# 1. Правила сдачи выводов формул

Когда требуется показать вывод, не используя ничего, то приемлемые формы ответа:

- В виде дерева вывода, узлы которого ”— аксиомы или modus ponens;
- В виде последовательности формул, каждая из которых ”— аксиома или modus ponens. Для modus ponens нужно уточнять, какие две формулы (из тех, что доказаны выше) подаются ему как посылки.
- В виде терма соответствующего типа на Haskell.

Когда требуется показать вывод, используя что угодно, то приемлемые формы ответа пополняются такими вариантами:

- В виде дерева вывода, узлы которого ”— что угодно, что известным способом транслируется в чистое исчисление;
- В виде последовательности формул с дополнительными правилами;
- В виде терма, для которого понятен механизм трансляции в чистое исчисление.

# 2. Исчисление высказываний

Для этого набора задач введём исчисление $X$ с такими правилами:

$$\frac{}{A \to B \to A}\text{ A1}$$
$$\frac{}{(A \to B \to C) \to (A \to B) \to A \to C}\text{ A2}$$
$$\frac{}{\bot \to A}\text{ A}\neg$$
$$\frac{}{((A \to B) \to A) \to A}\text{ A11}$$
$$\frac{A \quad A \to B}{B}\text{ modus ponens}$$

Вспомните, как мы кодировали пары и `Either` в чистом лямбда-исчислении (подсказка: в каком-то смысле `uncurry` эквивалентно паре, а `either` эквивалентно `Either`). Такую же схему можно применить и для логики высказываний: можно сказать, что высказывание $A \wedge B$ эквивалентно $\forall C. (A \to B \to C) \to C$, а $A \vee B$ эквивалентно $\forall C. (A \to C) \to (B \to C) \to C$.

Оттранслировав аксиомы исчисления высказываний по этой схеме, получим такое:

1. $A \to B \to A$
2. $(A \to B \to C) \to (A \to B) \to A \to C$
3. $(\forall c. (A \to B \to c) \to c) \to A$
4. $(\forall c. (A \to B \to c) \to c) \to B$
5. $A \to B \to \forall c((A \to B \to c) \to c)$
6. $A \to \forall c((A \to c) \to (B \to c) \to c)$
7. $B \to \forall c((A \to c) \to (B \to c) \to c)$
8. $(A \to C) \to (B \to C) \to \forall c((A \to c) \to (B \to c) \to c) \to C$
9. $A \to (A \to \bot) \to B$
10. $(A \to B) \to (A \to B \to \bot) \to A \to \bot$

11. $\forall c((A \to c) \to ((A \to \bot) \to c) \to c)$

Заметьте, что квантор $\forall$ здесь не по переменным, а по другим формулам, так что написанное здесь не выразить в логике предикатов (логике первого порядка), для этого нужна логика более высокого порядка.

- Предварите аксиомы 5, 6, 7, 9, 10 и 11 (только перемещая кванторы, не приводя к КНФ), уберите кванторы и докажите результирующие выражения в исчислении $X$. За каждую аксиому по баллу, только за 11 два балла. В представленных выводах не должно встречаться понятие контекста, хотя выводы и можно получить автоматизированно применением леммы о дедукции к выводам, использующим контекст.

# 3. Полнота исчисления высказываний

- Обоснуйте, почему исчисление $X$ корректное. (Плюсик ставится в задание 1 раздела).
- Напишите программу, которая для произвольной формулы в связках $\neg, \to$ либо приведёт контрпример, либо вернёт её вывод в исчислении $X$. (Плюсики ставятся в задания 2-6 раздела).

# 4. Секвенциальное и интуиционистское исчисления высказываний

- Будем считать, что секвенция представлена именно списком, а не множеством формул, и аксиома выглядит как $A, \Gamma \vdash A, \Delta$. Как тогда изменится множество выводимых в исчислении формул, если оттуда убрать
  - Оба правила расширения? Оба правила сокращения? *(1 б.)*
  - Оба правила перестановки? Вообще все структурные правила? *(1 б.)*
  - Как изменятся ответы на предыдущие пункты, если аксиома выглядит как $A \vdash A$? *(1 б.)*
- Докажите, используя что угодно, или опровергните формулы в интуиционистском исчислении высказываний:
  - $((a \to b) \to a) \to \neg\neg a$ *(1 б.)*
  - $a \vee \neg a \to \neg\neg a \to a$ *(1 б.)*
  - $a \vee \neg a \to ((a \to b) \to a) \to a$ *(1 б.)*

# 5. Логика предикатов

- Расширим квантор существования: пусть $\exists_P x Q(x)$ означает, что $Q(x)$ выполняется для такого количества различных $x$, которое удовлетворяет предикату $P$. Например, $\exists_{=1} x Q(x)$ означает, что число элементов носителя, для которых выполняется $Q$, ”— это $1$, не больше и не меньше, а $\exists_{<1} x Q(x)$ ”— то же самое, что $\neg\exists x Q(x)$. (*Кстати, это настолько полезно, что математики действительно вводят квантор $\exists! x Q(x)$, который эквивалентен определённому выше $\exists_{=1} x Q(x)$; это читается как “существует уникальный”*).
  - Как переписать в терминах обычных кванторов $\exists_{=n} x Q(x)$, где $n \geq 0$ ”— заранее выбранное число, если в сигнатуре имеется предикат равенства? *(1 б.)*
  - Как переписать в терминах обычных кванторов $\exists_{>n} x Q(x)$, где $n \geq 0$ ”— заранее выбранное число, если в сигнатуре имеется предикат равенства? *(1 б.)*
  - Как переписать в терминах обычных кванторов $\exists_{<n} x Q(x)$, где $n \geq 0$ ”— заранее выбранное число, если в сигнатуре имеется предикат равенства? *(1 б.)*

# 6. Общезначимые формулы логики предикатов

- Напишите программу, которая пытается найти контрпример к данной на вход формуле логики предикатов, используя носитель размера 3. Гарантируется, что:
  - Нет ни одного функционального символа арности 3 или выше;
  - Есть не более пяти функциональных символов арности 2;
  - Есть не более пяти функциональных символов арности 1;
  - Есть не более десяти функциональных символов арности 0;
  - Нет ни одного предикатного символа арности 4 или выше;
  - Есть не более двух предикатных символов арности 3;
  - Есть не более пяти предикатных символов арности 2;
  - Есть не более трёх предикатных символов арности 1;
  - Есть не более двух предикатных символов арности 0.

  Можно предполагать, что заранее задана какая-то сигнатура, удовлетворяющая данным ограничениям, и на вход подаются только формулы в ней. *(от 3 б.)*

# 7. Исчисление предикатов

- Реализуйте полуразрешающий алгоритм для тавтологий логики высказываний. Для этого напишите программу, которая генерирует все возможные выводы в исчислении высказываний, пока не найдётся нужный. Обратите внимание, что необходимо, чтобы со временем в процессе перебора встретился каждый корректный вывод. *(6 б.)*
- Используя что угодно, выведите $\forall x(P(x) \to Q(x)) \to \forall x P(x) \to \forall x Q(x)$ *(1 б.)*.
- Используя что угодно, выведите $\forall x(P(x) \to Q(x)) \to \neg\exists x Q(x) \to \forall x \neg P(x)$ *(1 б.)*.

# 8. Секвенциальное исчисление предикатов

- Известно, что не существует алгоритма, который говорил бы, является ли некоторая формула тавтологией логики предикатов, а значит, нет алгоритма, который находит вывод в секвенциальном исчлислении высказываний или контрпример. Что в доказательстве полноты секвенциального исчисления предикатов нельзя оформить как алгоритм? *(1 б.)*
- Выведите $\forall x(P(x) \to Q(x)) \to \forall x P(x) \to \forall x Q(x)$ и $\forall x(P(x) \to Q(x)) \to \neg\exists x Q(x) \to \forall x \neg P(x)$ *(1 б.)*.