Сошников Дмитрий Валерьевич
к.ф.-м.н., доцент
http://soshnikov.com

Лекция 4: Метод резолюции

Логическое программирование

https://soshnikov.com/courses/logpro/

В чем проблема синтаксического метода доказательства?

- Нет ориентации на результат
  - Мы начинаем с известного множества формул и «расширяем» это множество
  - Формула, которую требуется доказать, не учитывается
- На каждом шаге можно применить:
  - Правило вывода к любым двум формулам
  - Произвольную замену переменной

Методы решения

- Сведение всех правил вывода + замены переменной к одному универсальному правилу вывода
- Замена многократного перебора минимально возможным
- Исходим из целевой формулы и пытаемся придти к исходным посылкам

Простая резолюция для логики высказываний

- Modus ponens: $A,\ A \supset B \vdash B$
- Modus tollens: $A \supset B,\ \neg B \vdash \neg A$
- Эквивалентная форма записи:

$$\neg A \vee B,\ \neg B \vdash \neg A$$

- Расширенное правило резолюции:

$$L_1 \vee \ldots L_n \vee \neg A,\ M_1 \vee \ldots \vee M_k \vee A \vdash L_1 \vee \ldots \vee L_n \vee M_1 \vee \ldots \vee M_k$$

$L_i,\ M_k$ – литеры (с отрицанием или без)

Пример

- Если Петя опоздает на работу, то его уволят или сделают выговор. Петя опоздает, если не придет автобус, или не прозвонит будильник.

$$l \supset (f \vee t)$$
$$(a \vee b) \supset l$$

$$l \supset (f \vee t) = \neg l \vee f \vee t$$
$$(a \vee b) \supset l = \neg(a \vee b) \vee l = (\neg a \wedge \neg b) \vee l$$
$$= (\neg a \vee l) \wedge (\neg b \vee l)$$

Пример

- Рассмотрим случай, когда будильник не прозвонил и Петю не уволили

$\neg l \vee f \vee t$

$\neg a \vee l$

$\neg b \vee l$

a

$\neg f$

$\neg a \vee l$

$\neg l \vee t$

$\neg a \vee t$

a

t

- Получили вывод – Пете сделали выговор
- При этом мы все равно «думали», какие правила выбирать для вывода

Сложности с логикой предикатов

- Необходимо привести формулу к эквивалентному множеству дизьюнктов
- Возникает возможность «замены переменной» внутри предиката

1. Переход к предваренной (пренексной) нормальной форме
2. Сколемизация
3. Приведение к коньюнктивной нормальной форме (КНФ)
4. Получение множества дизьюнктов

Пренексная нормальная форма

- ***Предваренной*** *(или* ***пренексной****)* ***нормальной формой*** *формулы исчисления предикатов называется формула вида*
- $(\boxtimes x_1)\dots(\boxtimes x_n)\ F(x_1,\dots,x_n) \qquad (4)$
- *где* $\boxtimes$ *- квантор общности или существования,* $F(x_1,\dots,x_n)$ *- формула, не содержащая кванторов.*
- *Для любой замкнутой формулы существует эквивалентная ПНФ*

$$(\forall x)(P(x) \supset ((\exists y)Q(x,y) \vee (\exists y)R(x,y))) =$$
$$(\forall x)(P(x) \supset ((\exists y_1)Q(x,y_1) \vee (\exists y_2)R(x,y_2))) =$$
$$(\forall x)(\exists y_1)(\exists y_2)(P(x) \supset (Q(x,y_1) \vee R(x,y_2)))$$

Скулемизация

- A= $(\forall x)(\exists y)\ Q(x,y)$
- Рассмотрим формулу B = $(\forall x)\ Q(x,f(x))$
- Свойство: A выполнима <=> B выполнима

- Алгоритм Сколемизации:
  - Выбирается самое левое вхождение квантора существования. Пусть, например, это будет квантор по переменной x_k.
  - Выбирается не входящий в формулу F функциональный символ $f^{(k-1)}$, и все вхождения переменной x_k в формулу F заменяются на $f^{(k-1)}(x_1,\ldots,x_{k-1})$, т.е. формула приобретает вид $F(x_1, \ldots, x_{k-1,}f^{(k-1)}(x_1,\ldots,x_{k-1}),x_{k+1},\ldots,x_n)$.
  - Процесс повторяется с шага 1 до полного исключения кванторов существования.

Построение множества дизъюнктов

- $(\forall x)(A \wedge B) \;=\; ((\forall x)A) \wedge ((\forall x)B)$
- Сколемовская нормальная форма $(\forall x_1)\ldots(\forall x_n)\ (A_1 \wedge \ldots \wedge A_m)$ невыполнима <=> невыполнимо множество дизъюнктов $\{(\forall x_1)\ldots(\forall x_n)\ A_1, \ldots, (\forall x_1)\ldots(\forall x_n)\ A_m\}$
- Каждая формула A_i – это дизъюнкт $L_1 \vee .. \vee L_k$

Что мы получили?

- Произвольную формулу логики предикатов A можно преобразовать в множество дизъюнктов S_A : A несовместна <=> S_A несовместна

Процесс доказательства методом резолюции

- Необходимо доказать формулу A, т.е. $\Gamma \vdash A$
- Для этого докажем несовместность $\neg A$, т.е. противоречивость $\Gamma \cup \{\neg A\}$
- Преобразуем формулу к множеству дизьюнктов S
- Из множества дизьюнктов надо вывести противоречие, или пустой тождественно ложный дизьюнкт $\square$

Пример

- $(\forall x)\ H(x) \supset M(x)$ $\neg H(x) \vee M(x)$
- H(Сократ) H(Сократ)

- Как применить правило резолюции в этим формулам?
- Необходимо совершить замену переменной x=Сократ
- Чтобы формализовать этот процесс, вводят понятие **унификации**

Унификация

- ***Подстановкой*** *$\theta=\{X_1/t_1,\ldots,X_n/t_n\}$ называется множество пар переменных X_i и соответствующих им термов t_i, где $(\forall i\neq j)$ $X_i \neq X_j$, $t_i \neq X_i$.*
- ***Применением θ к формуле или терму F*** *(обозначается $F\theta$) называется формула или терм (строго определяемые рекурсивно), в которой все вхождения X_i заменены на соответствующие термы t_i.*
- ***Композиция унификаторов*** *$\theta=\sigma\tau$, если для всех формул F : $F\theta = (F\sigma)\tau$*
- *Подстановка θ называется* ***унификатором*** *для формул или термов F и G, если $F\theta = G\theta$. Формулы, для которых существует унификатор, называются* ***унифицирующимися****.*

Примеры

- $f(X,g(a,X))$ и $f(b,Y)$ - $\{X/b, Y/g(a,b)\}$
- $f(X,g(a,X))$ и $f(c,g(a,b))$ – не унифицируются
- $f(X,X)$ и $f(Z,Y)$ – $\{X/Z, Y/Z\}$ или $\{X/a, Z/a, Y/a\}$

Наиболее общий унификатор

- Очевидно, что если существует унификатор θ формул F и G, то существует бесконечное множество унификаторов
- Определим отношение порядка ≤ :
  - $\theta \leq \tau \iff \exists\sigma : \tau = \theta\sigma$
- Наиболее общий унификатор mgu(F,G)= sup { θ | Fθ=Gθ }
- Пример:
  - *f(X,Y) и f(g(Z),Z)*
  - $\theta_1 = \{X/g(Z), Y/Z\}$, $\theta_2 = \{X/g(a), Y/a, Z/a\}$
  - $\theta_2 = \{Z/a\}\,\theta_1 \Rightarrow \theta_1$ - наиболее общий

Обобщенное правило резолюции

$$L_1 \vee \ldots \vee L_n \vee \neg A, \quad M_1 \vee \ldots \vee M_k \vee B$$
$$\vdash \quad (L_1 \vee \ldots \vee L_n \vee M_1 \vee \ldots \vee M_k)\theta$$
$$\text{где } \theta = mgu(A,B)$$

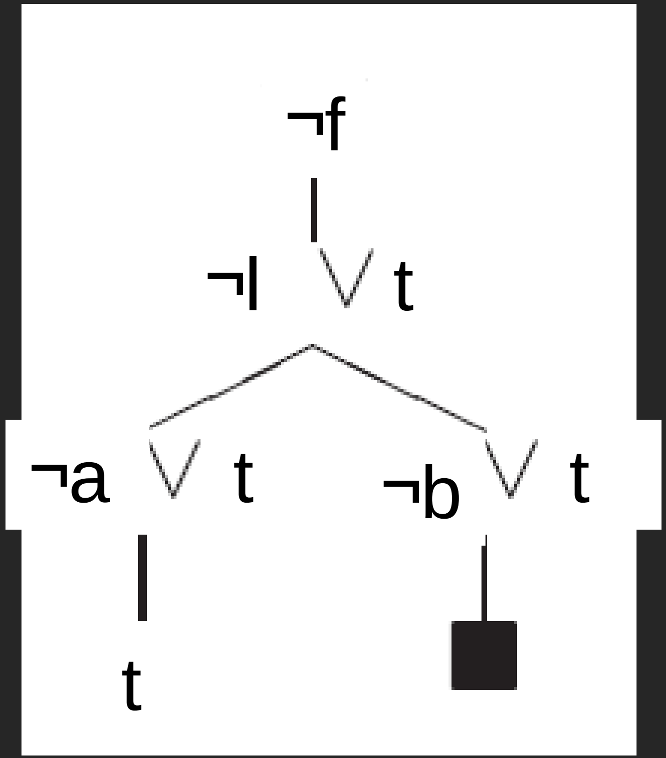
$\neg$H(x)$\vee$M(x) и H(Сократ)
mgu(H(x),H(Сократ)) = { x/Сократ }
M(x){x/Сократ} = M(Сократ)

Стратегии резолюции

- ***Линейная стратегия резолюции*** *(L-резолюция) - это стратегия, при которой на каждом шаге резолюции в качестве одной из посылок используется формула, полученная на предыдущем. Для первого шага в качестве посылки берется добавленное к множеству формул отрицание доказываемого предложения.*
- ***Резолюцией с выбирающим правилом*** *(S-резолюцией) называется стратегия, при которой литера для применения резолюции выбирается в соответствии с некоторым правилом*
- В ***упорядоченной стратегии резолюции*** *значение имеет порядок литер в дизъюнктах, которых сохраняется при применении правила вывода; при этом разрешается в качестве общего дизъюнкта использовать только самую левую литеру. Упорядоченная стратегия является частным случаем S-резолюции.*

SL-резолюция на примере

¬l∨f∨t
¬a∨l
¬b∨l
a
¬f



Дизъюнкты Хорна

- Оказывается, резолюция для логики предикатов обладает свойством полноты и непротиворечивости только для определенного вида дизъюнктов

- ***Фразой Хорна*** *(или* ***дизъюнктом Хорна****, Definite Clause) называется дизъюнкт вида*
- $A \vee \neg B_1 \vee \ldots \vee \neg B_n$
- $B_1 \wedge \ldots \wedge B_n \supset A$

Почему дизъюнкты Хорна?

- Нет возможности делать заключения с отрицаниями: «это число не делится на 3»
  - Мы всегда имеем дело с «позитивными» утверждениями
- Из множества посылок может быть только одно заключение
  - т.к. если бы было несколько, то возникало бы множество различных вариантов
- Логическое программирование с отрицаниями – позднее в нашем курсе!

SLD-резолюция

- $\neg H(x) \vee M(x)$ (1)
- H(Сократ) (2)
- Попробуем найти все решения, для этого добавим отрицание запроса $\neg M(z)$ (3)
- 1 шаг: начинаем с (3), подходит (1) {z/x}
- 2 шаг: $\neg H(z)$ (4)
- 3 шаг: (4) и (2), {z/Сократ}

Более сложный пример

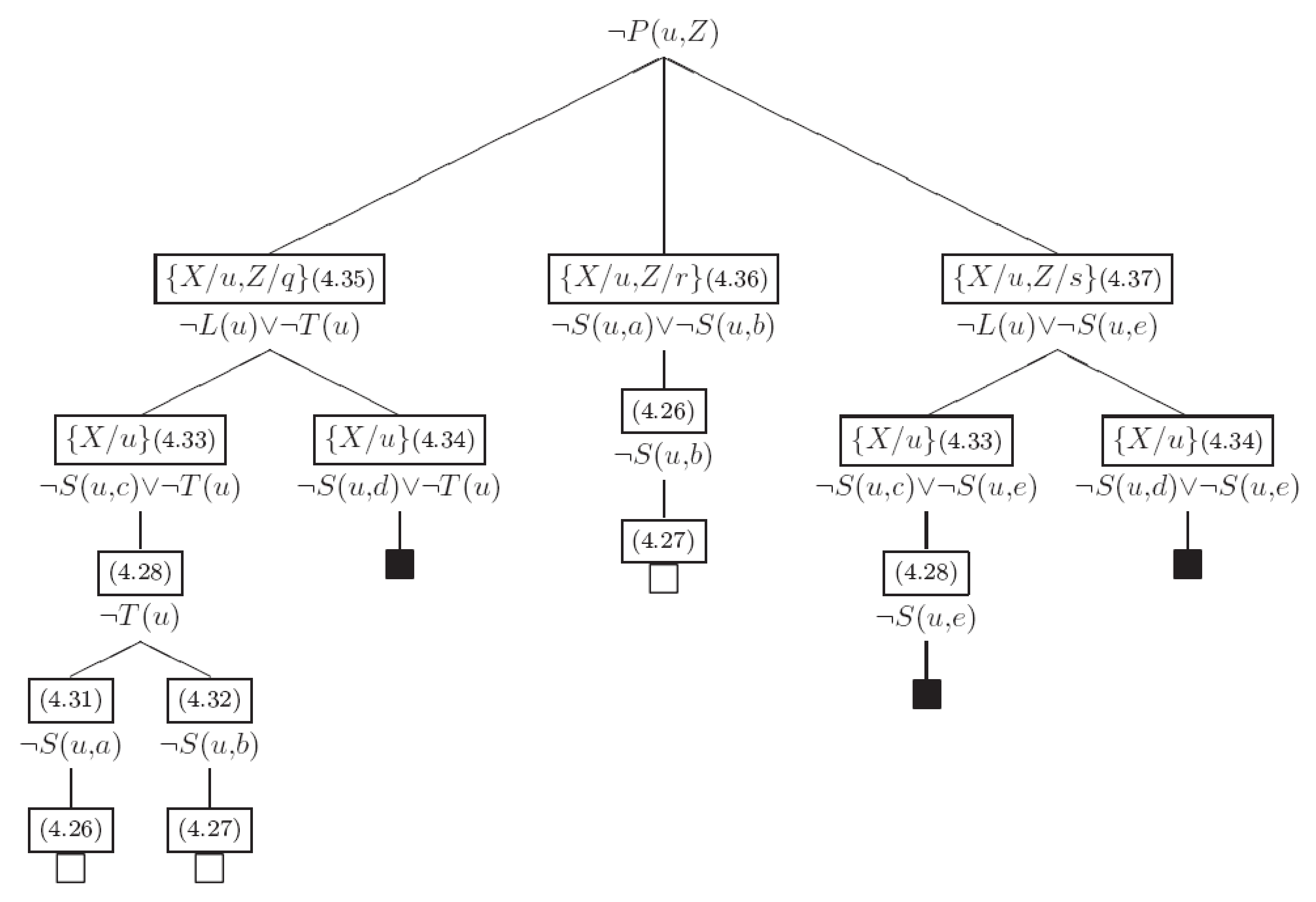
```
speciality(X,tech_translator) :-
       studied_languages(X),studied_technical(X).
speciality(X,programmer) :-
       studied(X,mathematics),studied(X, compscience).
speciality(X,lit_translator) :-
studied_languages(X),studied(X,literature).

studied_technical(X) :- studied(X,mathematics).
studied_technical(X) :- studied(X,compscience).
studied_languages(X) :- studied(X,english).
studied_languages(X) :- studied(X,german).

studied(petya,mathematics).  studied(vasya,german).

studied(petya,compscience).
       studied(vasya,literature).
studied(petya,english).
```

Более сложный пример

$$S(u,a) \wedge S(u,b) \wedge S(u,c)$$
$$S(v,d) \wedge S(v,e)$$
$$(\forall X) S(X,a) \vee S(X,b) \supset T(X)$$
$$(\forall X) S(X,c) \vee S(X,d) \supset L(X)$$
$$(\forall X) L(X) \wedge T(X) \supset P(X,q)$$
$$(\forall X) S(X,a) \wedge S(X,b) \supset P(X,r)$$
$$(\forall X) L(X) \wedge S(X,e) \supset P(X,s)$$

$$S(u,a)$$
$$S(u,b)$$
$$S(u,c)$$
$$S(v,d)$$
$$S(v,e)$$
$$\neg S(X,a) \vee T(X)$$
$$\neg S(X,b) \vee T(X)$$
$$\neg S(X,c) \vee L(X)$$
$$\neg S(X,d) \vee L(X)$$
$$\neg L(X) \vee \neg T(X) \vee P(X,q)$$
$$\neg S(X,a) \vee \neg S(X,b) \vee P(X,r)$$
$$\neg L(X) \vee \neg S(X,e) \vee P(X,s)$$

Дерево вывода



Стратегия поиска решений

- Для поиска решений в полученном дереве нужно совершить его обход
- **Обход в глубину** – движение по левой ветке с возвратом в случае необходимости
- **Обход в ширину** – одновременное рассмотрение нескольких решений

Полнота и достоверность

- **Теорема** О достоверности SLD-резолюции. *Пусть P - множество фраз Хорна, G - запрос. Если в процессе SLD-резолюции было получено решение в виде подстановки θ, то формула G θ будет являться логическим следствием P (P $\models$ Gθ).*

- **Теорема** О полноте SLD-резолюции. *Пусть P - логическая программа (множество фраз Хорна), G - запрос. Если P $\models$ Gσ, то существует SLD-резолютивный опровергающий вывод G с вычисляемой подстановкой θ, такой, что Gθ является частным случаем Gσ.*

А как же пример с предком?

- На прошлой лекции мы видели, что в некоторых случаях решение не находится!

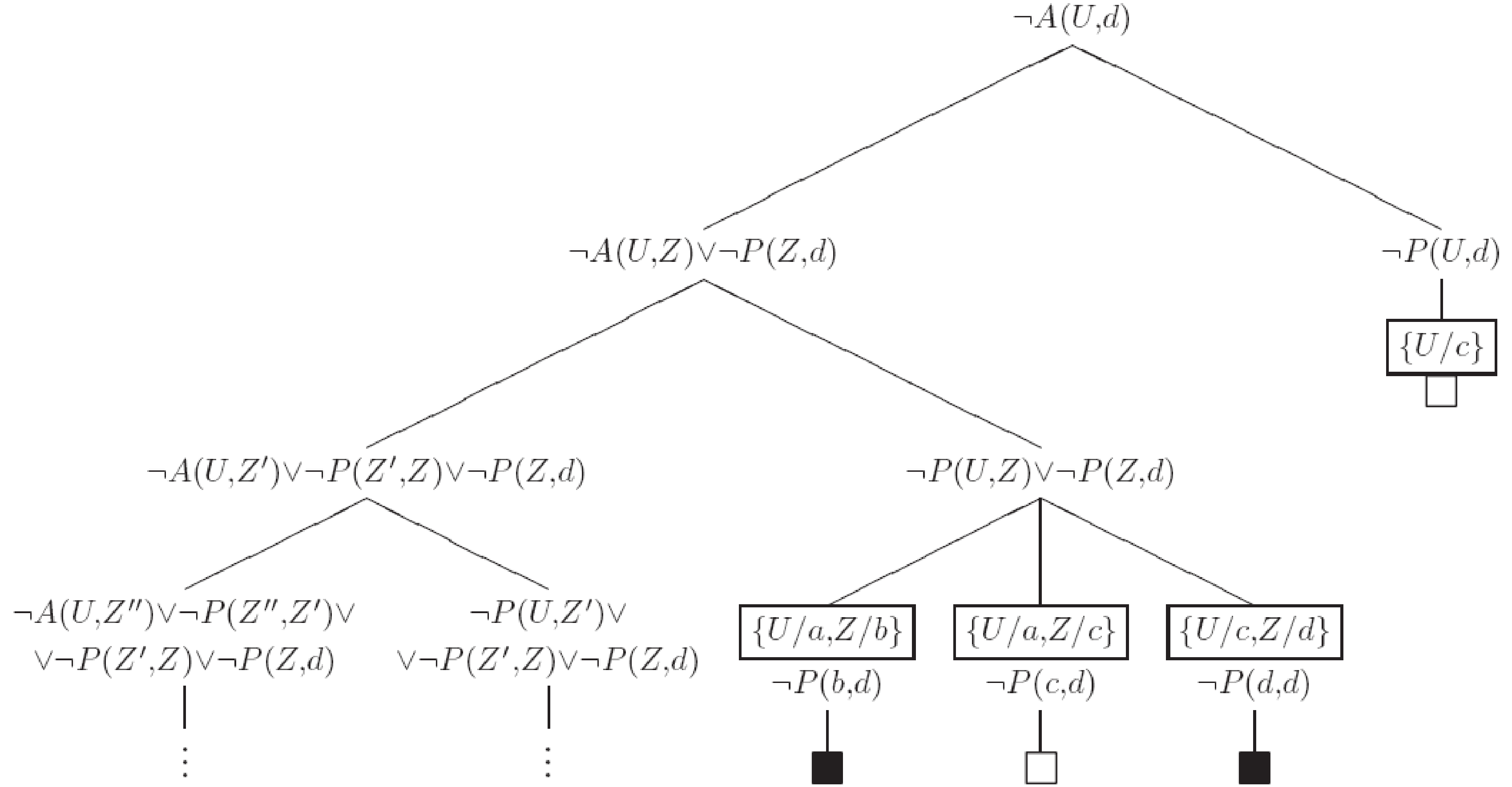
$$(\forall X)(\forall Y)(((\exists Z)A(X,Z) \wedge P(Z,Y)) \vee P(X,Y)) \supset A(X,Y)$$

$$\neg A(X,Z) \vee \neg P(Z,Y) \vee A(X,Y)$$
$$\neg P(X,Y) \vee A(X,Y)$$

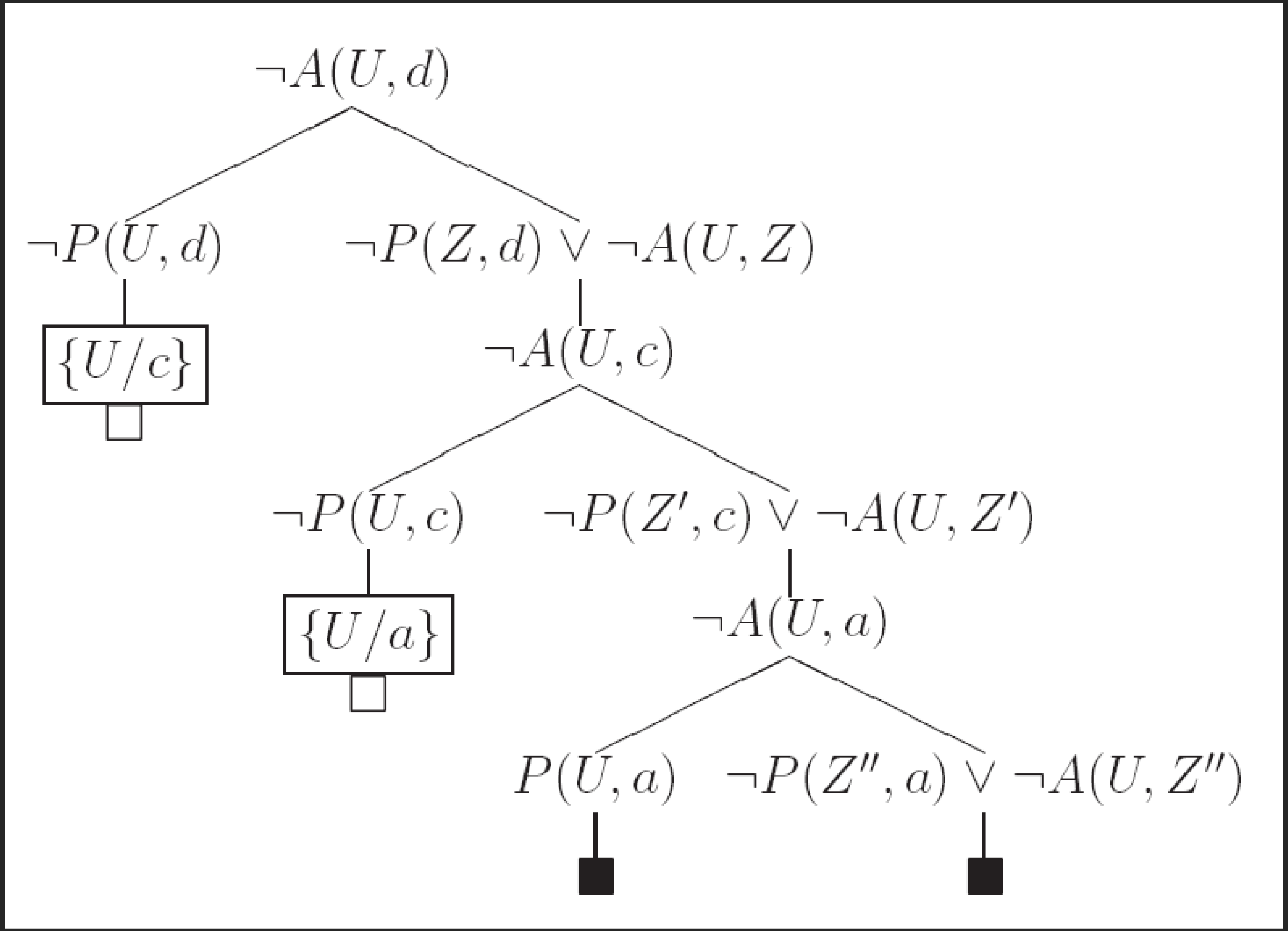
$$P(a,b)$$
$$P(a,c)$$
$$P(c,d)$$

$$\neg A(U,d)$$

Бесконечное дерево SLD-резолюции



Конечное дерево для эквивалентной программы



Еще раз о декларативной и процедурной семантике

- Теорема о полноте гарантирует присутствие всех решений в (потенциально бесконечном) дереве вывода
- Более «правильный» обход дерева позволяет во многих случаях находить такие решения даже для бесконечных деревьев
  - Хотя всегда можно построить контр-пример!
- Процедурная семантика языка должны учитывать стратегии генерации и обхода дерева вывода

Вопросы?

- http://t.me/log_pro
- https://soshnikov.com/courses/logpro/