



# Понятийный анализ и контекстная технология

Выхованец В.С.  
Институт проблем управления РАН  
<http://valery.vykhovanets.ru>

# План доклада



## I. Введение:

- проблематика;
- семантический разрыв;
- постановка задачи.



## II. Синтаксис и семантика:

- формальный язык;
- задание синтаксиса;
- описание семантики.



## III. Суть подхода:

- замыкание;
- протоязык;
- онтологический язык;
- проблемный язык;
- интерпретация;
- индукция и аксиома.



## IV. Примеры

(перевод, атрибуты, лифт)



## V. Понятийный анализ:

- основные положения;
- модель понятия;
- исчисление понятий.



## VI. Контекстная технология:

- разнесенный разбор;
- иерархия языков;
- эффективность.
- область применимости.



## VII. Формальный подход:

- образующие алгебры;
- аналитический синтез;
- сложность.



## VIII. Выводы

# Проблематика

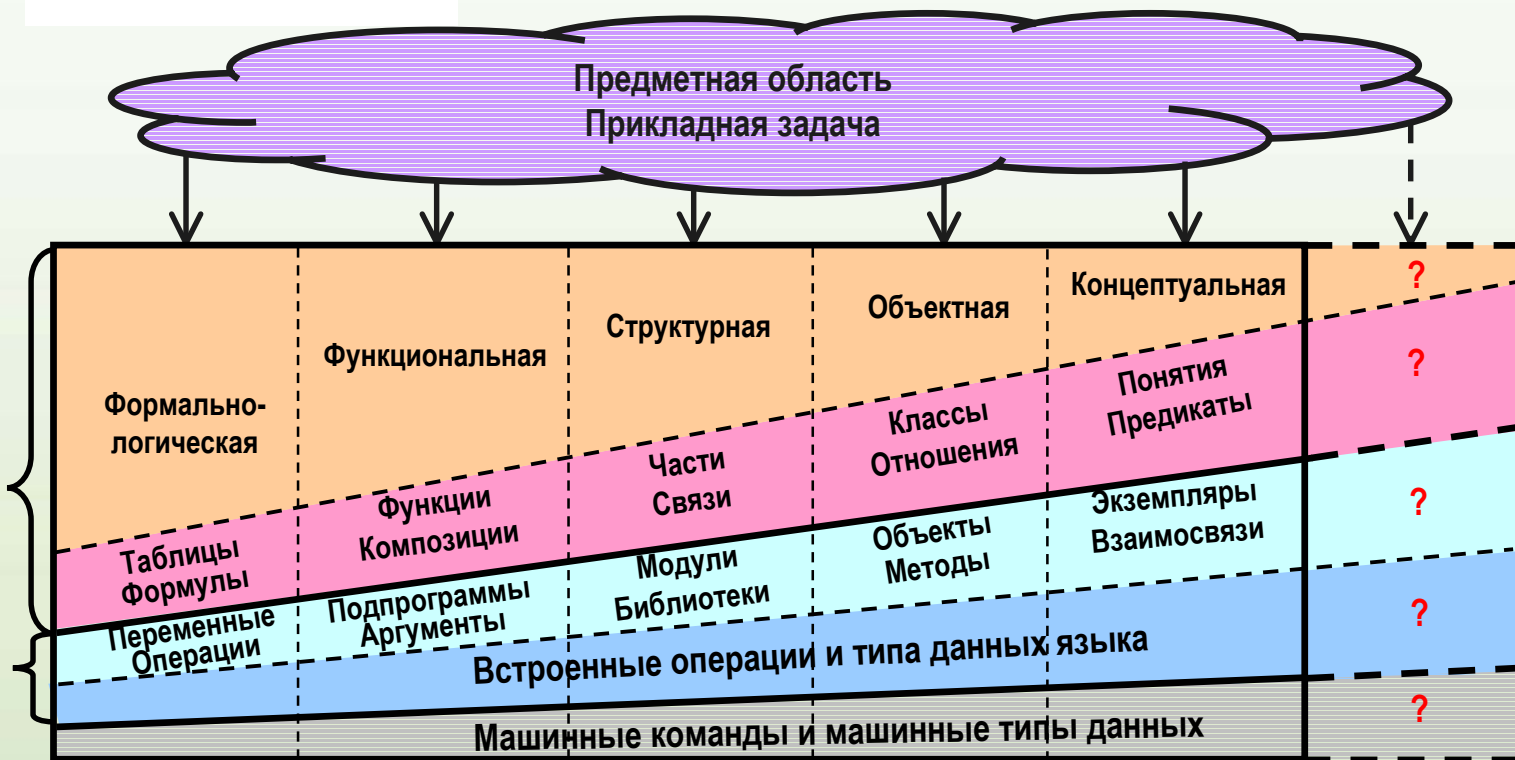
- Высокая сложность разработки
  - Низкая надежность программных средств
  - Недостаточное качество программ
- 
- Увеличенные сроки разработки
  - Превышение бюджета проектов
  - Неудовлетворенные ожидания заказчика
  - Прямой экономический ущерб

# Семантический разрыв

Методология

Технология

Платформа





# Постановка задачи

Дано многоэтажное здание с одним лифтом. На каждом этаже – кнопки для вызова лифта на движение вверх и вниз. В кабине имеется панель с кнопками для перемещения на один из девяти этажей.

Разработать систему управления лифтом.

## 1. Ожидание вызова

Если вызовов нет, то ожидать вызов (1). Если вызов со второго этажа, то перейти к открытию дверей (2), иначе – на принятие решения о движении (4).

## 2. Открытие дверей

Открыть дверь. Если дверь открылась, то перейти на принятие решения о движении (4), иначе повторить открытие двери (2).

## 3. Закрытие дверей

Закрыть дверь. Если дверь не закрылась, то повторить закрытие двери (3), иначе перейти к определению направления движения (4).

## 4. Направление движения

**Продолжение движения.** Если лифт двигался вверх (вниз) и имеются вызовы на движение вверх (вниз), то начать движение вверх (5) (вниз 6). **Изменение направления.** Если вызовов на движение вверх (вниз) нет, но есть вызовы на движение вниз (вверх), то начать движение вниз (6) (вверх 5). **Возврат в начало.** Если вызовов нет и при этом лифт выше (ниже) второго этажа, то начать движение вниз (6) (вверх 5), иначе перейти в состояние ожидания (1).

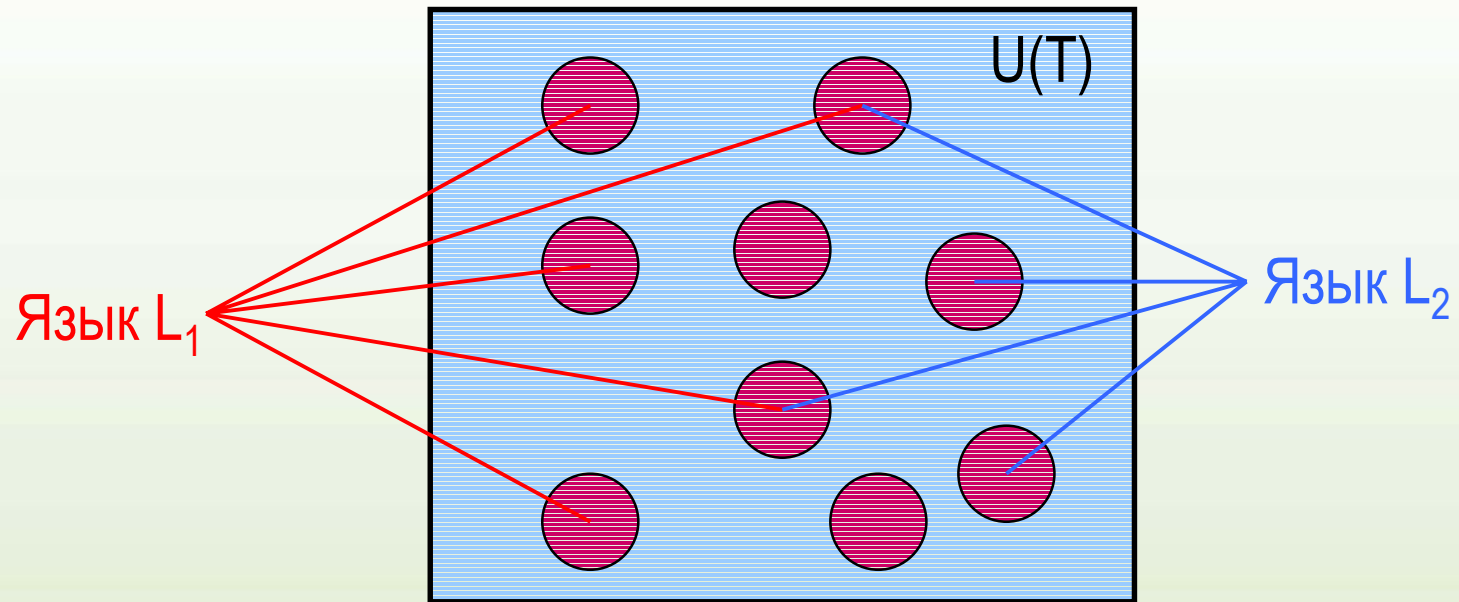
## 5. Движение вверх

Начать движение вверх. Если при проходе этажа имеется вызов на движение вверх, то остановить лифт и открыть дверь (2), иначе продолжить движение вверх (5).

## 6. Движение вниз

Начать движение вниз. Если при проходе этажа имеется вызов для движения вниз, то остановить лифт и открыть дверь (2), иначе продолжить движение вниз (6).

# Формальный язык



$$G(L) = \langle T, N, P, I \rangle$$

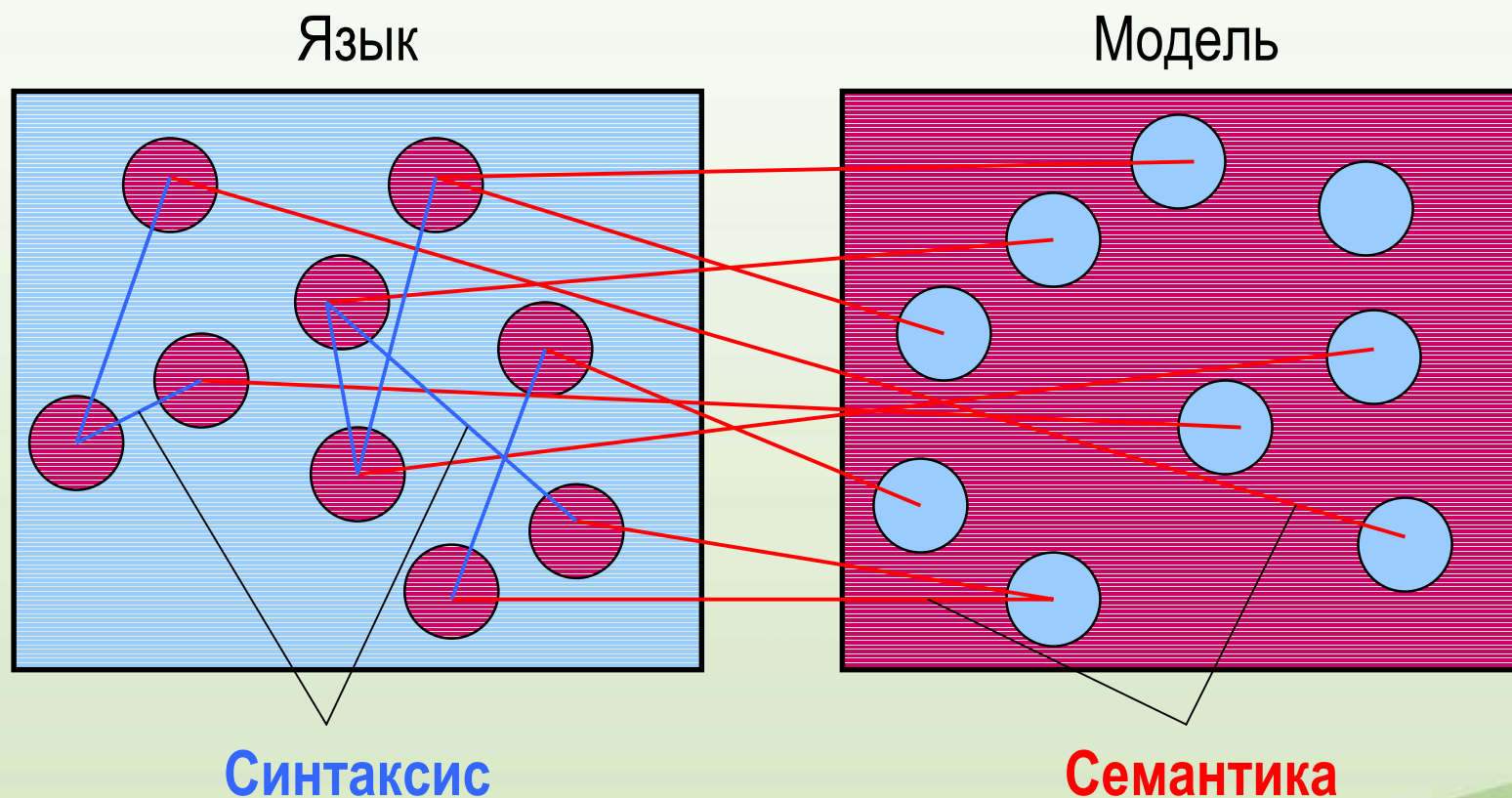
---

$$G(L) = \langle \{0,1\}, \{I, J\}, P, I \rangle,$$
$$P = \{I \rightarrow 0 \mid J, J \rightarrow 1 \mid J0 \mid J1\}$$

---

$$I \rightarrow 0, \quad I \rightarrow J \rightarrow 1, \quad I \rightarrow J \rightarrow J0 \rightarrow 10.$$

# Синтаксис и семантика



## II

### Синтаксис и семантика

Пример



**Перечисление** – сопоставление строк двух языков

Пример



**Аксиоматический** – сопоставление продукциям грамматики формул аксиоматической теории, задаваемой аксиомами и правилами вывода

Пример



**Алгебраический** – сопоставление продукциям грамматики формул алгебраической теории, задаваемой множеством и операциями на нем

Пример



**Денотационный** – сопоставление продукциям грамматики функциональных денотатов (сложных функций)

Пример



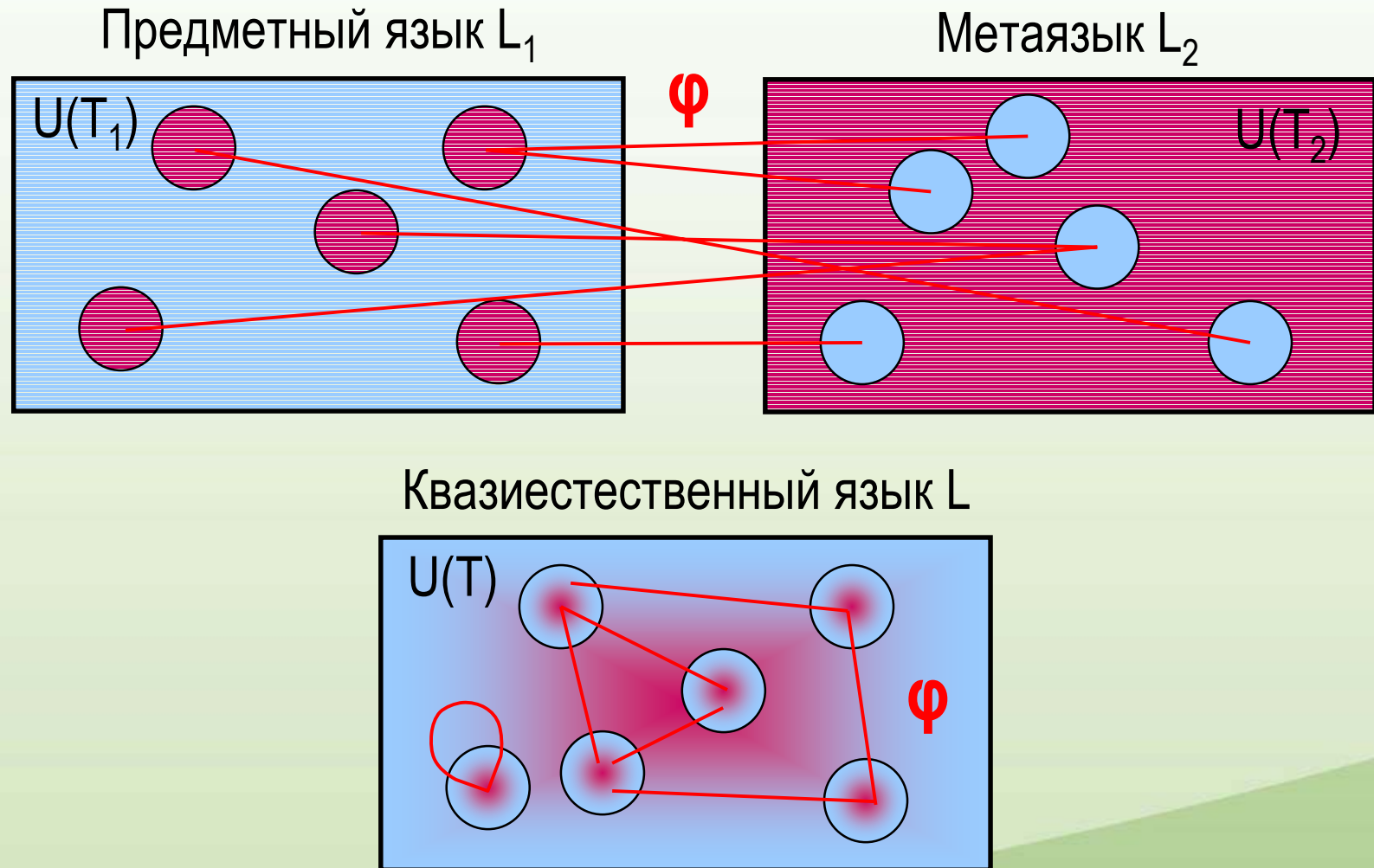
**Операционный** – сопоставление продукциям грамматики последовательности команд (операций)



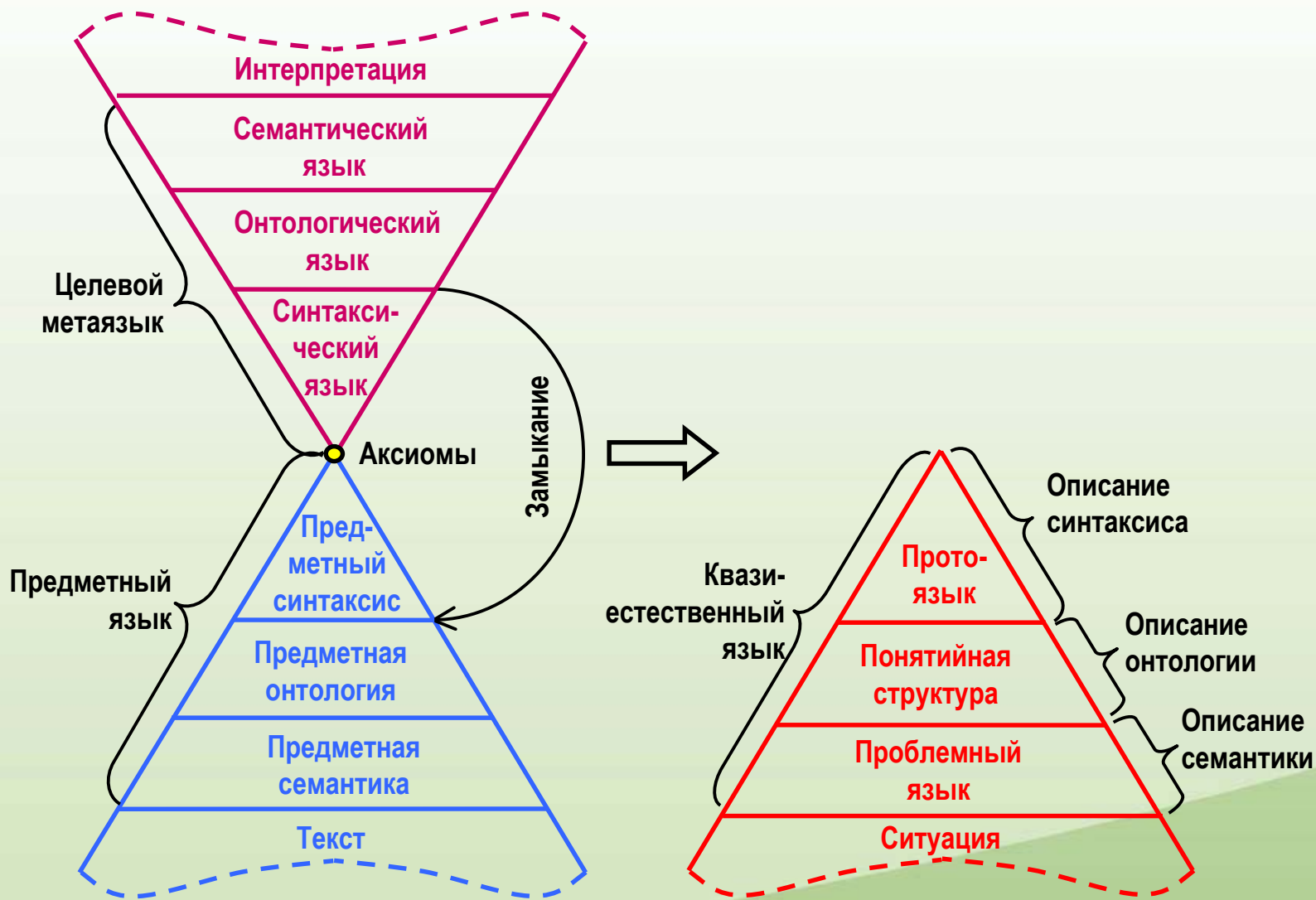
# Особенности описания семантики

- Семантика формального языка – это **интерпретация** его строк с помощью другого языка
- Принцип **композиционности** реализуется путем использования аппарата формальных грамматик
- Фиксируется **конечное множество** семантических категорий, через которые описывается семантика
- Описание семантики осуществляется **формальными** средствами
- Определяется только **одна семантика** (задается только одна интерпретация)

# Суть подхода

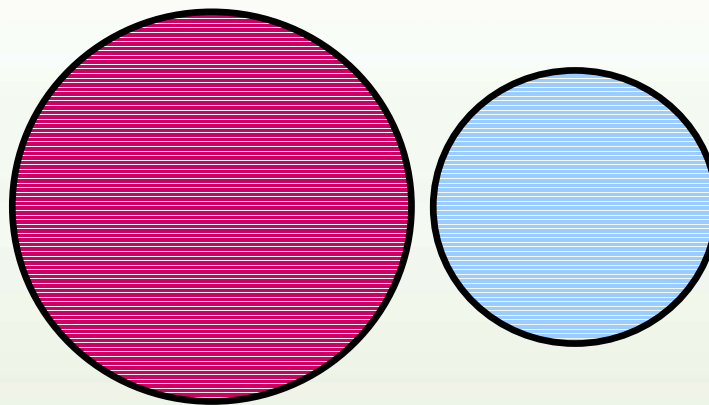


# Семантическое замыкание

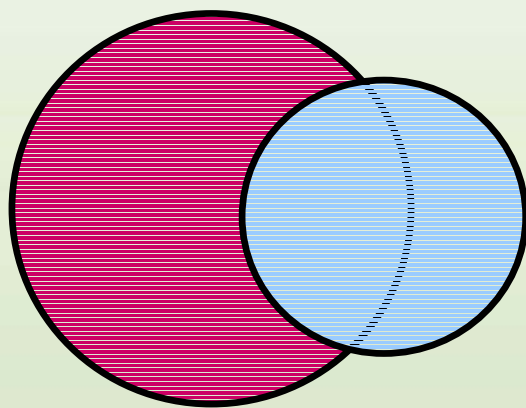




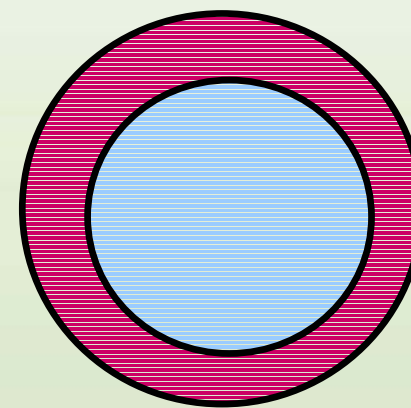
# Пространство признаков



Независимость



Дифференциация



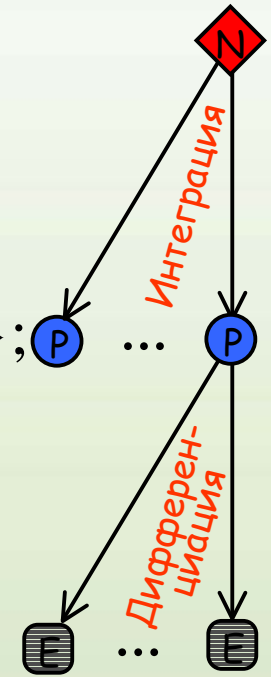
Интеграция

# Формализм понятия

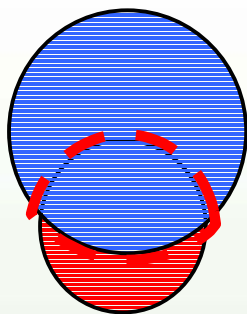
- Понятие** (сложный смысл)  $N = \begin{cases} \text{shm } N = (P^0, P^1, \dots, P^{n-1}); \\ \text{int } N = \{(P_j^0, P_j^1, \dots, P_j^{n-1}) \mid j = 0, 1, \dots\}; \\ \text{ext } N = \{N_0, N_1, \dots, N_{m-1}\}. \end{cases}$

- Признак** (простой смысл)  $P = \begin{cases} \text{shm } P = (P); \\ \text{int } P = \{(P_0), (P_1), \dots, (P_{u-1})\}; \\ \text{ext } P = \{E_0, E_1, \dots, E_{u-1}\}. \end{cases}$

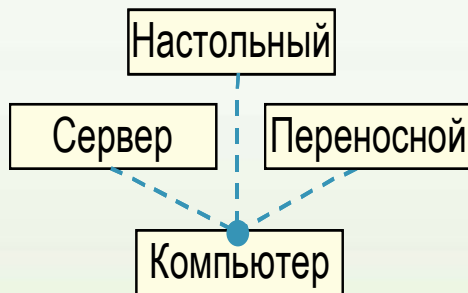
- Сущность** (элементарный смысл)  $E = \begin{cases} \text{shm } E = (E); \\ \text{int } E = \{(E)\}; \\ \text{ext } E = \{E\}. \end{cases}$



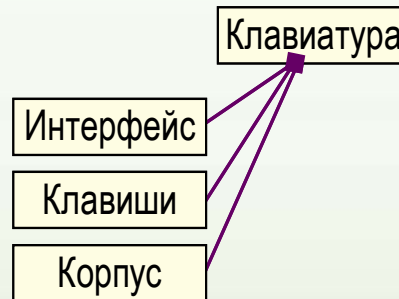
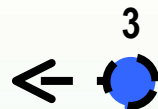
# Абстрагирование



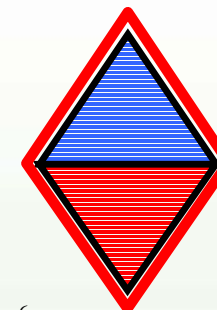
$$\begin{cases} \text{shm } C_G = \bigcap_{j=0}^{m-1} \text{shm } C_j; \\ \text{int } C_G \supseteq \bigcup_{j=0}^{m-1} \text{int } C_j; \\ \text{ext } C_G \supseteq \bigcup_{j=0}^{m-1} \text{ext } C_j. \end{cases}$$



Обобщение



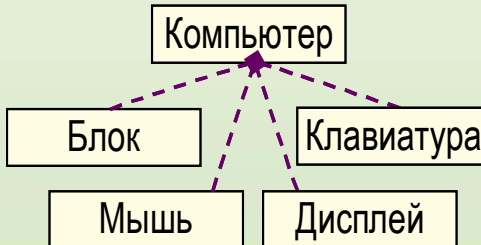
Агрегация



$$\begin{cases} \text{shm } C_A = \bigcup_{j=0}^{m-1} \text{shm } C_j; \\ \text{int } C_A = \times \text{int } C_j; \\ \text{ext } C_A = \times \text{ext } C_j. \end{cases}$$



$$\begin{cases} \text{shm } C_B \supseteq \bigcup_{j=0}^{m-1} \text{shm } C_j; \\ \text{int } C_B \subseteq \times \text{int } C_j; \\ \text{ext } C_B \subseteq \times \text{ext } C_j; \\ \text{link } C_B \subseteq \text{shm } C_B. \end{cases}$$

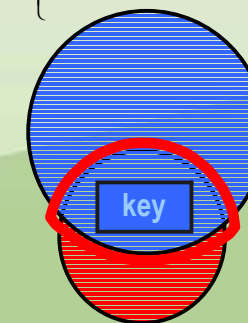
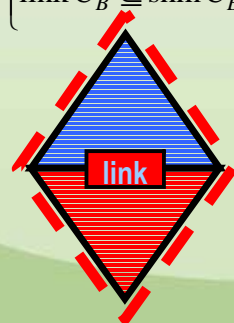


Ассоциация

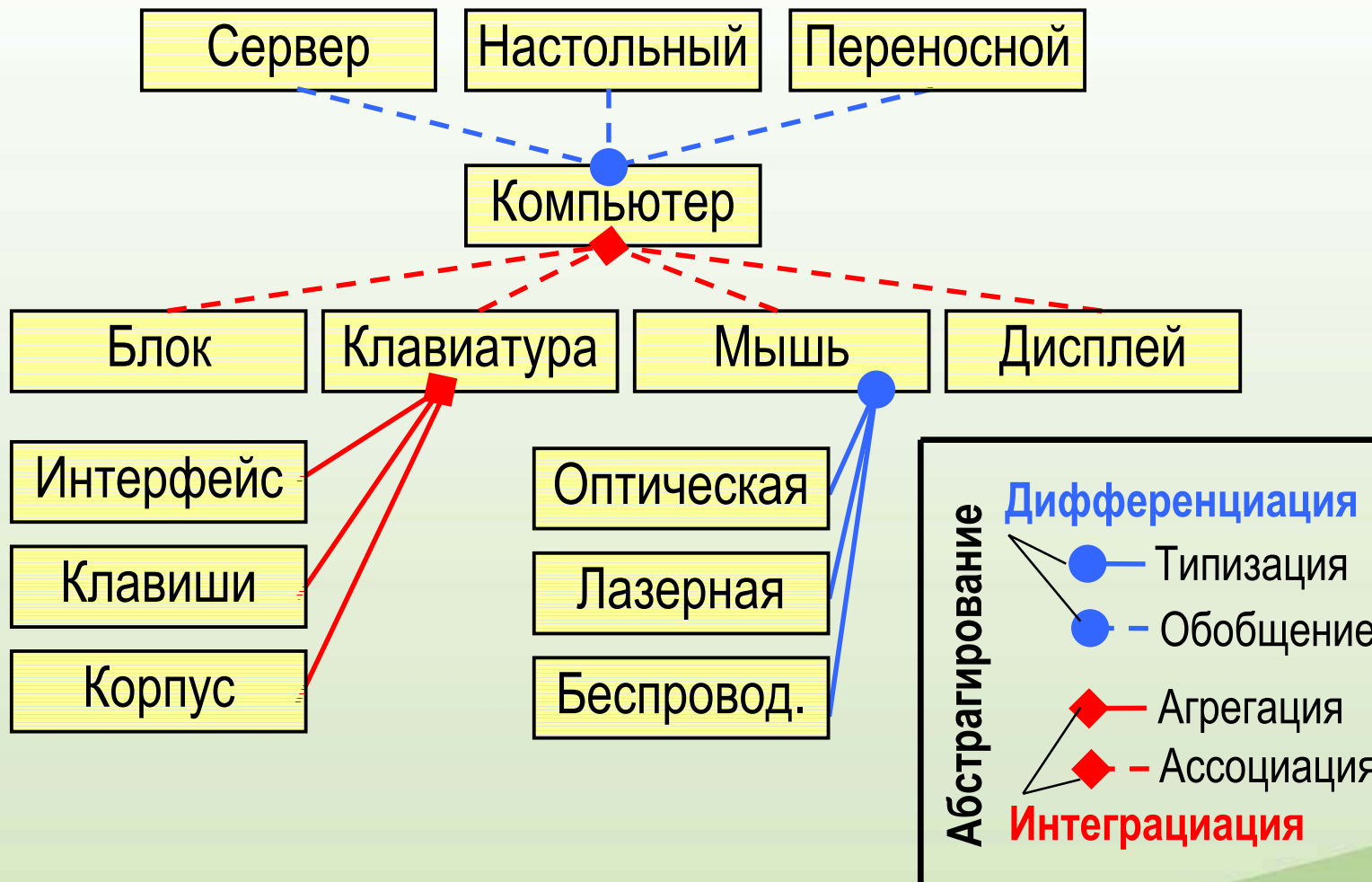
Типизация



$$\begin{cases} \text{shm } C_T = \bigcap_{j=0}^{m-1} \text{shm } C_j; \\ \text{int } C_T = \bigcup_{j=0}^{m-1} \text{int } C_j; \\ \text{ext } C_T = \bigcup_{j=0}^{m-1} \text{ext } C_j; \\ \text{key } C_T \subseteq \text{shm } C_T. \end{cases}$$



# Понятийная структура





# Онтологический язык

model → essences [model]  
 essences → differentiation *notion*  
 integration [intension]  
 differentiation → '(' [notions] ')'  
 integration → '(' [notions] ')'  
 notions → **notion** [notions]



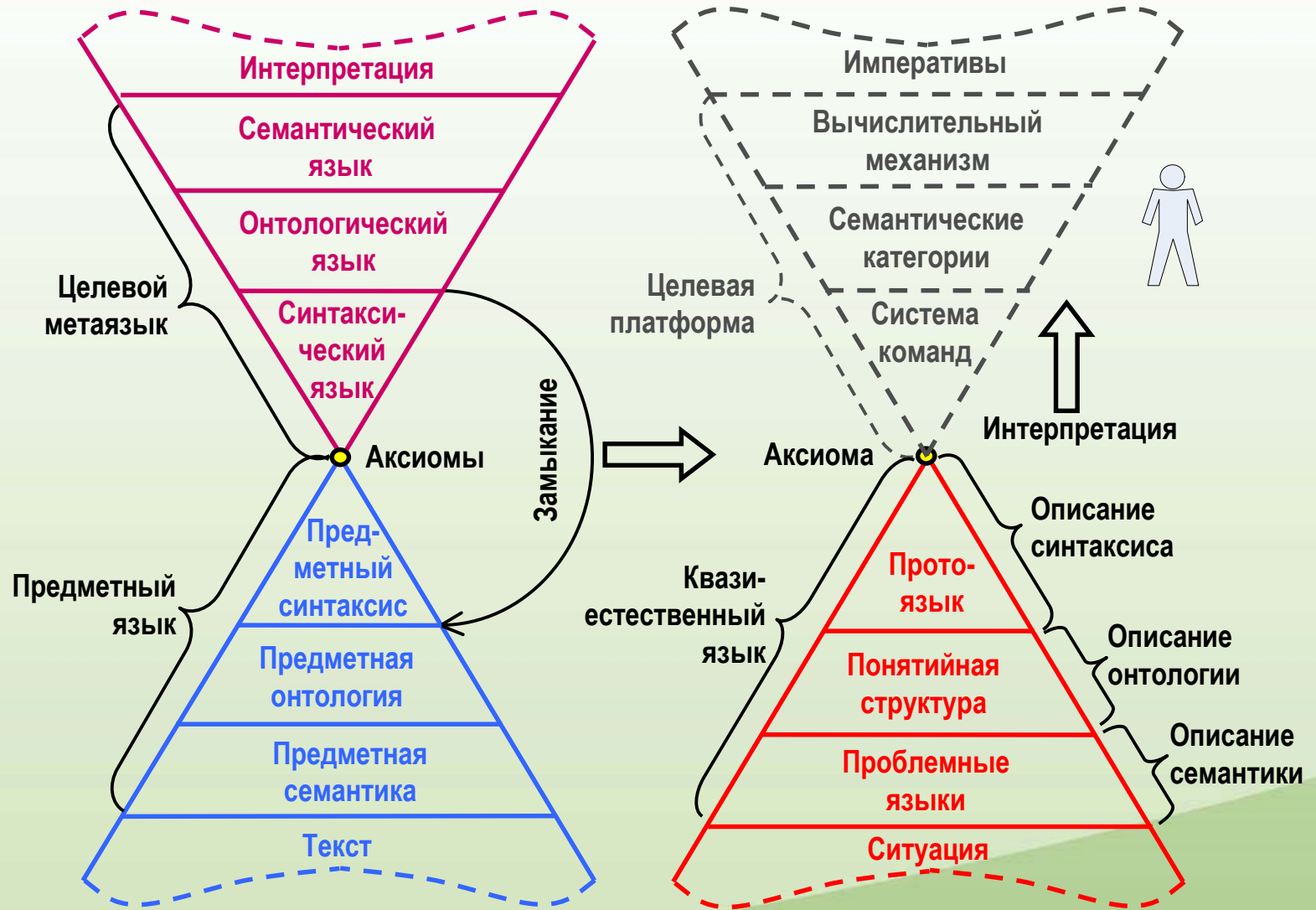
# Проблемный язык

cognition → model [situation] [cognition]  
 model → essences [model]  
 essences → differentiation *notion*  
                   integration [intension]  
 intension → sentence [intension]  
 sentence → syntax semantic  
 syntax → item [syntax]  
 item → **notion** | "'" [*terms*] "'"  
 semantic → pragmatic [semantic]  
 pragmatic → [*aspect*] '{' [text] '}'  
 text → phrase [text]  
 phrase → **terms** | [*aspect*] '{' text '}'  
 situation → [*aspect*] '<' [text] '>'

```

() Boolean ()
'false' {...}
'true' {...}
"[A-Za-z][A-Za-z0-9]*" {...}
'(' Boolean ')' {}
'not' Boolean {...}
Boolean 'and' Boolean {...}
Boolean [a] 'or' Boolean [b]
      { not (not a and not b) }
< (not x or y) and z >
fuzzy < not (x or y) and z >
  
```

# Интерпретация



# Аксиома

«Пустые» квадратные скобки реализуют запись в область императива (кода) того значения, которое задается текущим элементом предложения

```

() ()
  '#' "[0-9A-F][0-9A-F]" [] {}
() Boolean ()
  'not' Boolean
    { #58 #F7 #D0 #50 }
  Boolean 'or' Boolean
    { #58 #5A #0B #C2 #50 }
  Boolean [a] 'imp' [b] Boolean
    { not a or b }

```

```

() ()
  '"' "[0-9A-Za-z\s.,]+" [] '"' {}
() Boolean ()
  'not' Boolean
    { `pop eax` `not eax`
      `push eax` }
  Boolean 'or' Boolean
    { `pop eax` `pop edx`
      `or eax, edx` `push eax` }
  Boolean [a] 'imp' [b] Boolean
    { not a or b }

```

# Семантическая индукция

Определение семантических категорий по мере необходимости, в процессе определения проблемного языка и средствами этого языка:

- **база индукции** – первичные категории, которые непосредственно реализуются целевой вычислительной платформой и декларируются перед использованием;
- **предположение индукции** – все ранее определенные семантические категории, выраженные на проблемном языке;
- **индуктивный переход** – описание семантики нового или уже существующего предложения на уже определенном до этого проблемном подязыке;
- **заключение индукции** – определение новой или доопределение существующей семантической категории.

---

'not' Boolean { #58 #F7 #D0 #50 }

Boolean 'or' Boolean { #58 #5A #0B #C2 #50 }

Boolean [a] 'imp' Boolean [b] { not a or b }

# Управляемый перевод

() *String* ()

"".\*"" { ... }

*String* '&' *String* { ... }

() *Expression* ()

"[0-9]+'" [n] { n } *dif* { '0' }

'x' { 'x' } *dif* { '1' }

'(' *Expression* ')' [e]

{ '(' & e & ')' } *dif* { '(' & *dif* { e } & ')' }

'-' *Expression* [e]

{ '-' & e } *dif* { '-' & *dif* { e } }

*Expression* [e1] '\*' *Expression* [e2]

{ e1 & '\*' & e2 }

*dif* { '(' & e1 & '\*' & *dif* { e2 } & '+' & *dif* { e1 } & '\*' & e2 & ')' }

*Expression* [e1] "+|-" [o] *Expression* [e2]

{ e1 & o & e2 } *dif* { '(' & *dif* { e1 } & o & *dif* { e2 } & ')' }

---

*dif* < -x\*x+5\*(x-3) > → -(x\*1+1\*x)+(5\*(1-0)+0\*(x-3))

*equ*< *dif*{ -x\*x+5\*(x-3) } > → -2\*x+5

# Атрибутные грамматики

(Number) *Integer* ()

"[0-9]" [d]  
{ d }

*Integer* [i] "[0-9]" [d]  
{  $i * 10 + d$  }

(Float) *Fraction* ()

"[0-9]" [d]  
{ d }

"[0-9]" [d] *Fraction* [f]  
{  $d + f / 10$  }

() *Fixed* (Float)

*Integer* [i]  
{ i }

*Integer* [i] '.' *Fraction* [f]  
{  $i + f / 10$  }

() *Expr* (Number)

(Number) *Const* ()  
"[0-9]+" [c]  
{ c }

(Const) *Prim* ()

(' *Expr* [e] ')  
{ e }

(Prim) *Mul* ()

*Prim* [p1] '\*' *Prim* [p2]  
{  $p1 * p2$  }

(Mul) *Add* ()

*Mul* [p1] '+' *Mul* [p2]  
{  $p1 + p2$  }

() *Expr* ()

*Add* [a]  
{ a }

# Управление лифтом



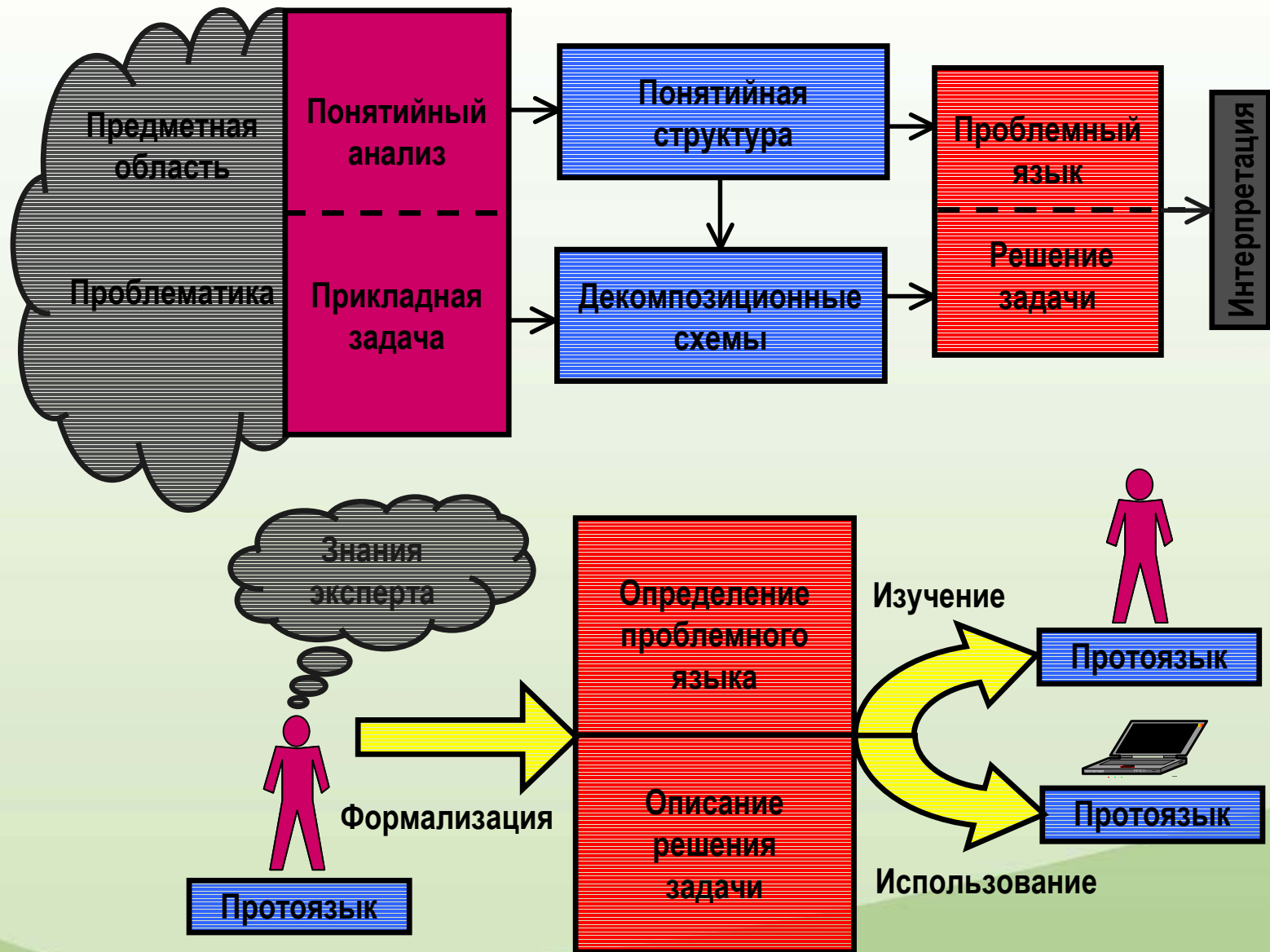
Постановка  
задачи

```
( ) Движение ( )
    "[Дд]вижение" {...}
( ) Дверь ( )
    "[Дд]верь" {...}
( ) Этаж ( )
    'этаж' "[0-9]" {...}
    'этаж' 'вызова' "[вверх|вниз]" {...}
    'этаж' 'лифта' {...}
( ) Вызов (Этаж Движение)
    'вызов' {...}
( ) Лифт (Этаж Движение Дверь)
    'лифт' {...}
( ) Метка ( )
    "[А-Яа-я][А-Яа-я0-9]" {...}
( ) Переход ( )
    Метка {...}
( ) Логическое ( )
    Этаж "выше|ниже|равен" Этаж {...}
    Вызов "вверх|вниз|нет" {...}
    Дверь "открыта|закрыта" {...}
    'не' Логическое { }
( ) ( )
    ':' {...}
    Метка ':' {...}
    Движение "вверх|вниз|останов" {...}
    Дверь "открыть|закрыть" {...}
    'Если' Логическое ', 'то' Переход {...}
    'Если' Логическое ', 'то' Переход ', '
    'иначе' Переход {...}
```

```
Ожидание:
    Если вызов нет, то Ожидание.
    Если этаж вызова равен этаж 2,
        то Открыть, иначе Решение.
Открыть:
    Дверь открыть.
    Если дверь открыта, то Решение,
        иначе Открыть.
Закрыть:
    Дверь закрыть.
    Если дверь закрыта, то Решение,
        иначе Закрыть.
Решение:
    Если лифт вверх и вызов вверх, то Вверх.
    Если лифт вниз и Вызов вниз, то Вниз.
    Если не вызов вверх и вызов вниз, то Вниз.
    Если не вызов вниз и вызов вверх, то Вверх.
    Если вызов нет и этаж лифта выше этаж 2,
        то Вниз.
    Если вызов нет и этаж лифта ниже этаж 2,
        то Вверх, иначе Ожидание.
Вверх:
    Движение вверх.
    Если этаж лифта равен этаж вызова вверх,
        то Открыть, иначе Вверх.
Вниз:
    Движение вниз.
    Если этаж лифта равен этаж вызова вниз,
        то Открыть, иначе Вниз.
```



# Основные положения



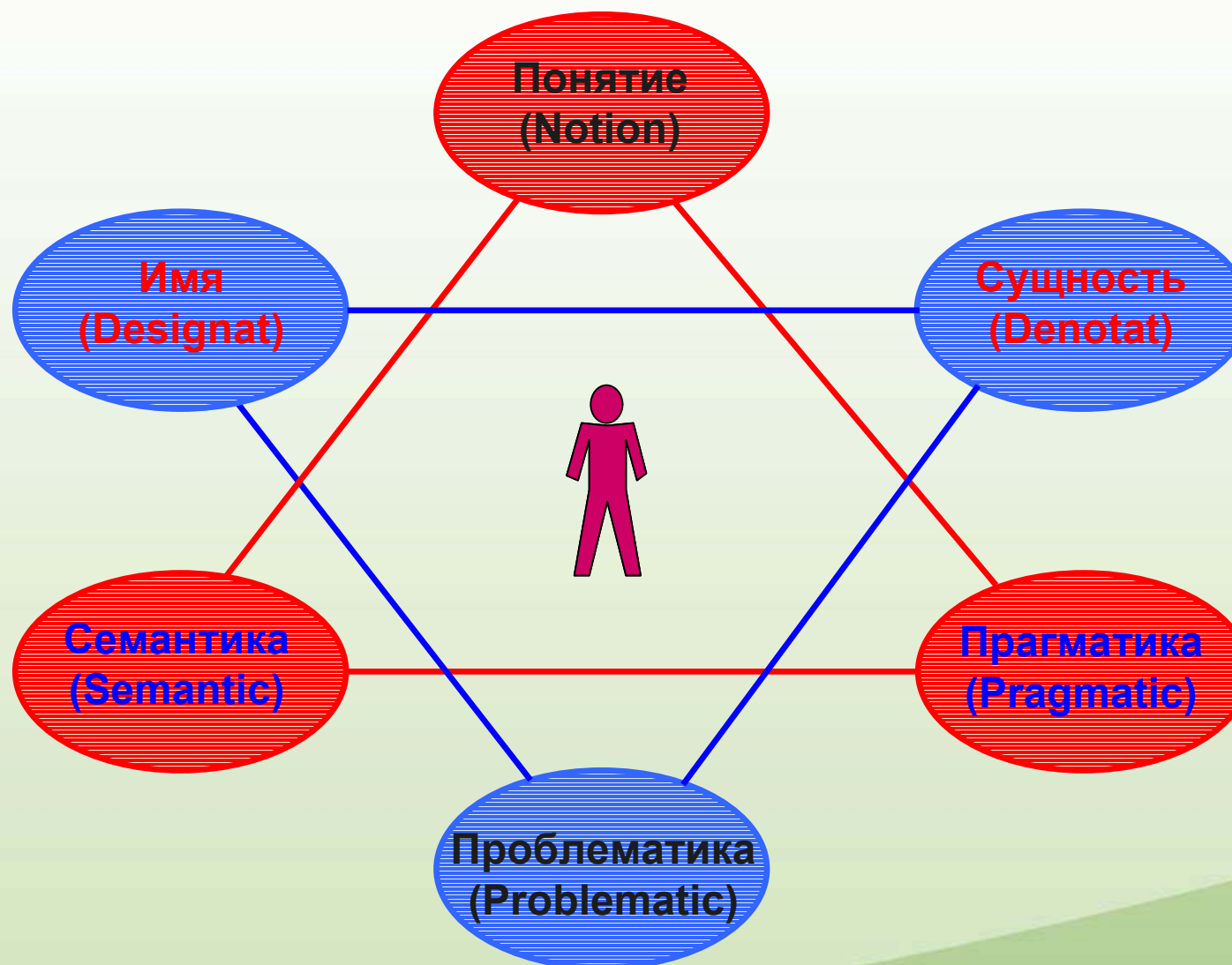


Пример  
понятийного  
анализа

# Понятийный анализ

- **Разделение** сущностей предметной области на сигнификативные и денотационные;
- **Означивание** сигнификативных сущностей и выявление существенных признаков у денотационных;
- **Сопоставление** денотационных сущностей и определение их общих и различающихся признаков;
- **Образование** новых или определение уже существующих понятий на основе интеграции и дифференциации признаков;
- **Создание** понятийной структуры предметной области путем описания отображений одних понятий на другие;
- **Уточнение** способа абстрагирования понятий (обобщения или типизации, ассоциации или агрегации);
- **Вычисление** схем понятий и задание ключей – для типизации, связей – для ассоциации.

# Модель понятия



## Исчисление понятий

- **Алфавит:**
  - $N, N_1, N_2, \dots$ ;
  - операции  $\neg, \Pi, \sqcup$ ;
  - отношения  $\supset, \supseteq$ ;
  - пунктуация  $(, ), \_$ .
- **Аксиоматика:**
  - пустое понятие.
- **Правила вывода:**
  - простых понятий ( $\Pi$ );
  - абстрактных понятий ( $O, A$ );
  - сложных понятий ( $C$ ).

$$\Pi \frac{()() \neg^N ()N()}{()} \vdash \frac{()N()}{(N)}$$

$$O \frac{(\dots)N_1(\dots) \dots (\dots)N_m(\dots)}{\text{shm } N_1 \dots \text{shm } N_m} \neg^N \vdash \frac{(N_1 \dots N_m)N()}{\prod_{i=1}^m \text{shm } N_i \supset ()} \frac{m}{\prod_{i=1}^m \text{shm } N_i}$$

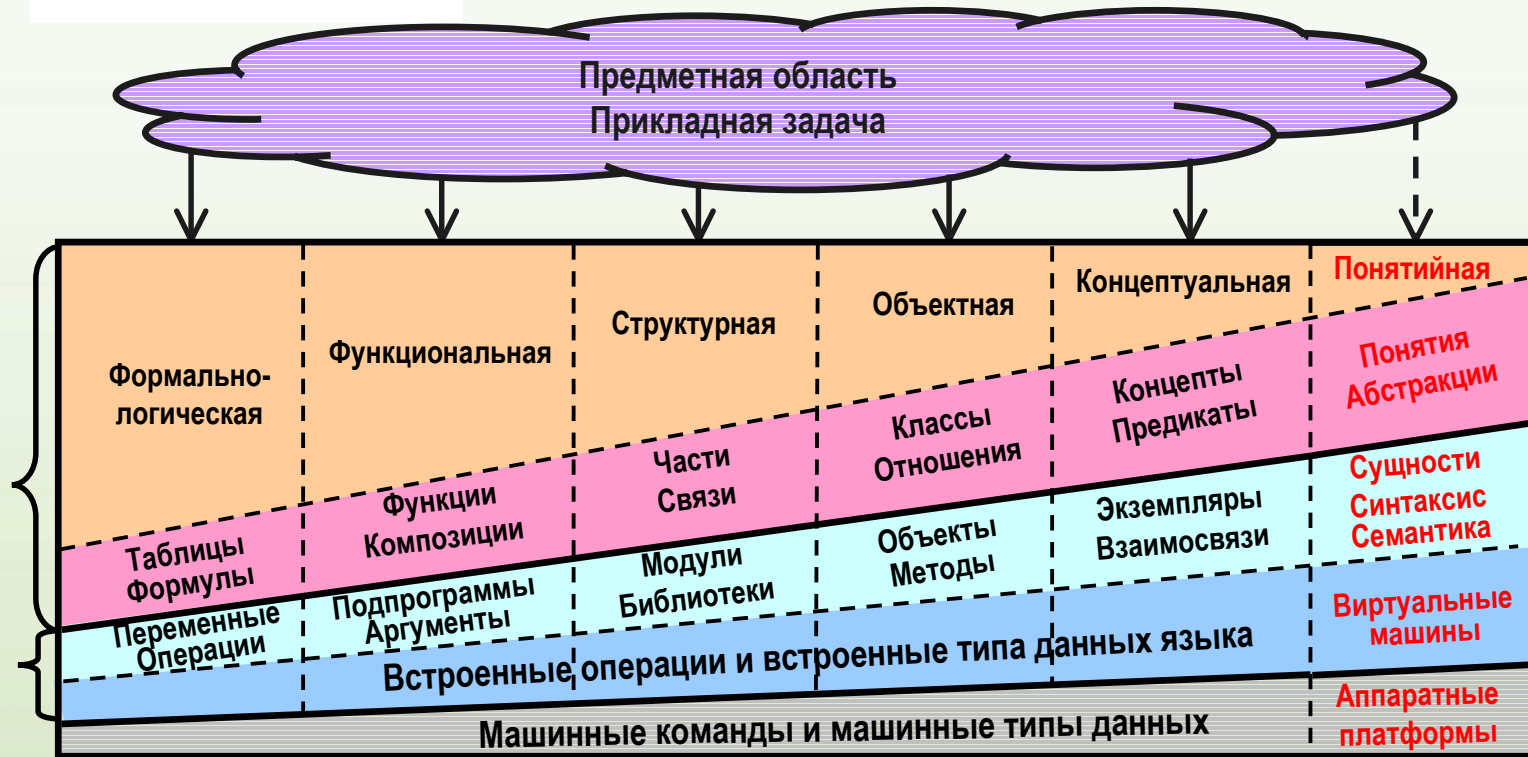
$$A \frac{(\dots)N_1(\dots) \dots (\dots)N_m(\dots)}{\text{shm } N_1 \dots \text{shm } N_m} \neg^N \vdash \frac{()N(N_1 \dots N_m)}{\prod_{i=1}^m \text{shm } N_i \supset ()} \frac{m}{\prod_{i=1}^m \text{shm } N_i}$$

$$C \frac{(\dots)N_1(\dots) \dots (\dots)N_m(\dots)}{\text{shm } N_1 \dots \text{shm } N_m} \neg^N \vdash \frac{(N_1 \dots N_t)N(N_{t+1} \dots N_m)}{\prod_{i=1}^t \text{shm } N_i \supseteq \prod_{j=t+1}^m \text{shm } N_j} \frac{m}{\prod_{j=t+1}^m \text{shm } N_j}$$



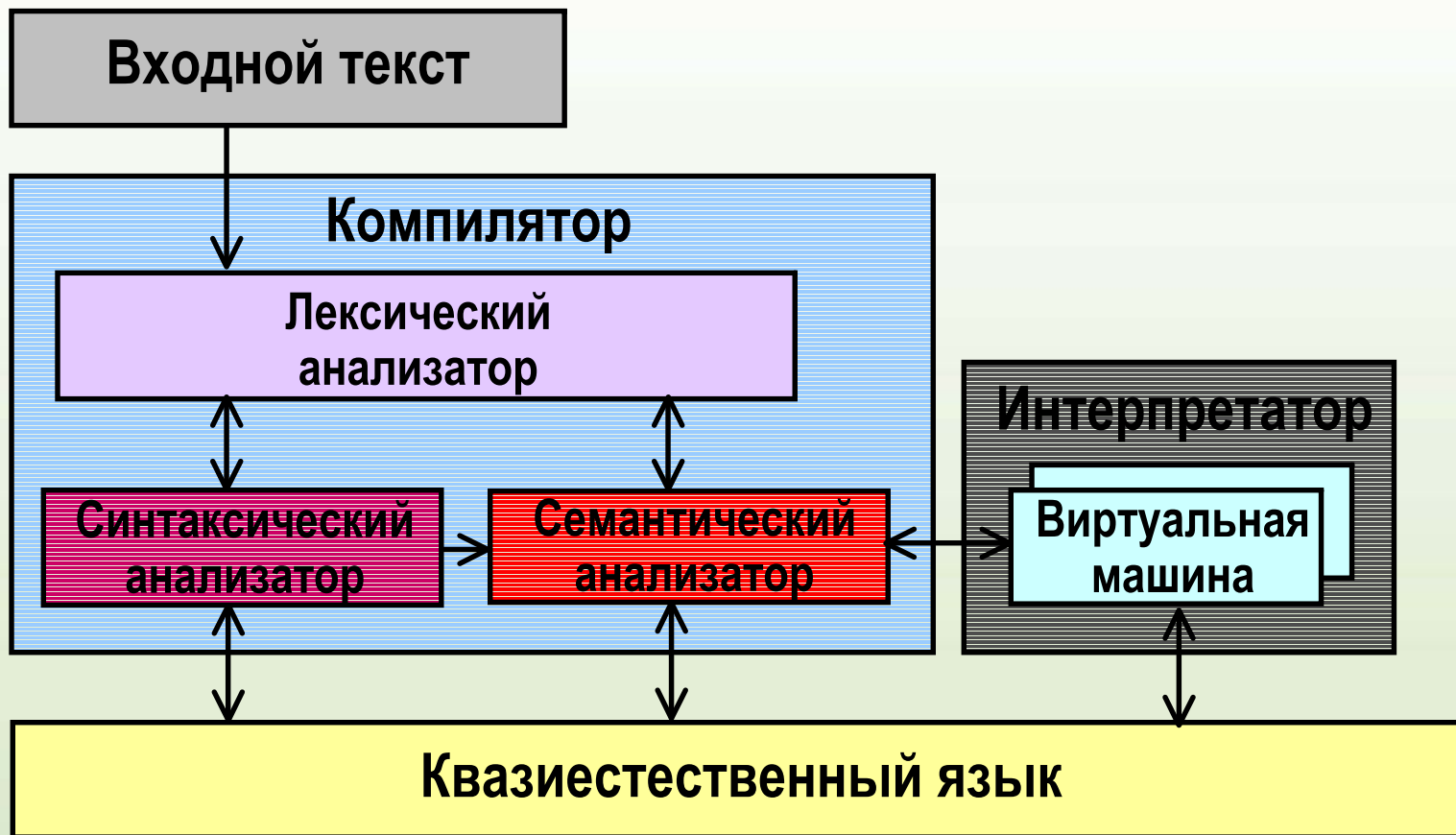
Пример вывода

# Методология и технология



Известные теории  
концептуального  
анализа

# Контекстная технология



## Проблемные языки:

- определяемые;
- дополняемые;
- библиотечные;
- общедоступные.

## Интерпретаторы:

- контроллеры, процессоры;
- абстрактные машины;
- операционные системы;
- системы программирования.

# VI

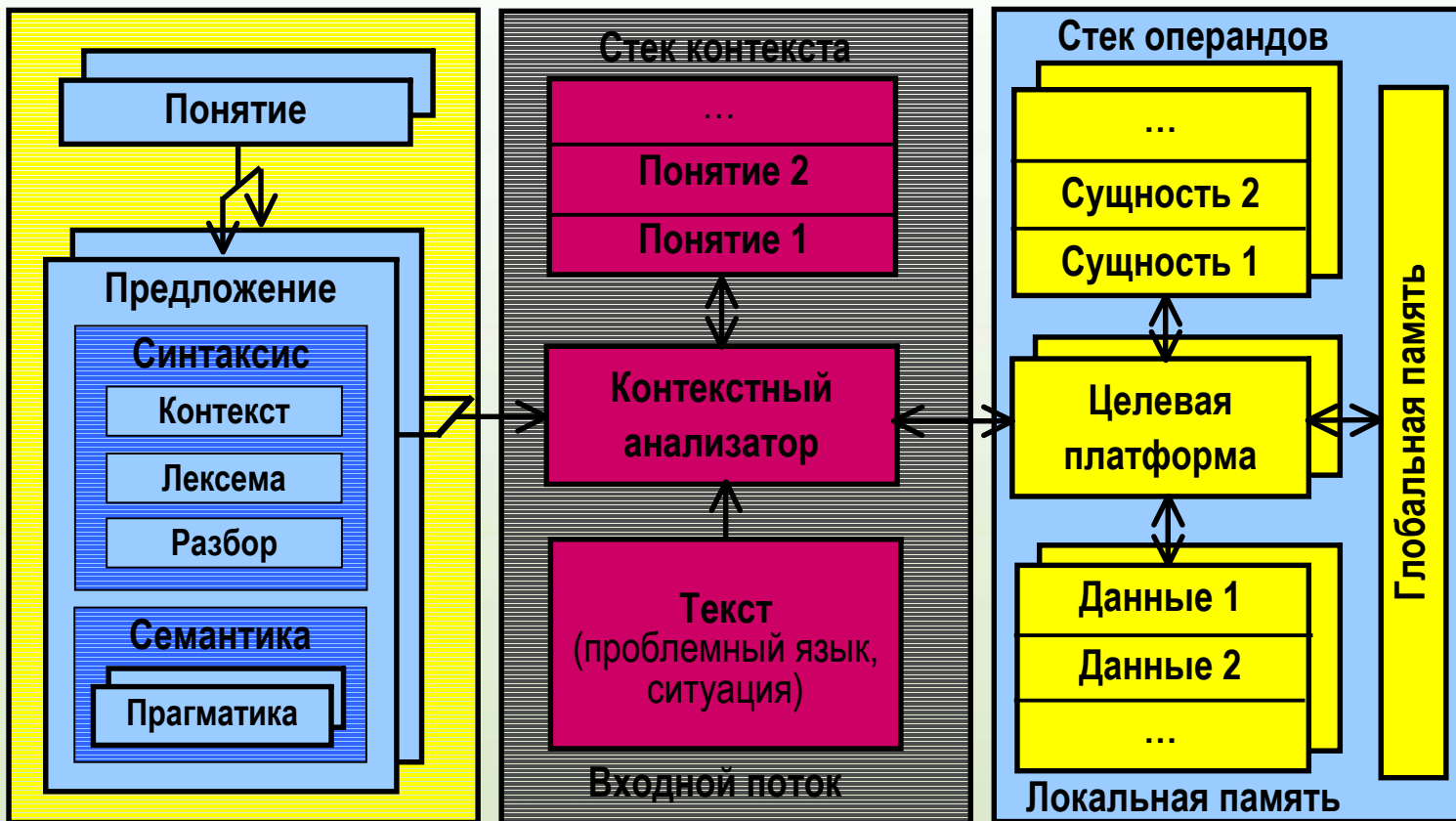
Контекстная технология

# Разнесенный разбор

Квазиестественный язык

Компилятор

Интерпретатор

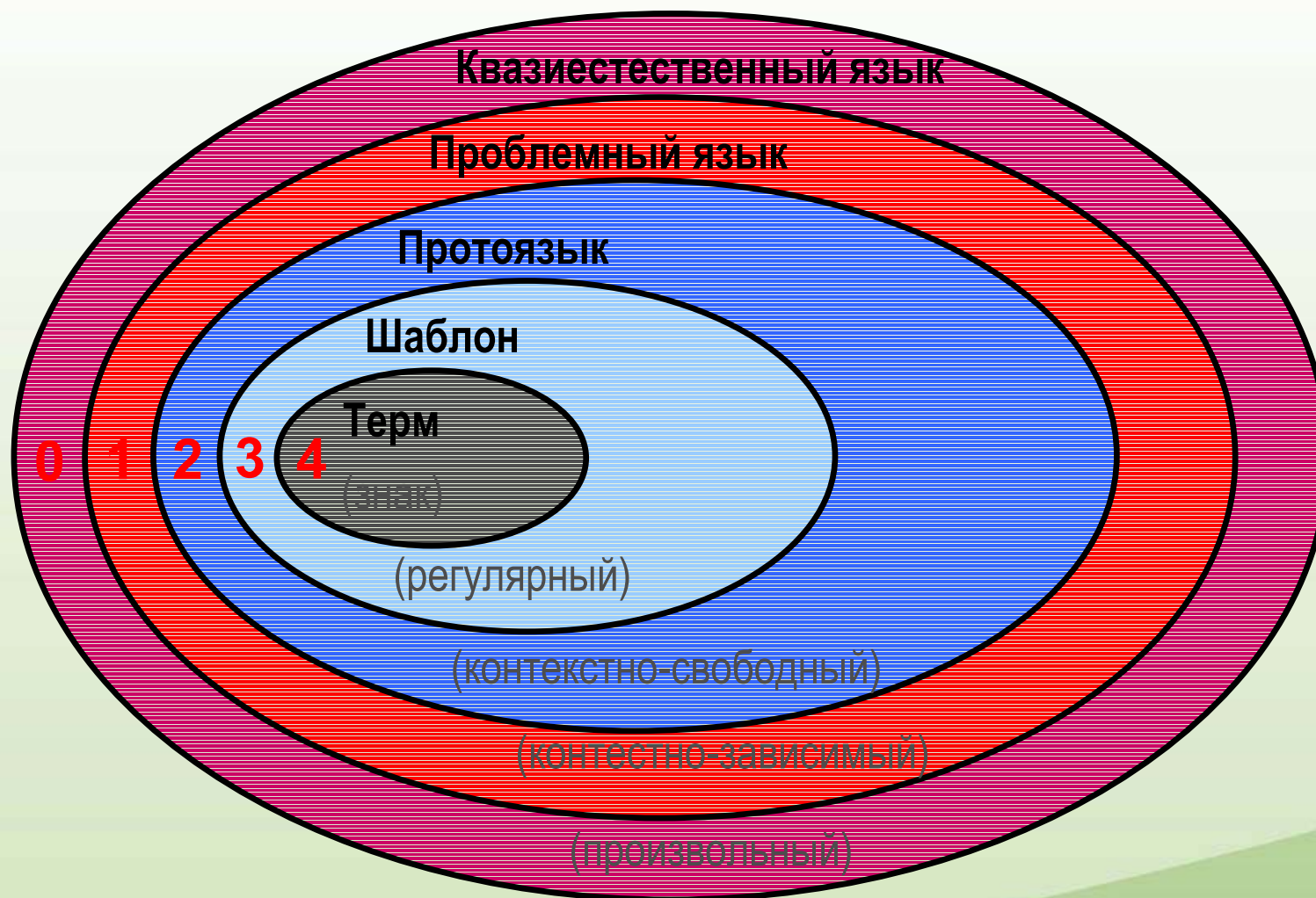


Известные формализмы для разбора текстов

**Boolean** [a] *'imp'* [b] **Boolean** { not a or b }

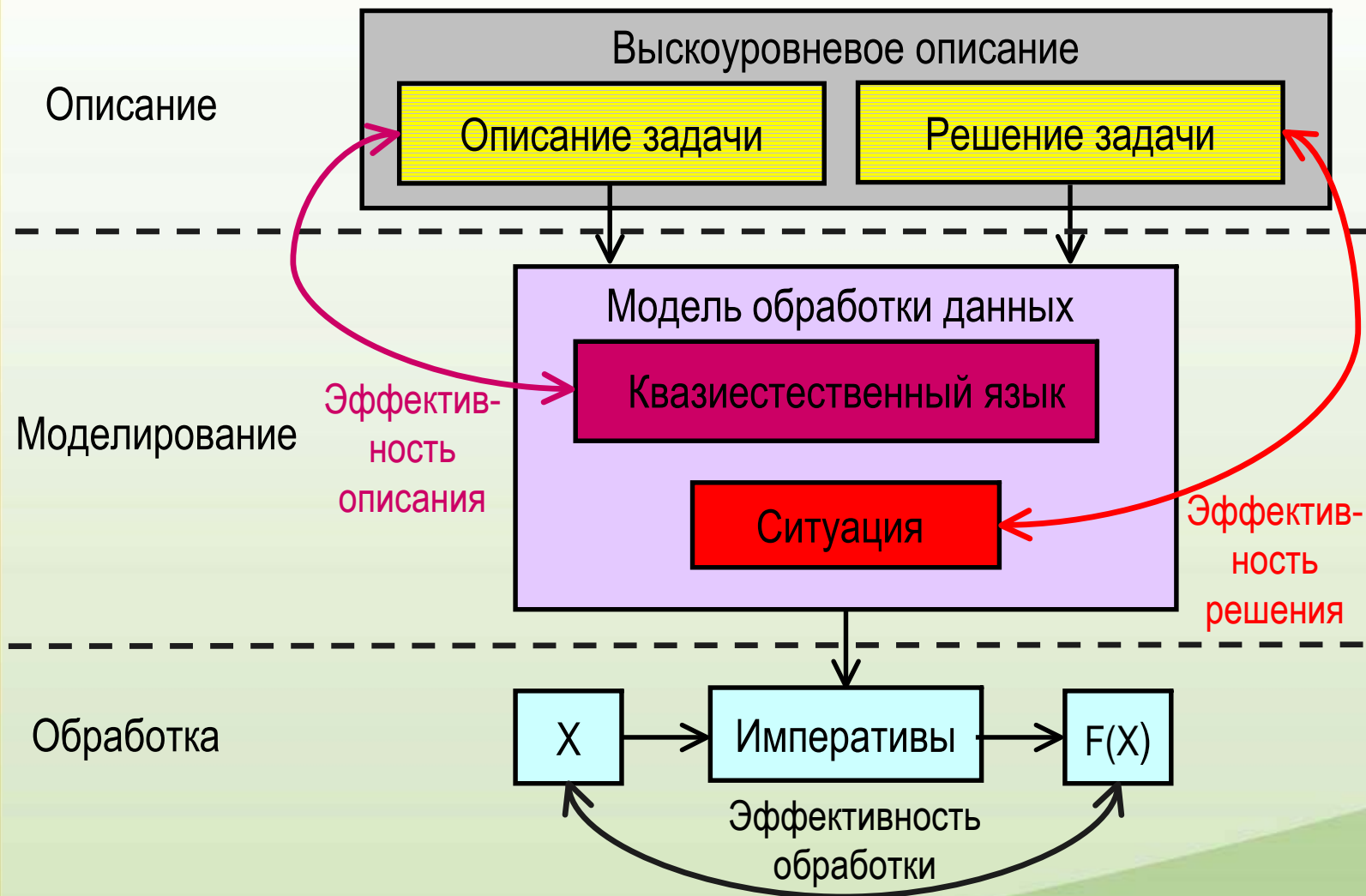
Контекст      Лексема      Разбор      Прагматика

# Иерархия языков

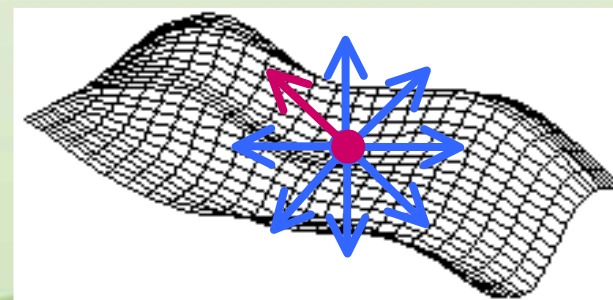
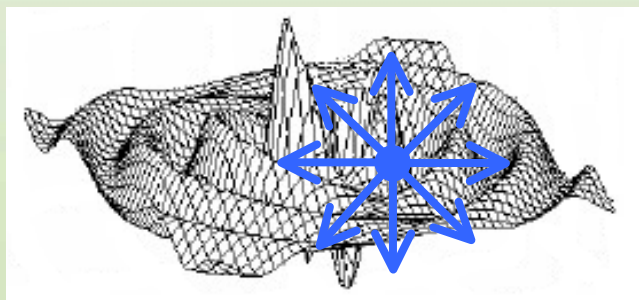
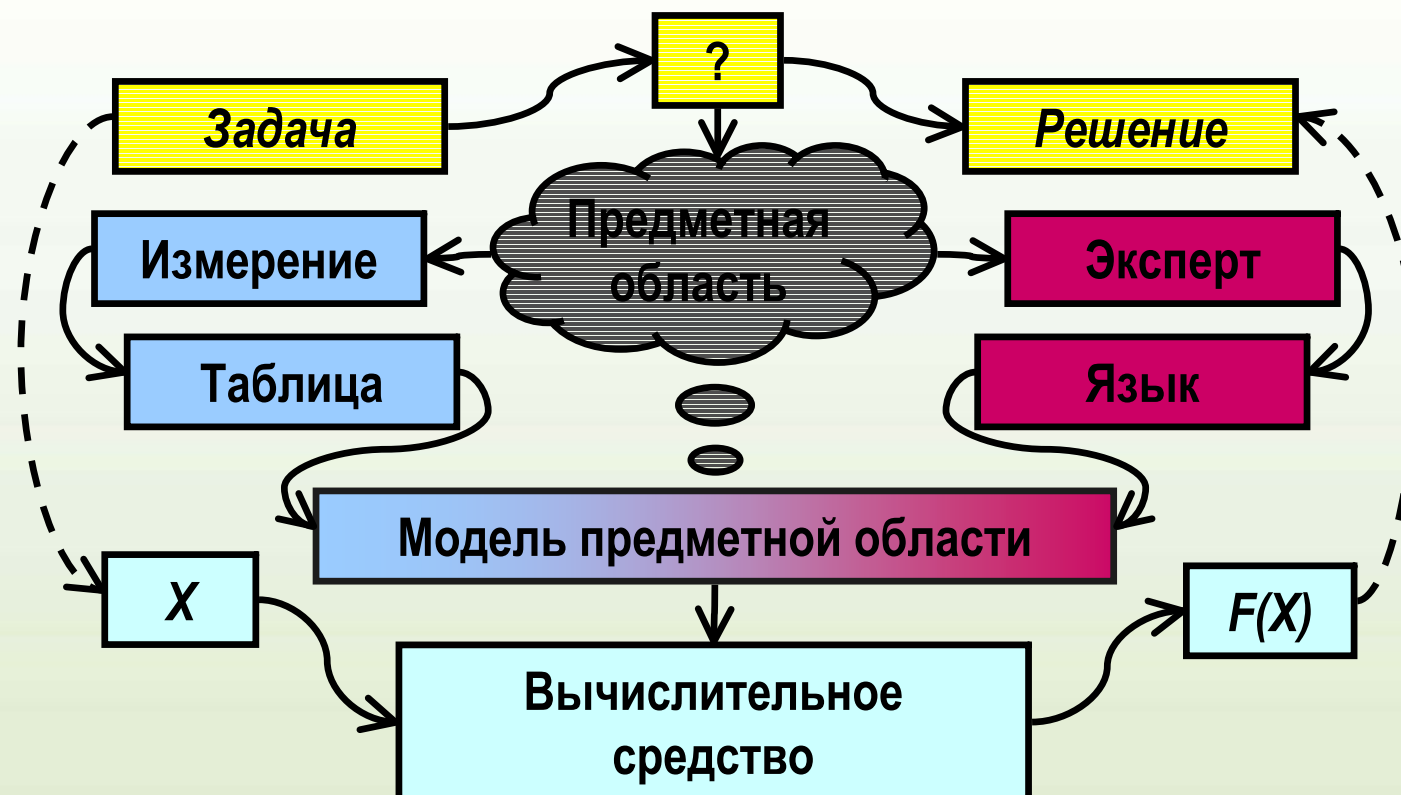




# Эффективность



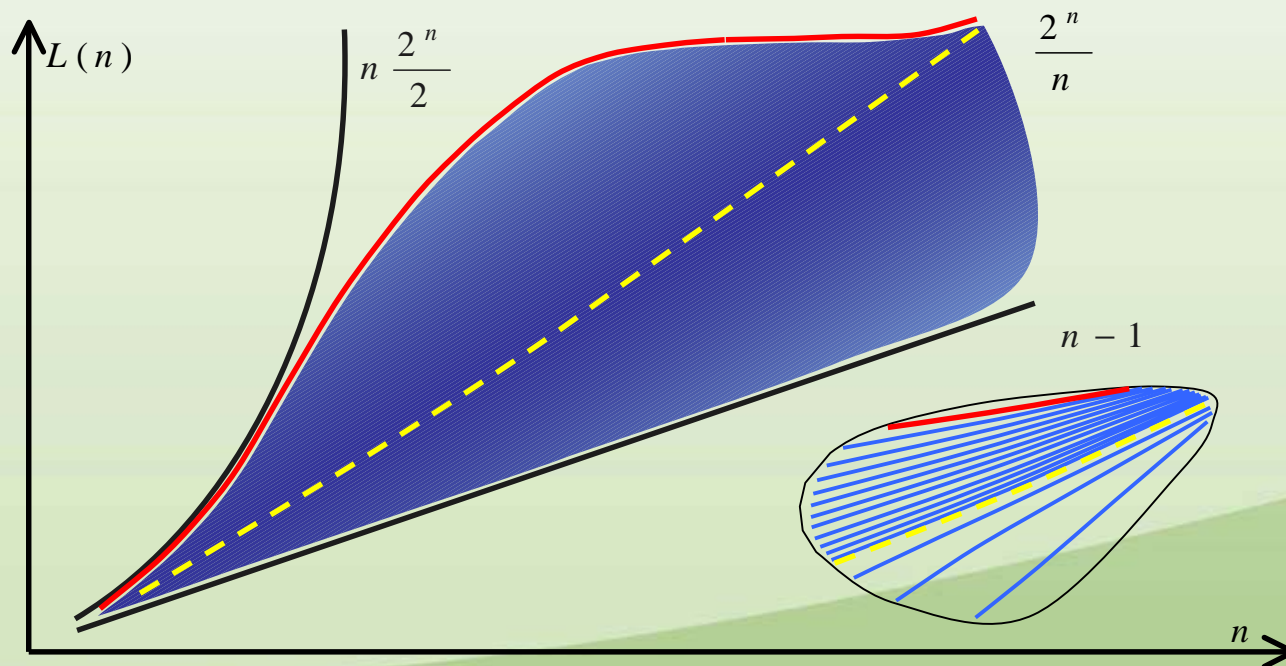
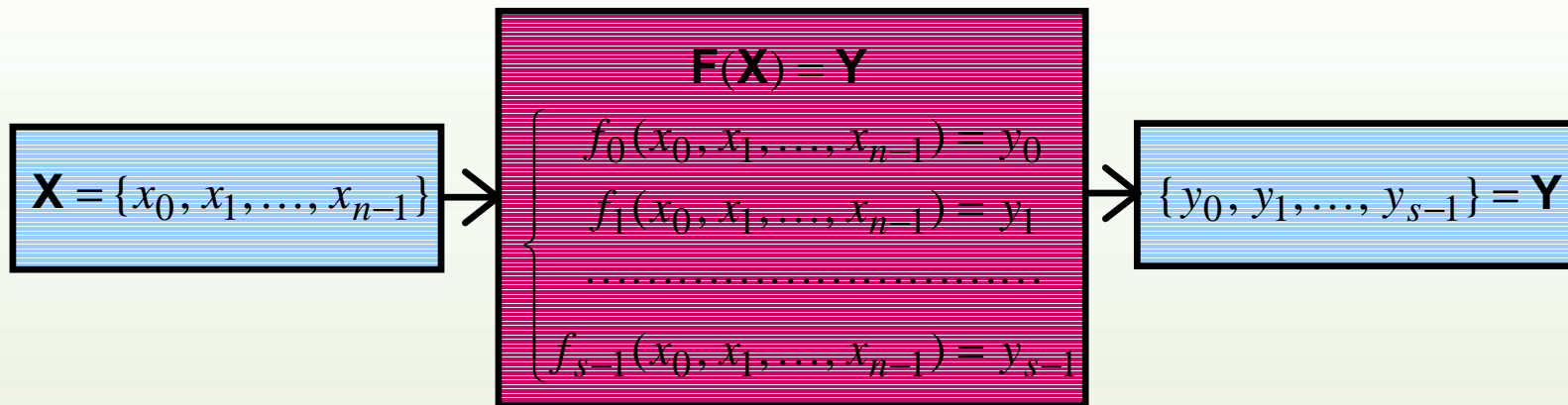
# Область применимости



# VII

Формально-логический подход

## Дискретная обработка данных



# VII

Формально-логический подход

## Образующие алгебры

$$f(X) = \sum_{i=0}^{M-1} g_i(X') \times h_i(X''), \quad \mathbf{F} = \mathbf{G} \times \mathbf{H}$$

**Алгебра логики**  $R_L = \langle N_k, +, \times \rangle$

$$+ = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & * & * & * \\ 2 & * & * & * \\ 3 & * & * & * \end{bmatrix} \quad \times = \begin{bmatrix} 0 & 0 & 0 & 0 \\ * & * & * & * \\ * & * & * & * \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

**G** – матрица перестановок (двухзначные унимодальные функции)

**Мультипликативная**  $R_M = \langle N_k, +, \times \rangle$

$$+ = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & * & * & * \\ 2 & * & * & * \\ 3 & * & * & * \end{bmatrix} \quad \times = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 1 \\ 0 & 3 & 1 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

**G** – мономиальная (многозначные унимодальные функции)

**Аддитивная**  $R_A = \langle N_k, +, \times \rangle$

$$+ = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \quad \times = \begin{bmatrix} 0 & 0 & 0 & 0 \\ * & * & * & * \\ * & * & * & * \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

**G** – биполярная (двухзначные мультимодальные функции)

**Фундаментальная**  $R_F = \langle N_k, +, \times \rangle$

$$+ = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \quad \times = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 1 \\ 0 & 3 & 1 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

**G** – фундаментальная (многозначные мультимодальные функции)

## VII

Формально-  
логический  
подход

# Аналитический синтез

$$f(X) = \sum_{i=0}^{M-1} g_i(X') \times h_i(X''), \quad X = X' \cup X''$$

$$g(X') = x'_0 \circ_0 x'_1 \circ_1 x'_2 \circ_2 \dots \circ_{n'-2} x'_{n'-1}$$

$$h(X'') = x''_0 \bullet_0 x''_1 \bullet_1 x''_2 \bullet_2 \dots \bullet_{n''-2} x''_{n''-1}$$



Пример синтеза

$$\left\{ \begin{array}{l} \text{fix } +, \times; \\ \text{var } X', X'', \circ_{ij}, \bullet_{ij}; \\ \text{min } M. \end{array} \right.$$

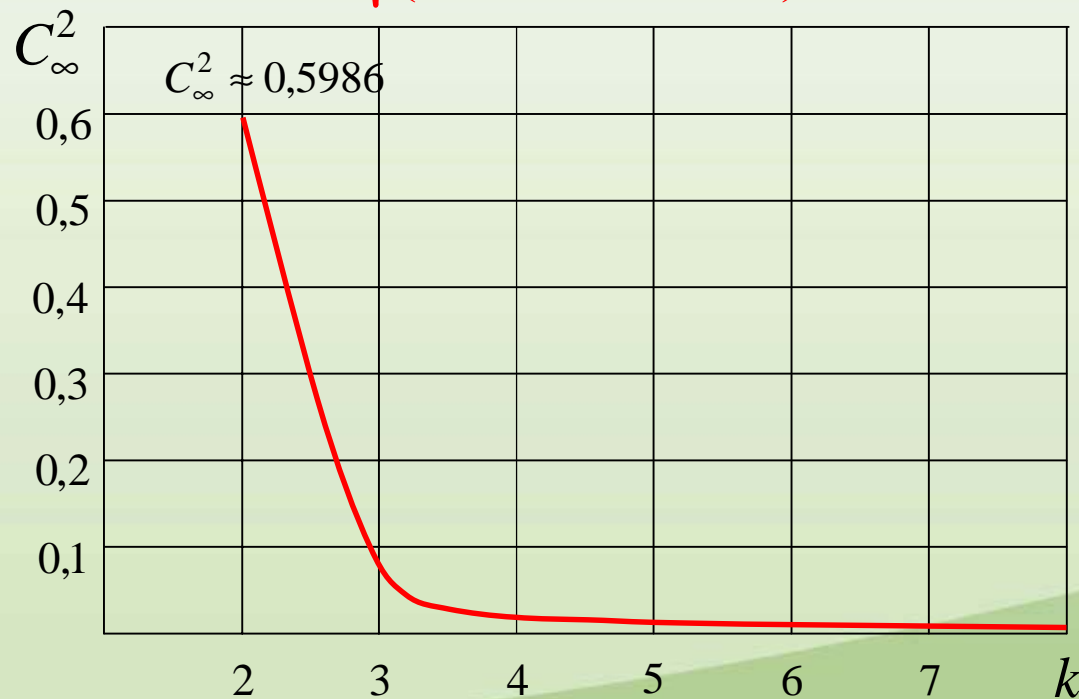
# VII

Формально-логический подход

## Сложность представления

$$L(k, k_0, k_1, \dots, k_{n-1}) = \frac{C^2}{n} \prod_{i=0}^{n-1} k_i$$

$$C = \frac{1}{k - \alpha} \sqrt{\left( \frac{1}{n} \left( \sum_{i=0}^{n-1} k_i - \sum_{j=0}^{\gamma-1} \beta_j \right) \right)^2 - \rho^2}$$





# Выводы

- Предложена **методология** понятийного анализа, основанная на 4-х видах связей между понятиями.
- Разработана **технология** представления и обработки знаний, сокращающая семантический разрыв между содержательными представлениями относительно предметной области и средствами их формальной спецификации.
- Решена **задача** индуктивного определения семантики формальных языков.

# Перечислительная семантика

Семантический язык – **натуральные числа**

$$N = \{0,1,2,3,\dots\}$$

$$0 \quad \{S \leftarrow 0\}$$

$$1 \quad \{S \leftarrow 1\}$$

$$10 \quad \{S \leftarrow 2\}$$

$$11 \quad \{S \leftarrow 3\}$$

$$100 \quad \{S \leftarrow 4\}$$

$$101 \quad \{S \leftarrow 5\}$$

$$110 \quad \{S \leftarrow 6\}$$

$$111 \quad \{S \leftarrow 7\}$$

...



Методы описания  
семантики



# Аксиоматическая семантика

Семантический язык – **формальная арифметика**

$$I \rightarrow 0 \quad \{S_I \leftarrow 0\}$$

$$I \rightarrow J \quad \{S_I \leftarrow S_J\}$$

$$J \rightarrow 1 \quad \{S_J \leftarrow 0'\}$$

$$J \rightarrow J0 \quad \{S_J \leftarrow (S_J) * (0' + 0')\}$$

$$J \rightarrow J1 \quad \{S_J \leftarrow (S_J) * (0' + 0') + 0'\}$$

$$I \rightarrow J \rightarrow J1 \rightarrow J01 \rightarrow 101$$

$$S_J \leftarrow 0'$$

$$S_J \leftarrow (S_J) * (0' + 0')$$

$$S_J \leftarrow (S_J) * (0' + 0') + 0'$$

$$S_I \rightarrow S_J$$

$$S_I \leftarrow ((0') * (0' + 0')) * (0' + 0') + 0'$$



Методы описания  
семантики

# Алгебраическая семантика

Семантический язык – **алгебра натуральных чисел**

$$A = \langle N, +, * \rangle$$

$$I \rightarrow 0 \quad \{S_I \leftarrow 0\}$$

$$I \rightarrow J \quad \{S_I \leftarrow S_J\}$$

$$J \rightarrow 1 \quad \{S_J \leftarrow 1\}$$

$$J \rightarrow J0 \quad \{S_J \leftarrow (S_J) * 2\}$$

$$J \rightarrow J1 \quad \{S_J \leftarrow (S_J) * 2 + 1\}$$

$$I \rightarrow J \rightarrow J1 \rightarrow J01 \rightarrow 101$$

$$S_J \leftarrow 1$$

$$S_J \leftarrow (1) * 2$$

$$S_J \leftarrow ((1) * 2) * 2 + 1$$

$$S_I \rightarrow S_J$$

$$S_I \leftarrow ((1) * 2) * 2 + 1$$



Методы описания  
семантики

# Денотационная семантика

Семантический язык –  **$\lambda$ -исчисление Черча**

$$F = \langle f, g \rangle, \quad gx = x * 2, \quad fx = x + 1$$

$$I \rightarrow 0 \quad \{S_I \leftarrow 0\}$$

$$I \rightarrow J \quad \{S_I \leftarrow S_J\}$$

$$J \rightarrow 1 \quad \{S_J \leftarrow 1\}$$

$$J \rightarrow J0 \quad \{S_J \leftarrow (\lambda x. gx)S_J\}$$

$$J \rightarrow J1 \quad \{S_J \leftarrow (\lambda x. fgx)S_J\}$$

$$I \rightarrow J \rightarrow J1 \rightarrow J01 \rightarrow 101$$

$$S_J \leftarrow 1$$

$$S_J \leftarrow (\lambda x. gx)1$$

$$S_J \leftarrow (\lambda x. fgx)(\lambda x. gx)1$$

$$S_I \rightarrow S_J$$

$$S_I \leftarrow (\lambda x. fgx)(\lambda x. gx)1$$



Методы описания  
семантики

# Операционная семантика

Семантический язык – **команды процессора**

$$K = \{mov\ A, B; shl\ A, B; inc\ A\}$$

$$I \rightarrow 0 \quad \{S_I \leftarrow mov\ V, 0;\}$$

$$I \rightarrow J \quad \{S_I \leftarrow S_J\}$$

$$J \rightarrow 1 \quad \{S_J \leftarrow mov\ V, 1;\}$$

$$J \rightarrow J0 \quad \{S_J \leftarrow S_J\ shl\ V, 1;\}$$

$$J \rightarrow J1 \quad \{S_J \leftarrow S_J\ shl\ V, 1; inc\ V;\}$$

$$I \rightarrow J \rightarrow J1 \rightarrow J01 \rightarrow 101$$

$$S_J \leftarrow mov\ V, 1;$$

$$S_J \leftarrow S_J\ shl\ V, 1;$$

$$S_J \leftarrow S_J\ shl\ V, 1; inc\ V;$$

$$S_I \leftarrow S_J$$

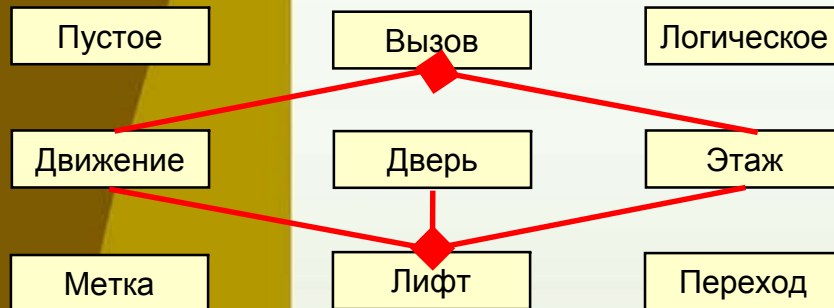
$$S_I \leftarrow mov\ V, 1; shl\ V, 1; shl\ V, 1; inc\ V;$$



Методы описания  
семантики

# Понятийный анализ задачи

## Денотационные:



Понятийный анализ

## Сигнификативные:

ожидать, открыть,  
закрыть, вверх, вниз,  
останов, двигать, выше,  
ниже, начать, продолжить.

### 1. Ожидание вызова

Если вызовов нет, то ожидать вызов (1). Если вызов со второго этажа, то перейти к открытию дверей (2), иначе – на принятие решения о движении (4).

### 2. Открытие дверей

Открыть дверь. Если дверь открылась, то перейти на принятие решения о движении (4), иначе повторить открытие двери (2).

### 3. Закрытие дверей

Закрыть дверь. Если дверь не закрылась, то повторить закрытие двери (3), иначе перейти к определению направления движения (4).

### 4. Направление движения

**Продолжение движения.** Если лифт двигался вверх (вниз) и имеются вызовы на движение вверх (вниз), то начать движение вверх (5) (вниз 6). **Изменение направления.** Если вызовов на движение вверх (вниз) нет, но есть вызовы на движение вниз (вверх), то начать движение вниз (6) (вверх 5). **Возврат в начало.** Если вызовов нет и при этом лифт выше (ниже) второго этажа, то начать движение вниз (6) (вверх 5), иначе перейти в состояние ожидания (1).

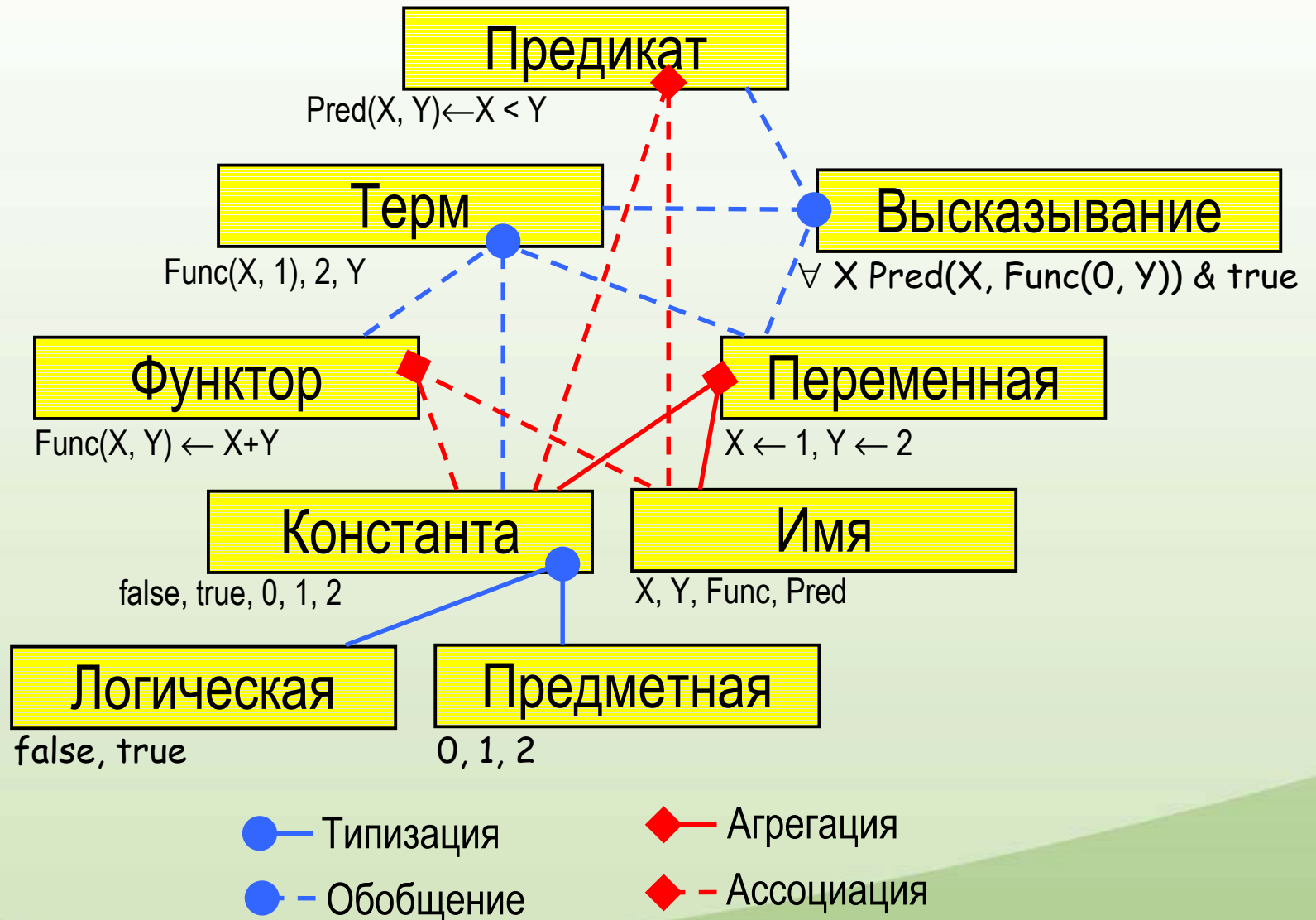
### 5. Движение вверх

Начать движение вверх. Если при проходе этажа имеется вызов на движение вверх, то остановить лифт и открыть дверь (2), иначе продолжить движение вверх (5).

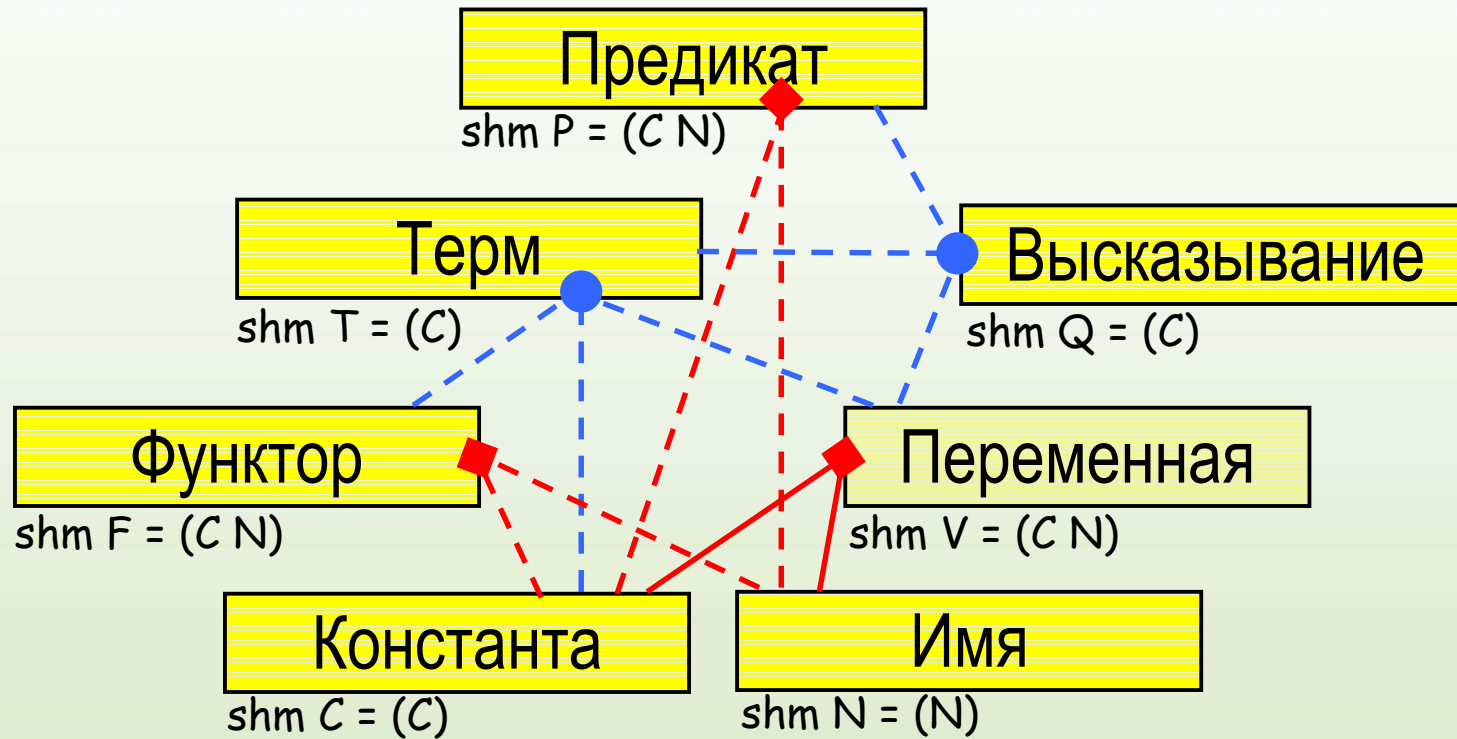
### 6. Движение вниз

Начать движение вниз. Если при проходе этажа имеется вызов для движения вниз, то остановить лифт и открыть дверь (2), иначе продолжить движение вниз (6).

# Исчисление предикатов



# Схемы понятий



● — Типизация

◆ — Агрегация

● - Обобщение

◆ - Ассоциация

# Вывод понятийной структуры

Константа

$$\frac{()() \neg^C ()C()}{() \vdash (C)} \quad \text{П}$$

Имя

$$\frac{()() \neg^N ()N()}{() \vdash (N)} \quad \text{П}$$

Переменная

$$\frac{()C() ()N() \neg^V ()V(C N)}{(C) (N) \vdash (C N)} \quad \text{А}$$

Функтор

$$\frac{()C() ()N() \neg^F ()F(C N)}{(C) (N) \vdash (C N)} \quad \text{А}$$

Предикат

$$\frac{()C() ()N() \neg^P ()P(C N)}{(C) (N) \vdash (C N)} \quad \text{А}$$

Терм

$$\frac{()F() ()V() ()C() \neg^T (F V C)T()}{(C N) (C N) (C) \vdash (C)} \quad \text{О}$$

Высказывание

$$\frac{()P() ()V() ()T() \neg^Q (P V T)Q()}{(C N) (C N) (C) \vdash (C)} \quad \text{О}$$





# Теории концептуального анализа

- Семантические сети (Collins & Quillian, 1969)
- Концептуальный анализ (Никаноров, 1972)
- Концептуальная зависимость (Shank & Rieger, 1974)
- Исчисление предикатов (Colmerauer, 1975)
- Модель «сущность-связь» (Chen, 1976)
- Концептуальные графы (Sowa, 1984)
- Онтологический анализ (Gruber, 1993)
- Объектный анализ (Booch, 1995)
- Формальный анализ понятий (Ganter & Wille, 1999)
- «Миварный» анализ (Варламов, 2002)
- Категорный подход (Бениаминов, 2003)
- Силлогистический вывод (Плесневич, 2004)



# Формализмы разбора текстов

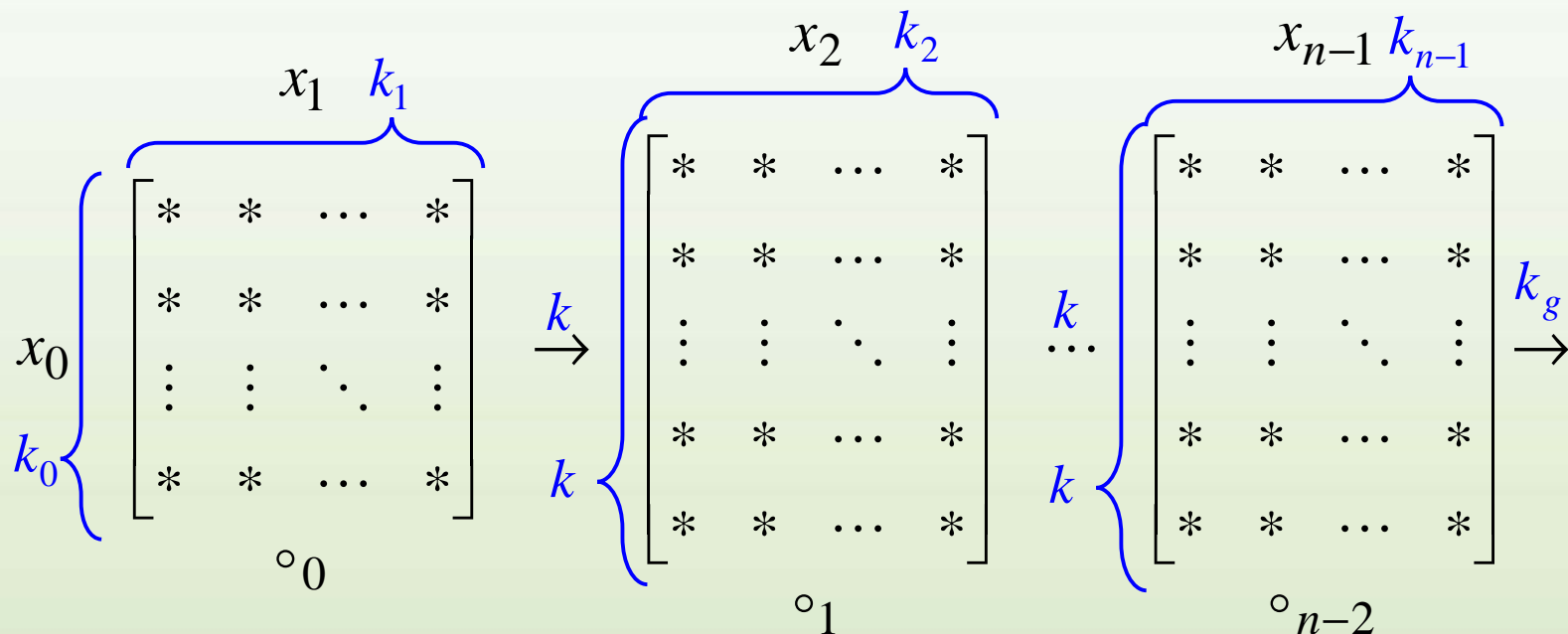
- Категориальные грамматики (Lambek, 1958)
- Порождающие грамматики (Chomsky, 1965 )
- Грамматики конечных автоматов (Johnson, 1972)
- Логические грамматики (Knuth, 1968; Борщев и Хомяков, 1973; Colmerauer, 1975)
- АТН-грамматики (Woods, 1978)
- Сопоставление с образцом (Пильщиков, 1983)
- Функциональные грамматики (Halliday, 1985)
- Трансформационные грамматики (Chomsky, 1986)
- Расширения контекстно-свободных грамматик (Heidorn, 1975; Allen, 1987)
- Унификационные формализмы (Kasper & Rounds, 1986)
- Программирование в ограничениях (Saraswat, 1993)



Разнесенный  
разбор

# Аналитическая конструкция

$$g(X) = x_0 \circ_0 x_1 \circ_1 x_2 \circ_2 \dots \circ_{n-2} x_{n-1}$$



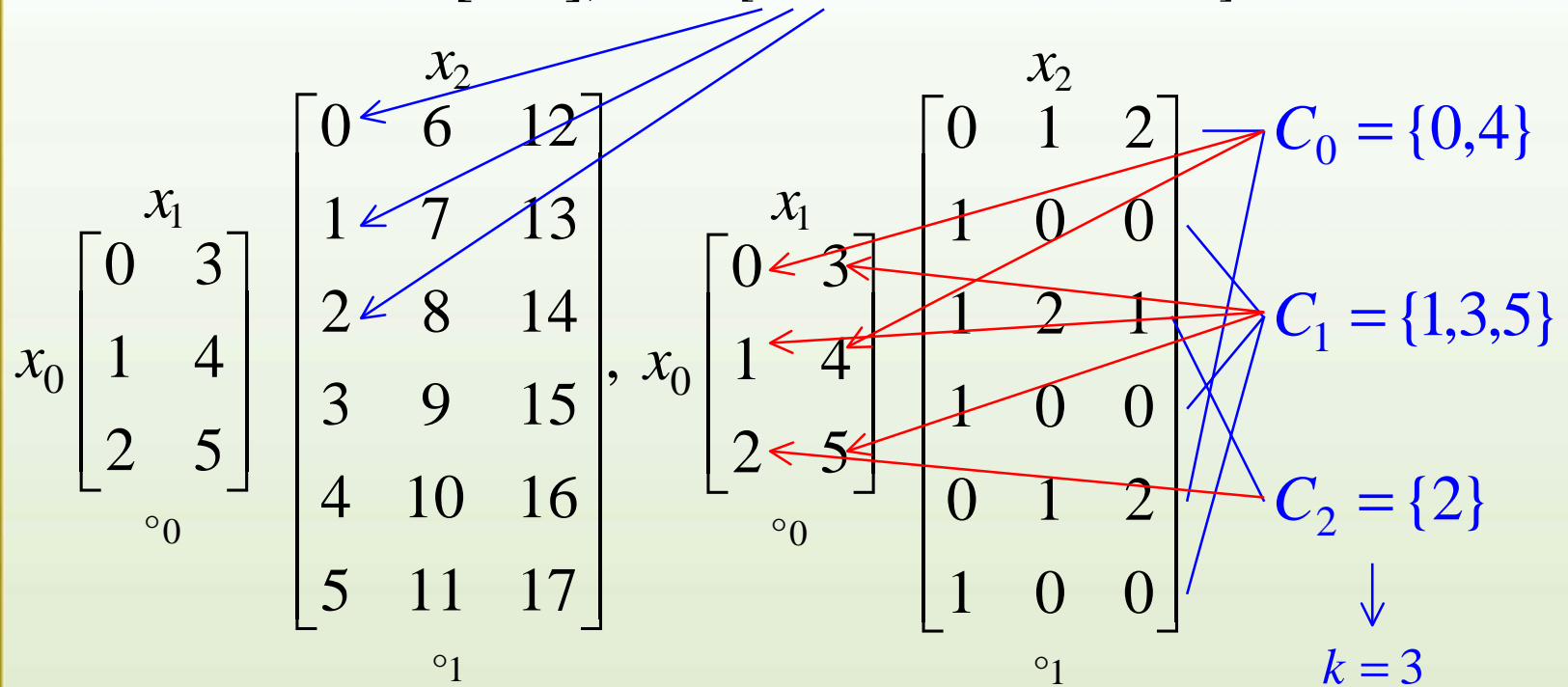
$k$  - значность аналитической конструкции

$$g(X) = ((\dots((x_0 \circ_0 x_1) \circ_1 x_2) \circ_2 \dots) \circ_{n-2} x_{n-1})$$



# Редукция

$$\mathbf{K} = [323], \quad \mathbf{F} = [011101102010201020]$$



$$x_0 \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{matrix} x_1 \\ x_2 \end{matrix} \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = g(X), \quad g(X) = x_0 \circ_0 x_1 \circ_1 x_2$$



Аналитический  
синтез

# Остаточный вектор

$$\mathbf{K} = [2223], \quad \mathbf{F} = [011120202022010020210202]$$

$$\begin{array}{c}
 x_3 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \\
 \begin{array}{c} x_0 \\ \circ 00 \end{array} \begin{bmatrix} 0 & 3 \\ 1 & 4 \\ 2 & 5 \end{bmatrix} \quad \begin{array}{c} x_1 \\ \circ 01 \end{array} \begin{bmatrix} 0 & 1 & * \\ 1 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 2 & 2 \\ 4 & * & 2 \\ 5 & 1 & * \end{bmatrix} \quad k=3 \\
 \begin{array}{c} x_2 \\ \circ 02 \end{array} \begin{bmatrix} 2 & 0 \\ 0 & 2 \\ 1 & 0 \end{bmatrix}, \quad \begin{array}{c} x_0 \\ \circ 00 \end{array} \begin{bmatrix} 2 & 1 \\ 0 & 2 \\ 0 & 2 \end{bmatrix} \quad \begin{array}{c} x_1 \\ \circ 01 \end{array} \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 1 & 0 \end{bmatrix} \quad \begin{array}{c} x_2 \\ \circ 02 \end{array} \begin{bmatrix} 2 & 0 \\ 0 & 2 \\ 1 & 0 \end{bmatrix} = g_0
 \end{array}$$

$$\mathbf{G}_0 = [022100222122200022000000]$$

$$\mathbf{F}_1 = \mathbf{F} - \mathbf{G}_0 = [022100122121200012200002]$$

