, возникает вопрос:

Поведение каких вычислительных систем полностью описывается формулами логики S1S?

Ответ на этот вопрос был получен еще в 1961 г. в работах Бюхи, Эллгота и Трахтенброта.

ЛОГИКА S1S и ω-АВТОМАТЫ

Теорема 12.1. Для любого автомата Бюхи $\mathcal{B} = (\{0,1\}^k, S, In, Fin, T)$ существует формула логики S1S $\varphi_{\mathcal{B}}(X_1, X_2, \ldots, X_k)$, удовлетворяющая равенству $L(\mathcal{B}) = L(\varphi_{\mathcal{B}}(X_1, X_2, \ldots, X_k))$.

Теорема 12.1. Для любого автомата Бюхи $\mathcal{B} = (\{0,1\}^k, S, In, Fin, T)$ существует формула логики S1S $\varphi_{\mathcal{B}}(X_1, X_2, \dots, X_k)$, удовлетворяющая равенству $L(\mathcal{B}) = L(\varphi_{\mathcal{B}}(X_1, X_2, \dots, X_k))$.

Доказательство. Для простоты обозначений ограничимся рассмотрением случая $k = 1$.

ЛОГИКА S1S и ω-АВТОМАТЫ

Теорема 12.1. Для любого автомата Бюхи $\mathcal{B} = (\{0,1\}^k, S, In, Fin, T)$ существует формула логики S1S $\varphi_{\mathcal{B}}(X_1, X_2, \ldots, X_k)$, удовлетворяющая равенству $L(\mathcal{B}) = L(\varphi_{\mathcal{B}}(X_1, X_2, \ldots, X_k))$.

Доказательство. Для простоты обозначений ограничимся рассмотрением случая $k = 1$.

Пусть $Q = \{s_1, s_2, \ldots, s_m\}$. Введем вспомогательные переменные-множества $Y_1, Y_2, \ldots, Y_m$: каждому состоянию автомата s_ℓ соответствует переменная Y_ℓ , $1 \leq \ell \leq m$, которая служит индикатором этого состояния.

ЛОГИКА S1S и ω-АВТОМАТЫ

Построим формулу $\Phi(X, Y_1, Y_2, \dots, Y_m)$, моделирующую все успешные вычисления автомата $\mathcal{B}$:

$$I \models \Phi(X, Y_1, Y_2, \dots, Y_m)$$

$$\Longleftrightarrow$$

существует успешное вычисление автомата $\mathcal{B}$

$$s_{i_0} \xrightarrow{\sigma_0} s_{i_1} \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_{j-1}} s_{i_j} \xrightarrow{\sigma_j} \cdots ,$$

которое удовлетворяет условиям

1. $[X]_I = \sigma_0\sigma_1 \dots \sigma_{j-1}\sigma_j \dots$,
2. для любого $\ell, 1 \le \ell \le m$, и $j, j \ge 0$
$$i_j \in Y_\ell \;\Leftrightarrow\; s_{i_j} = s_\ell.$$

ЛОГИКА S1S и ω-АВТОМАТЫ

Построим формулу $\Phi(X, Y_1, Y_2, \ldots, Y_m)$, моделирующую все успешные вычисления автомата $\mathcal{B}$:

$$I \models \Phi(X, Y_1, Y_2, \ldots, Y_m)$$

$$\Longleftrightarrow$$

существует успешное вычисление автомата $\mathcal{B}$

$$s_{i_0} \xrightarrow{\sigma_0} s_{i_1} \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_{j-1}} s_{i_j} \xrightarrow{\sigma_j} \cdots ,$$

которое удовлетворяет условиям

1. $[X]_I = \sigma_0\sigma_1 \ldots \sigma_{j-1}\sigma_j \ldots$,
2. для любого $\ell, 1 \le \ell \le m$, и $j, j \ge 0$
$$i_j \in Y_\ell \;\Leftrightarrow\; s_{i_j} = s_\ell.$$

Очевидно, что тогда

$$\varphi_{\mathcal{B}}(X) = \exists Y_1 \ldots \exists Y_m \Phi(X, Y_1, Y_2, \ldots, Y_m).$$

ЛОГИКА S1S и ω-АВТОМАТЫ

Формула $\Phi(X, Y_1, Y_2, \dots, Y_m)$, моделирующая работу автомата $\mathcal{B}$, устроена так:

$$\Phi = \Phi_0 \wedge \Phi_1 \wedge \Phi_2 \wedge \Phi_3 \ ,$$

где каждая из подформул обеспечивает

1. Φ_0 — корректность индикации состояний автомата переменными $Y_i, 1 \leq i \leq m,$
2. Φ_1 — условие инициализации автомата,
3. Φ_2 — корректность переходов автомата,
4. Φ_3 — условие допустимости вычисления.

Рассмотрим каждую из подформул по отдельности.

ЛОГИКА S1S и ω-АВТОМАТЫ

$$\Phi_0 = \forall t \left(\left(\bigvee_{\ell=1}^{m} Y_\ell(t) \right) \wedge \bigwedge_{1 \le \ell_1 < \ell_2 \le m} (\neg Y_{\ell_1}(t) \vee \neg Y_{\ell_2}(t)) \right)$$

ЛОГИКА S1S и ω-АВТОМАТЫ

$$\Phi_0 = \forall t\ ((\bigvee_{\ell=1}^{m} Y_\ell(t)) \wedge \bigwedge_{1\le\ell_1<\ell_2\le m} (\neg Y_{\ell_1}(t) \vee \neg Y_{\ell_2}(t)))$$

Подформула Φ_0 означает, что в каждый момент времени автомат

- находится в каком-то состоянии, и
- не может находится в двух разных состояниях.

ЛОГИКА S1S и ω-АВТОМАТЫ

$$\Phi_0 = \forall t \; ((\bigvee_{\ell=1}^{m} Y_\ell(t)) \wedge \bigwedge_{1 \le \ell_1 < \ell_2 \le m} (\neg Y_{\ell_1}(t) \vee \neg Y_{\ell_2}(t)))$$

Подформула Φ_0 означает, что в каждый момент времени автомат

- находится в каком-то состоянии, и
- не может находится в двух разных состояниях.

$$\Phi_1 = \forall t \; \left(t = 0 \rightarrow \bigvee_{s_\ell \in In} Y_\ell(t)\right)$$

ЛОГИКА S1S и ω-АВТОМАТЫ

$$\Phi_0 = \forall t\ \Big(\big(\bigvee_{\ell=1}^{m} Y_\ell(t)\big) \wedge \bigwedge_{1\le\ell_1<\ell_2\le m} (\neg Y_{\ell_1}(t) \vee \neg Y_{\ell_2}(t))\Big)$$

Подформула Φ_0 означает, что в каждый момент времени автомат

- находится в каком-то состоянии, и
- не может находится в двух разных состояниях.

$$\Phi_1 = \forall t\ \Big(t = 0 \rightarrow \bigvee_{s_\ell \in In} Y_\ell(t)\Big)$$

Подформула Φ_1 означает, что в начальный момент времени автомат находится в одном из начальных состояний.

ЛОГИКА S1S и ω-АВТОМАТЫ

$$\Phi_2 = \forall t \ \Big(\bigwedge_{(\ell_1,\sigma,\ell_2)\in T} \big(Y_{\ell_1}(t) \land X^{\sigma}(t) \to Y_{\ell_2}(s(t))\big)\Big)$$

ЛОГИКА S1S и ω-АВТОМАТЫ

$$\Phi_2 = \forall t \ (\bigwedge_{(\ell_1,\sigma,\ell_2)\in T} (Y_{\ell_1}(t) \wedge X^\sigma(t) \to Y_{\ell_2}(s(t))))$$

Подформула Φ_2 означает, что в каждый следующий момент времени автомат обязательно будет находиться в одном из тех состояний, в которые возможен переход из предыдущего состояния.

Здесь запись $X^\sigma(t)$ обозначает

формулу $\neg X(t)$, если $\sigma = 0$;

формулу $X(t)$, если $\sigma = 1$.

ЛОГИКА S1S и ω-АВТОМАТЫ

$$\Phi_3 = \forall t\ \exists t' \Big((t < t') \wedge \bigvee_{s_\ell \in Fin} Y_\ell(t') \Big)$$

ЛОГИКА S1S и ω-АВТОМАТЫ

$$\Phi_3 = \forall t\ \exists t' \left((t < t') \wedge \bigvee_{s_\ell \in Fin} Y_\ell(t') \right)$$

Подформула Φ_3 означает, что в автомат бесконечно часто пребывает в финальных состояниях.

ЛОГИКА S1S и ω-АВТОМАТЫ

$$\Phi_3 = \forall t\ \exists t' ((t < t') \wedge \bigvee_{s_\ell \in Fin} Y_\ell(t'))$$

Подформула Φ_3 означает, что в автомат бесконечно часто пребывает в финальных состояниях.

Из описания формулы $\Phi(X, Y_1, Y_2, \ldots, Y_m)$ видно, что ее моделями являются все успешные вычисления автомата $\mathcal{B}$. Поэтому для формулы

$$\varphi_{\mathcal{B}}(X) = \exists Y_1 \ldots \exists Y_m \Phi(X, Y_1, Y_2, \ldots, Y_m)$$

справедливо соотношение

$$I \models \varphi_{\mathcal{B}}(X) \ \Leftrightarrow \ [X]_I \in L(\mathcal{B}).$$

QED

ЛОГИКА S1S и ω-АВТОМАТЫ

Теорема 12.2. Для любой формулы логики S1S

$$\varphi(x_1, \ldots, x_m, X_1, X_2, \ldots, X_m),$$

зависящей от свободных предметных переменных $x_1, \ldots, x_m$ и свободных переменных-множеств $X_1, \ldots, X_n$, существует автомат Маллера $\mathcal{B}_\varphi = (\{0,1\}^{n+m}, S, In, T, ACC)$, удовлетворяющий равенству

$$L(\mathcal{B})_\varphi = L(\varphi(x_1, \ldots, x_m, X_1, X_2, \ldots, X_m)).$$

ЛОГИКА S1S и ω-АВТОМАТЫ

Теорема 12.2. Для любой формулы логики S1S

$$\varphi(x_1, \dots, x_m, X_1, X_2, \dots, X_m),$$

зависящей от свободных предметных переменных $x_1, \dots, x_m$ и свободных переменных-множеств $X_1, \dots, X_n$, существует автомат Маллера $\mathcal{B}_\varphi = (\{0,1\}^{n+m}, S, In, T, ACC)$, удовлетворяющий равенству

$$L(\mathcal{B})_\varphi = L(\varphi(x_1, \dots, x_m, X_1, X_2, \dots, X_m)).$$

Доказательство. Построение автомата $\mathcal{B}_\varphi$ проведем индукцией по структуре формулы φ , используя средства композиции автоматов.

ЛОГИКА S1S и ω-АВТОМАТЫ

Атомарные формулы $\psi(x, X) = X(\underbrace{s \ldots s}_{k \text{ раз}}(x))$.

ЛОГИКА S1S и ω-АВТОМАТЫ

Атомарные формулы $\psi(x, X) = X(\underbrace{s \dots s}_{k \text{ раз}}(x))$.

Детерминированный автомат Бюхи $\mathcal{B}_\psi$ прочитывает бесконечную последовательность пар

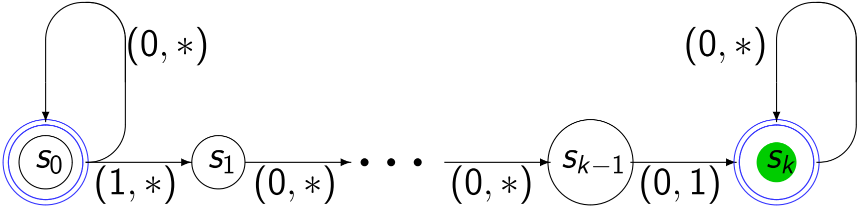
$$(\sigma_1^1, \sigma_1^2), (\sigma_2^1, \sigma_2^2), \dots, (\sigma_i^1, \sigma_i^2), \dots$$

и следит за тем, чтобы

1. только для одной пары (σ_i^1, σ_i^2) верно равенство $\sigma_i^1 = 1$, и при этом
2. для пары $(\sigma_{i+k}^1, \sigma_{i+k}^2)$ верно равенство $\sigma_{i+k}^2 = 1$.

ЛОГИКА S1S и ω-АВТОМАТЫ

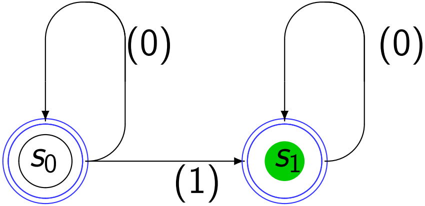
Атомарные формулы $\psi(x, X) = X(\underbrace{s \ldots s}_{k\ \text{раз}}(x))$.



ЛОГИКА S1S и ω-АВТОМАТЫ

Формула $Single(X)$.

$B_{Single(X)}$



ЛОГИКА S1S и ω-АВТОМАТЫ

Формулы $\psi = \psi_1(\vec{x}, \vec{X}) \wedge \psi_2(\vec{x}, \vec{X})$.

ЛОГИКА S1S и ω-АВТОМАТЫ

Формулы $\psi = \psi_1(\vec{x}, \vec{X}) \wedge \psi_2(\vec{x}, \vec{X})$.

Предположим, что построены детерминированные автоматы Маллера $\mathcal{B}_1$ и $\mathcal{B}_2$, соответствующие формулам ψ_1 и ψ_2 .

Подобно тому, как это было сделано для автоматов Бюхи (см. Утверждение 11.3), можно построить такой детерминированный автомат Маллера $\mathcal{B}_\cap$, для которого верно равенство

$$L(\mathcal{B}_\cap) = L(\mathcal{B}_1) \cap L(\mathcal{B}_2).$$

Постройте автомат $\mathcal{B}_\cap$ самостоятельно .

Нетрудно видеть, что $L(\mathcal{B}_\cap) = L(\psi)$.

ЛОГИКА S1S и ω-АВТОМАТЫ

Формулы $\psi = \neg\psi_1(x_1, \ldots, x_n, \vec{X})$.

Формулы $\psi = \neg\psi_1(x_1, \ldots, x_n, \vec{X})$.

Допустим, что построен детерминированный автомат Маллера $\mathcal{B}_1$, соответствующий формуле ψ_1 .

Тогда на основании Теоремы 11.9 можно построить такой детерминированный автомат Маллера $\mathcal{B}_\neg$, для которого верно равенство

$$L(\mathcal{B}_\neg) = \overline{L(\mathcal{B}_1)}.$$

Нетрудно видеть, что

$$L(\psi) = L(\mathcal{B}_\neg) \cap \bigcap_{i=1}^{n} L(\mathcal{B}_{Single(x_i)}).$$

ЛОГИКА S1S и ω-АВТОМАТЫ

Формулы $\psi = \exists X_0 \psi_1(\vec{x}, X_0, \vec{X})$.

ЛОГИКА S1S и ω-АВТОМАТЫ

Формулы $\psi = \exists X_0 \psi_1(\vec{x}, X_0, \vec{X})$.

Допустим, что построен детерминированный автомат Маллера $\mathcal{B}_1$, соответствующий формуле $\psi_1(\vec{x}, X_0, \vec{X})$.

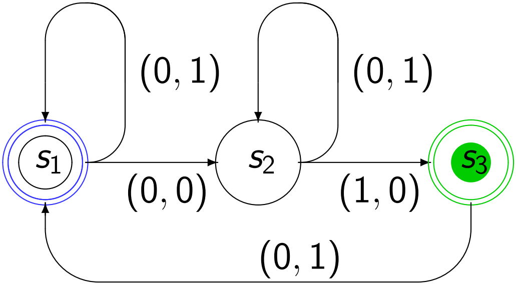
Для построения автомата Маллера $\mathcal{B}_\psi$ достаточно изменить в автомате $\mathcal{B}_1$ отношение переходов:

$$(s, (\sigma_1, \dots, \sigma_n, \delta_1, \dots, \delta_m), s') \in T_\psi \Leftrightarrow \\ \exists \delta_0 : (s, (\sigma_1, \dots, \sigma_n, \delta_0, \delta_1, \dots, \delta_m), s') \in T_1.$$

Нетрудно видеть, что $L(\mathcal{B}_\psi) = L(\psi)$.

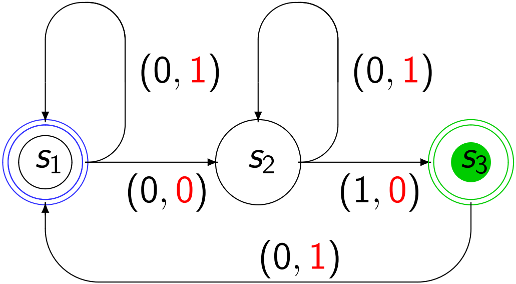
АВТОМАТЫ БЮХИ

Автомат Бюхи $\mathcal{B}$ для формулы $\psi(X_1, X_2)$



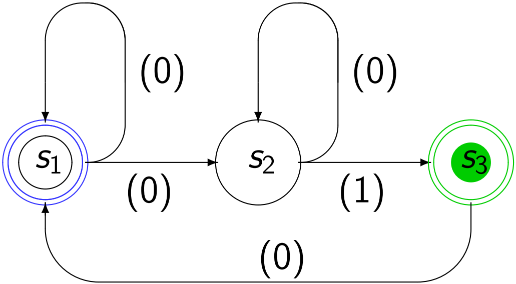
Автоматы Бюхи

Автомат Бюхи $\mathcal{B}'$ для формулы $\exists X_2 \psi(X_1, X_2)$



Автоматы Бюхи

Автомат Бюхи $\mathcal{B}'$ для формулы $\exists X_2 \psi(X_1, X_2)$



ЛОГИКА S1S и ω-АВТОМАТЫ

Формулы $\psi = \exists x_0 \psi_1(x_0, \vec{x}, \vec{X})$.

Формулы $\psi = \exists x_0 \psi_1(x_0, \vec{x}, \vec{X})$.

Поступаем так же, как и в предыдущем случае.
Но здесь нужно принять во внимание, что x_0 — предметная переменная, и ее значением является натуральное число, а не множество чисел.

Поэтому приходится вместо $\psi_1(x_0, \vec{x}, \vec{X})$ использовать равносильную формулу

$$\psi_1'(x_0, \vec{x}, \vec{X}) = \exists X_0 (Single(X_0) \wedge X_0(x_0) \wedge \psi_1(x_0, \vec{x}, \vec{X})).$$

QED

ЛОГИКА S1S и ω-АВТОМАТЫ

Следствие. Проблема выполнимости формул логики S1S разрешима.

Следствие. Проблема выполнимости формул логики S1S разрешима.

Доказательство. Формула φ выполнима $\iff$ существует такая интерпретация I , что $I \models \varphi$ $\iff$ $L(\varphi) \neq \varnothing$.

Последнее соотношение можно проверить, построив, следуя доказательству Теоремы 12.2, автомат Маллера $\mathcal{B}_\varphi$ и проверив, воспользовавшись Теоремой 11.7, пустоту языка $L(\mathcal{B}_\varphi)$. QED

ЛОГИКА S1S и ω-АВТОМАТЫ

Следствие. Проблема выполнимости формул логики S1S разрешима.

Доказательство. Формула φ выполнима $\iff$ существует такая интерпретация I , что $I \models \varphi$ $\iff L(\varphi) \neq \varnothing$.

Последнее соотношение можно проверить, построив, следуя доказательству Теоремы 12.2, автомат Маллера $\mathcal{B}_\varphi$ и проверив, воспользовавшись Теоремой 11.7, пустоту языка $L(\mathcal{B}_\varphi)$. QED

Но какова сложность разрешающего алгоритма?

ЛОГИКА S1S и ω-АВТОМАТЫ

Предположим, что $\varphi = \neg\exists X \neg\exists Y \psi$ и автомат $\mathcal{B}_\psi$ имеет размер n .

Тогда согласно теореме Сафры размер автомата $\mathcal{B}_{\neg\exists\psi}$ может иметь размер $2^{O(n \log n)}$, а автомат $\mathcal{B}_\varphi$ может иметь размер, превышающий величину 2^{2^n}

ЛОГИКА S1S и ω-АВТОМАТЫ

Предположим, что $\varphi = \neg\exists X \neg\exists Y \psi$ и автомат $\mathcal{B}_\psi$ имеет размер n .

Тогда согласно теореме Сафры размер автомата $\mathcal{B}_{\neg\exists\psi}$ может иметь размер $2^{O(n \log n)}$, а автомат $\mathcal{B}_\varphi$ может иметь размер, превышающий величину 2^{2^n}

Отсюда следует, что размер автомата $\mathcal{B}_\varphi$ для произвольной формулы S1S может иметь размер

$$\left. 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \right\} O(|\varphi|)$$

И эта оценка сложности достижима!!!

ДРУГИЕ ЛОГИКИ 2-ГО ПОРЯДКА

Что дальше?

ДРУГИЕ ЛОГИКИ 2-ГО ПОРЯДКА

Что дальше?

1. Полученные результаты можно распространить на более широкий класс логик 2-го порядка.

S2S — монадическая логика предикатов 2-го порядка с двумя функциями следования $left(x)$ и $right(x)$. Формулы этой логики интерпретируются на бесконечных бинарных деревьях.

М.О. Рабин показал, что класс древесных языков, выражаемых формулами логики S2S, совпадает с классом ω-регулярных древесных языков, а также с классом древесных ω-языков, распознаваемых конечными древесными ω-автоматами.

Другие логики 2-го порядка

Что дальше?

Другие логики 2-го порядка

Что дальше?

2. Полученные результаты можно распространить на более простые интерпретации.

WS1S — слабая монадическая логика предикатов 2-го порядка, в которой значениями переменных-множеств могут быть только **конечные** множества натуральных чисел. Формулы этой логики интерпретируются на **конечных** словах.

Оказывается, что класс языков, выражаемых формулами логики WS1S, совпадает с классом регулярных языков, и, следовательно, с классом языков, распознаваемых конечными автоматами.

ДРУГИЕ ЛОГИКИ 2-ГО ПОРЯДКА

Что дальше?

Другие логики 2-го порядка

Что дальше?

3. Полученные результаты сохраняют справедливость и для более узких классов формул S1S.

LTL — темпоральная логика линейного времени — подкласс S1S, в котором разрешается использовать отношение $\leq$ на множестве натуральных чисел, но не разрешается применять кванторы к переменным-множествам.

Оказывается, что для каждой формулы LTL φ существует такой автомат Бюхи $\mathcal{B}_\varphi$, что $L(\mathcal{B}_\varphi) = L(\varphi)$ и при этом $|\mathcal{B}_\varphi| = O(2^{|\varphi|})$.

КОНЕЦ ЛЕКЦИИ 12