СТРУКТУРИРОВАНИЕ ЗАДАЧ, ПОИСК ОШИБОК И ТЕОРИЯ LEARNING SPACES В ИНТЕРАКТИВНОЙ СИСТЕМЕ ОБУЧЕНИЯ EDLETS

Турунтаев И.С.
аспирант 3 г.о. АШ по КН,
научный руководитель:
проф., д.ф.-м.н. Данилов В.Г.

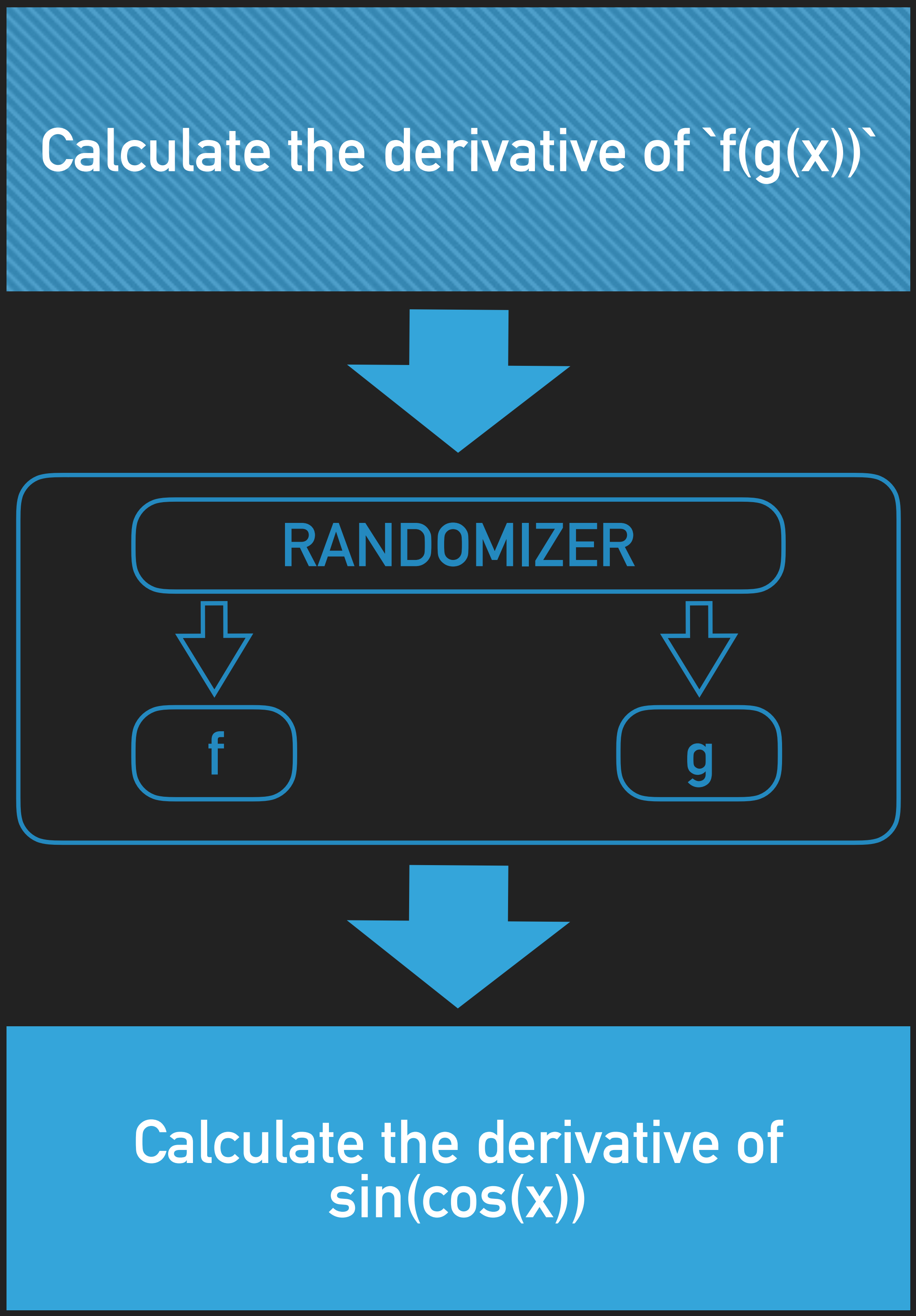
- EdLeTS
  - Что это?
  - Тренировочные задачи
  - Classrooms, training mode, quiz
- EdLeTS + Learning Spaces
  - Основные определения
  - Связь с EdLeTS
  - State uncovery mode

EDLETS

- Интерактивная система обучения для поддержки практических занятий по различным дисциплинам из области точных наук
- Ключевой объект – тренировочные задачи
- Основное внимание уделяется автоматизации процессов составления и сопровождения тренировочных задач, способам их организации, получении и анализе информации о состоянии и уровне знаний студентов

ШАБЛОН ТРЕНИРОВОЧНОЙ ЗАДАЧИ

- Описывает общую схему построения условий, решений и ответов для некоторого класса однотипных задач
- Зависит от набора динамических параметров
- Подстановка конкретных значений динамических параметров определяет конкретную тренировочную задачу
- Случайная генерация на основе случайного выбора значений из соответствующих множеств



МНОЖЕСТВО ПОРОЖДАЕМЫХ ЗАДАЧ

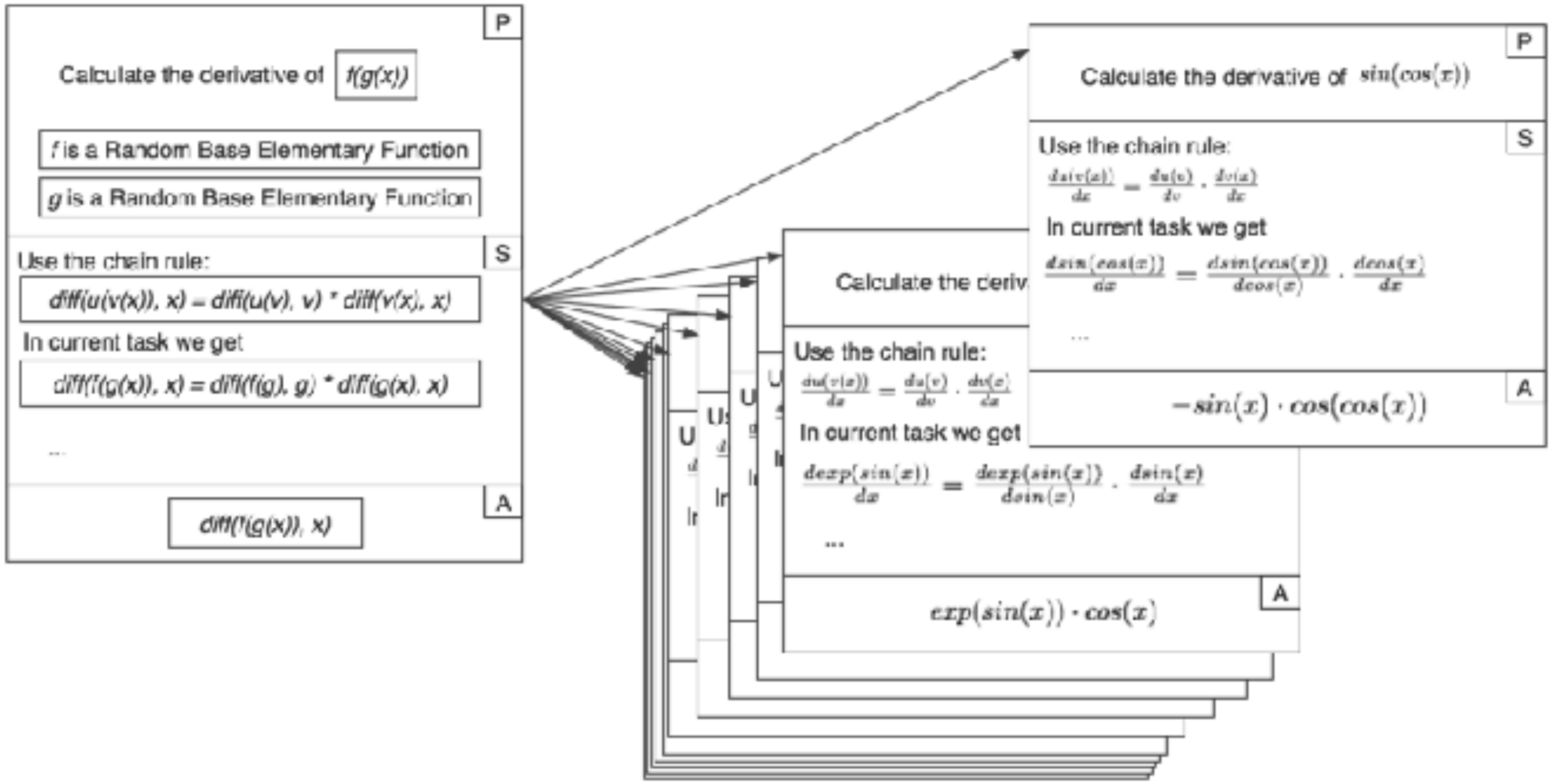
- Количество порождаемых конкретных задач растет $\sim m^n$ (где n – количество динамических параметров, m – min мощность множества значений)

- Автоматическая проверка ответов позволяет спокойно обрабатывать весь этот объем

  ⇒ возможность использования в тренировочном режиме

  ⇒ success ratio

  ⇒ автоматическая генерация контрольных работ



ОПИСАНИЕ ШАБЛОНОВ

- SmallTask для описания динамических параметров задач
  - специфика области применения
  - максимальная простота синтаксиса
- Markdown для разметки сопроводительного текста
- Редактор задач, дающий возможность избежать необходимости изучать язык и вникать в его нюансы

ОРГАНИЗАЦИЯ ПОЛЬЗОВАТЕЛЕЙ И ЗАДАЧ

- Объединение пользователей и тренировочных задач в группы – Classrooms
- Каждый пользователь может создать свой Classroom
- Два типа пользователей Classroom – преподаватели и студенты
  - преподаватели управляют, добавляют задачи, формируют к.р., просматривают статистику и пр.
  - студенты ~~страдают~~ обучаются

CLASSROOMS: РЕЖИМЫ ДОСТУПА

- **PUBLIC**: виден всем; любой пользователь может присоединиться;
- **PROTECTED**: виден всем; только преподаватель может приглашать студентов;
- **XPROTECTED**: скрыт; только преподаватель может приглашать студентов;
- **PRIVATE**: виден всем; только организатор может приглашать пользователей;
- **XPRIVATE**: скрыт; только организатор может приглашать пользователей;
- **CLOSED**: скрыт; никого приглашать нельзя; просмотр возможен только для организаторов.

QUIZES: КОНТРОЛЬНЫЕ РАБОТЫ

- Quiz – мета-шаблон из нескольких задач
- Одна генерация Quiz – «билет» – случайная генерация каждой задачи
- Quiz генерируется один-единственный раз на запрос студента
- Генерации хранятся в БД (в отличие от тренировочного режима)
- Отправленный Quiz не подлежит «переписыванию»
- Автоматическая проверка + метки об использовании поточечной проверки

LEARNING SPACES

JEAN-CLAUDE FALMAGNE · JEAN-PAUL DOIGNON

- (Falmagne, 2010) Falmagne J. C., Doignon J. P. Learning spaces: Interdisciplinary applied mathematics. – Springer Science & Business Media, 2010.
- (Falmagne, 2006) Falmagne, J. C., Cosyn, E., Doignon, J. P., & Thiéry, N. (2006). The assessment of knowledge, in theory and in practice. In Formal concept analysis (pp. 61-79). Springer Berlin Heidelberg.

LEARNING SPACES

- **Domain** – Q – множество объектов (задач)
- **Knowledge state** – некоторое подмножество Q
- **Knowledge structure** $\{Q, \mathcal{K}\}$: Q – domain, $\mathcal{K}$ – множество knowledge states над Q;
- **Knowledge space** – knowledge structure, замкнутая относительно объединения

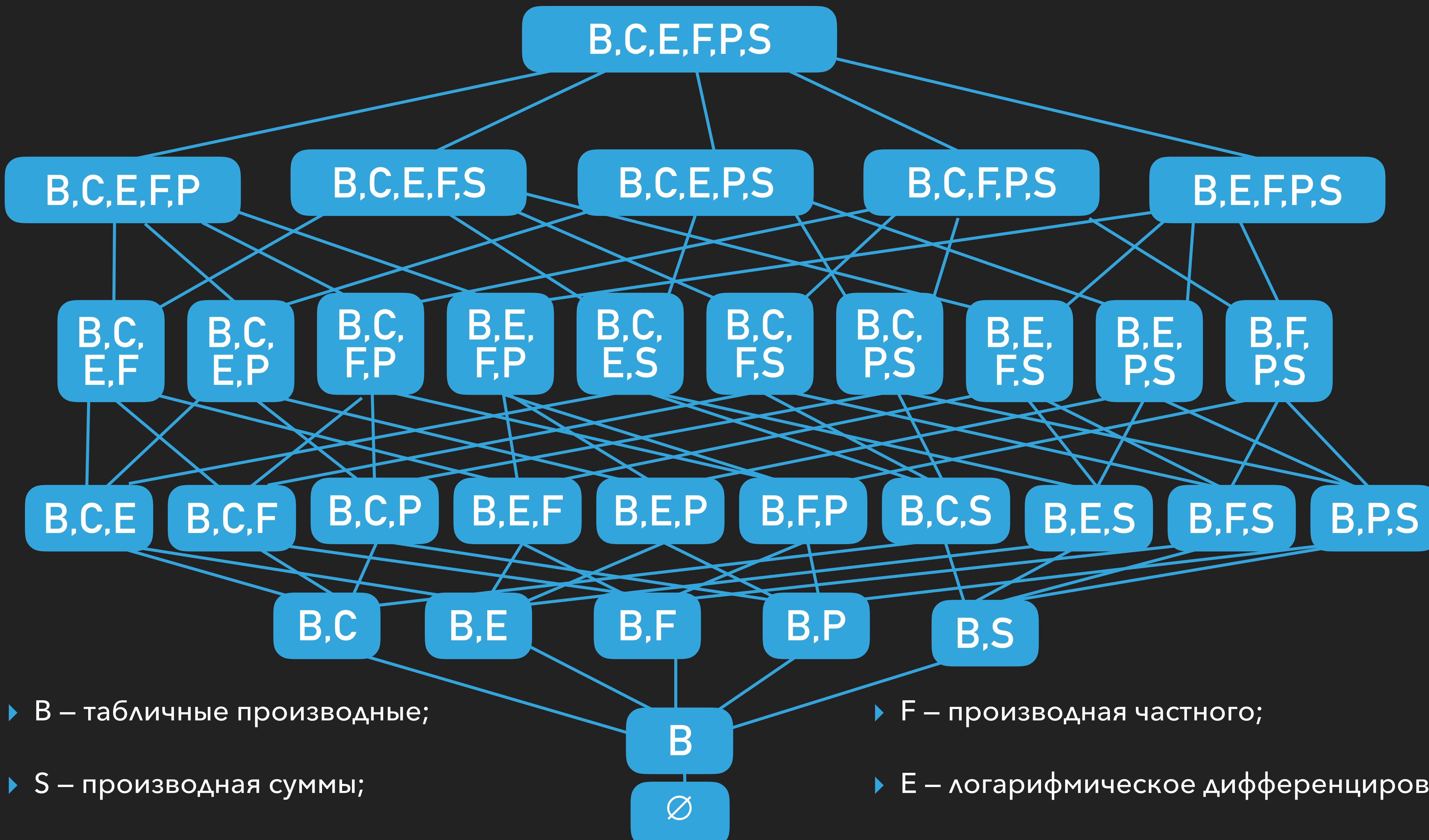
$$\forall \mathcal{F} \subseteq \mathcal{K} \quad \bigcup_{F \in \mathcal{F}} F \in \mathcal{K}$$

- **Learning space** – well-graded knowledge space

wellgradedness:

$$\forall K, L \in \mathcal{F},\ K \neq L \quad \exists \{K = K_0, K_1, \ldots, K_n = L\}, \text{ s.t.}$$
$$|K_{i-1} \Delta K_i| = 1 (\forall i = 1 \ldots n) \text{ and } |K \Delta L| = n$$

ПРИМЕР KNOWLEDGE SPACE ДЛЯ ЗАДАЧ НА ВЫЧИСЛЕНИЕ ПРОИЗВОДНЫХ

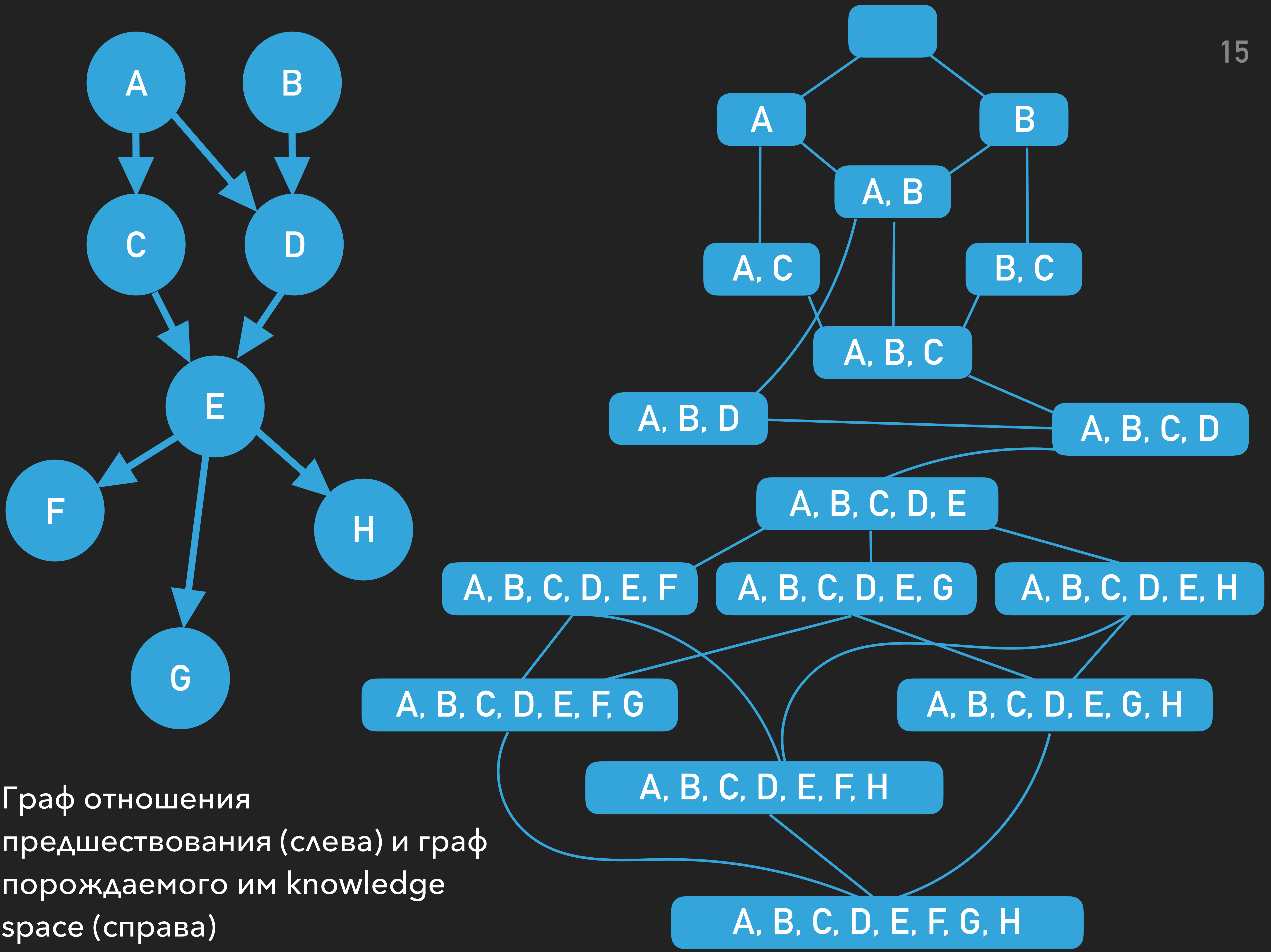


- B – табличные производные;
- S – производная суммы;
- P – производная произведения;
- F – производная частного;
- E – логарифмическое дифференцирование;
- C – производная композиции;

- Основная задача теории LS – автоматическое определение состояния (assessment machine)
  - вычисление начального уровня студента в заданной области
  - оценка текущих знаний и коррекция состояния
- Построение learning paths
- Нас в первую очередь интересуют:
  - fill-the-gaps – вычисление потенциальны пробелов (inner fringe)
  - suggestions – предложения следующих шагов (outer fringe)
- В LS не рассматривается вопрос вычисления inner fringe и outer fringe

КАКОЕ ЭТО ИМЕЕТ ОТНОШЕНИЕ К EDLETS?

- Множетво задач формирует domain (Q)
- Отношение предшествования (R) на задачах группы задает частичный порядок (рефлексивно + транзитивно + антисиммертично)
- На основе этого отношения можно породить пространство $(Q,\mathcal{K})$ по правилу: $(K\in\mathcal{K}) \Leftrightarrow (\forall(p,q)\in R\ (q\in K)\Rightarrow(p\in K))$
- Порожденное таким образом $\mathcal{K}$ есть knowledge space, замкнутый относительно пересечения (Falmagne, 2010)



Граф отношения предшествования (слева) и граф порождаемого им knowledge space (справа)

FILL-THE-GAPS & SUGGESTIONS

- Для данной задачи требуется вычисление ближайших предшественников, либо задачи следующего уровня
- Малое число задач в группе
- Удобно хранить само отношение: поиск концов ребер, входящих в вершину или исходящих из нее
- В такой постановке определение состояния не требуется

STATE UNCOVERY

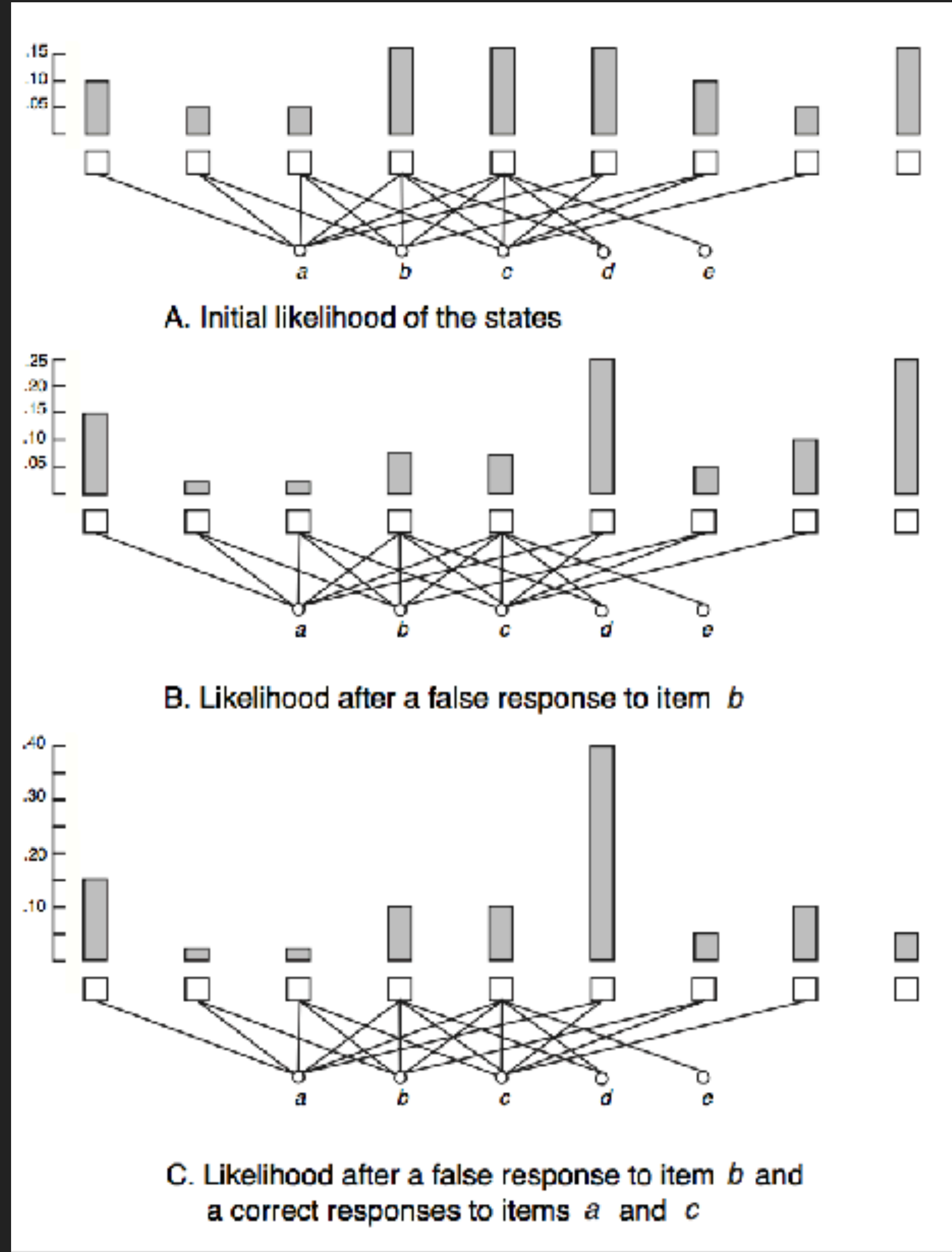
- Режим, позволяющий определить текущий уровень знаний студентов в интересующей области
- Задача определения knowledge state в заданном knowledge space
  - построение knowledge space по заданному precedence relation
  - вычисление состояния на основе ответов на предлагаемые задачи
- Markov process procedure
- Проверка на well-gradedness и использование Markov chain procedure

СПАСИБО ЗА ВНИМАНИЕ!

isturunt@gmail.com

ИДЕЯ АЛГОРИТМА

- (b, -), (a, +), (c, +)
- Рассм. (Q, $\mathcal{K}$, L)
- Уточняем вероятности состояний
    - $L_1 = L$
    - Шаг: при неправильном ответе на очередной вопрос q уменьшаем вер. состояний с q и увеличиваем вер-ти состояний без q. И наоборот.



A. Initial likelihood of the states

B. Likelihood after a false response to item *b*

C. Likelihood after a false response to item *b* and a correct responses to items *a* and *c*

MARKOV PROCESS PROCEDURE (R_n, Q_n, L_n)

$(Q, \mathcal{K}, L)$	a finite probabilistic knowledge structure;
Λ_+	the set of all positive probability distributions on $\mathcal{K}$;
Γ	the state space of the process $(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n)_{n \in \mathbb{N}}$;
K_0	the latent knowledge state of the subject;
$\mathbf{L}_1 = L$	the initial probability distribution on $\mathcal{K}$, $0 < L < 1$;
$\mathbf{L}_n(K)$	a r.v. representing the probability of state K on trial n;
$\mathbf{Q}_n$	a r.v. representing the question asked on trial n;
$\mathbf{R}_n$	a r.v. representing the response given on trial n;
Ψ	$(q, \mathbf{L}_n) \mapsto \Psi(q, \mathbf{L}_n)$, the questioning rule;
u	$(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n) \mapsto u(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n)$, the updating rule;
$\mathbf{W}_n$	random history of the process from trial 1 to trial n;
ι_A	the indicator function of a set A.

- Questioning Rule:
  $\Psi: (q, L_n) \longmapsto \Psi(q, L_n) \in [0,1]$ – вероятность $Q_n = q \in Q$
- Updating Rule:
  $L_{n+1} = u(R_n, Q_n, L_n)$ *(a.s.)*
- Если выполнено условие $P(\lim_{n \longrightarrow \infty}(L_n(K_0)) = 1) = 1$, то K_0 называют uncoverable
- При определенных условиях на процесс, questioning rule и updating rule состояние вычислимо

Stochastic assessment process for $(\mathbf{Q}, \mathcal{K}, \mathbf{L})$, parameterized by u, Ψ and K_0.

[U] **Updating Rule.** We have $\mathbb{P}(\mathbf{L}_1 = L) = 1$, and for any positive integer n and all measurable sets $B \subseteq \Lambda_+$,

$$\mathbb{P}(\mathbf{L}_{n+1} \in B \,|\, \mathbf{W}_n) = \iota_B\big(u(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n)\big),$$

where u is a function mapping $\{0,1\} \times Q \times \Lambda_+$ to Λ_+. Writing u_K for the coordinate of u associated with the knowledge state K, we thus have

$$\mathbf{L}_{n+1}(K) \overset{\text{a.s.}}{=} u_K(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n).$$

Moreover, the function u satisfies the following condition:

$$u_K(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n) \begin{cases} > \mathbf{L}_n(K) & \text{if } \iota_K(\mathbf{Q}_n) = \mathbf{R}_n, \\ < \mathbf{L}_n(K) & \text{if } \iota_K(\mathbf{Q}_n) \neq \mathbf{R}_n. \end{cases}$$

[Q] **Questioning Rule.** For all $q \in Q$ and all positive integers n,

$$\mathbb{P}(\mathbf{Q}_n = q \,|\, \mathbf{L}_n, \mathbf{W}_{n-1}) = \Psi(q, \mathbf{L}_n)$$

where Ψ is a function mapping $Q \times \Lambda_+$ to the interval $[0,1]$.

[R] **Response Rule.** For all positive integers n,

$$\mathbb{P}(\mathbf{R}_n = \iota_{K_0}(q) \,|\, \mathbf{Q}_n = q, \mathbf{L}_n, \mathbf{W}_{n-1}) = 1$$

where K_0 is the latent state.

13.4.2 Definition. The updating rule u of Axiom [U] will be called *convex with parameters* $\theta_{q,r}$, where $0 < \theta_{q,r} < 1$ for $q \in Q$ and $r \in \{0,1\}$, if the function u of Axiom [U] satisfies the following condition:

For all $K \in \mathcal{K}$ and with $\mathbf{L}_n = L_n$, $\mathbf{R}_n = r$, and $\mathbf{Q}_n = q$,

$$u_K(r,q,L_n) = (1-\theta_{q,r})L_n(K) + \theta_{q,r} g_K(r,q,L_n) \tag{13.7}$$

where

$$g_K(r,q,L_n) = \begin{cases} r\frac{L_n(K)}{L_n(\mathcal{K}_q)}, & \text{if } K \in \mathcal{K}_q \\ (1-r)\frac{L_n(K)}{L_n(\mathcal{K}_{\bar{q}})}, & \text{if } K \in \mathcal{K}_{\bar{q}}. \end{cases}$$

13.4.4 Definition. The updating rule is called *multiplicative with parameters* $\zeta_{q,r}$, where $1 < \zeta_{q,r}$ for $q \in Q, r = 0,1$, if the function u of Axiom [U] satisfies the condition: with $\mathbf{Q}_n = q$, $\mathbf{R}_n = r$, $\mathbf{L}_n = L_n$ and

$$\zeta_{q,r}^K = \begin{cases} 1 & \text{if } \iota_K(q) \neq r, \\ \zeta_{q,r} & \text{if } \iota_K(q) = r \end{cases} \tag{13.9}$$

we have

$$u_K(r,q,L_n) = \frac{\zeta_{q,r}^K L_n(K)}{\sum_{K' \in \mathcal{K}} \zeta_{q,r}^{K'} L_n(K')}. \tag{13.10}$$