

Степан Остапенко

# Проверка выполнимости SMT-формул с помощью нейронных сетей

Выпускная квалификационная работа

Научный руководитель: Дмитрий Сергеевич Шалымов

Научный консультант: Дмитрий Аркадьевич Иванов

13.06.2024



Факультет математики и компьютерных наук СПбГУ  
Программа «Современное программирование»

# Введение в предметную область

$$(x^2 + y - z^2 = 2) \wedge (x^2 + 2y - 3z^2 \leq 2) \wedge (x^2 + 2y - 4z^2 \geq 1),$$

$$\text{где } x, y, z \in \mathbb{Z} \longrightarrow x = 1, y = 2, z = -1.$$

- $\mathbb{B}$  – true, false, логические операции;
- $\mathbb{Z}$  – целые числа, арифметика, сравнения;
- $\mathbb{R}$  – вещественные числа, арифметика, сравнения;
- BitVec – битовые векторы, логические и арифметические операции;



# Зачем это нужно

## Применения

- составление расписания и планирование,
- верификация АО и ПО,
- вывод типов,
- **символьное исполнение**,
- статический анализ программ.

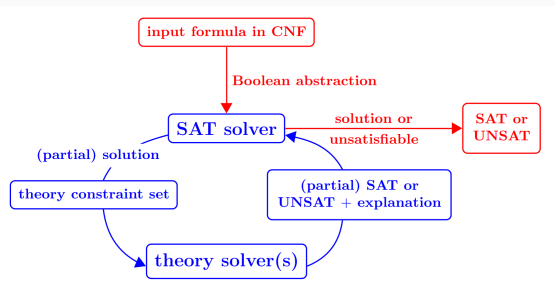
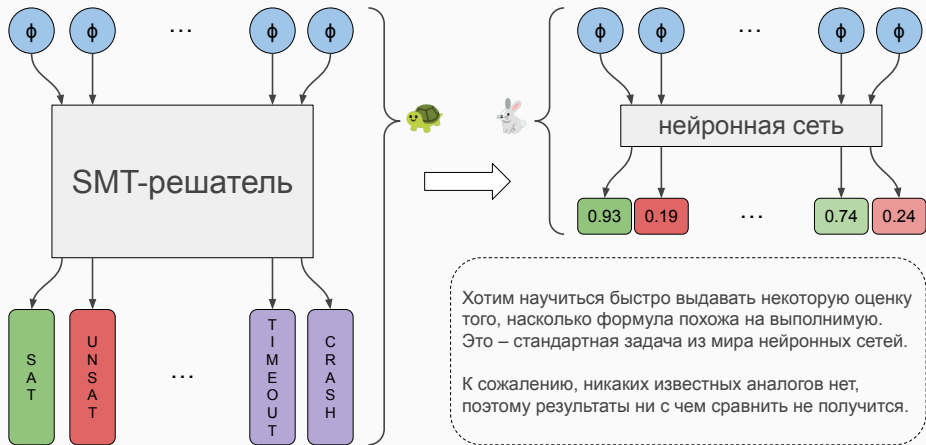


Рис. 1: Схема SMT-решателя [1].

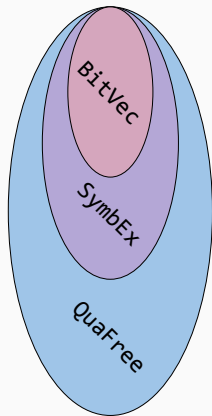


# Постановка задачи



# Данные для обучения и оценки качества

## SMT-COMP [2]



## USVM [3] [4]

Наборы данных, состоящие из формул, возникших в процессе работы символического движка:

- обучающий набор — запуск движка на нём же;
- 18 валидационных — запуск движка на разных проектах на JVM-языках.



# Метрики

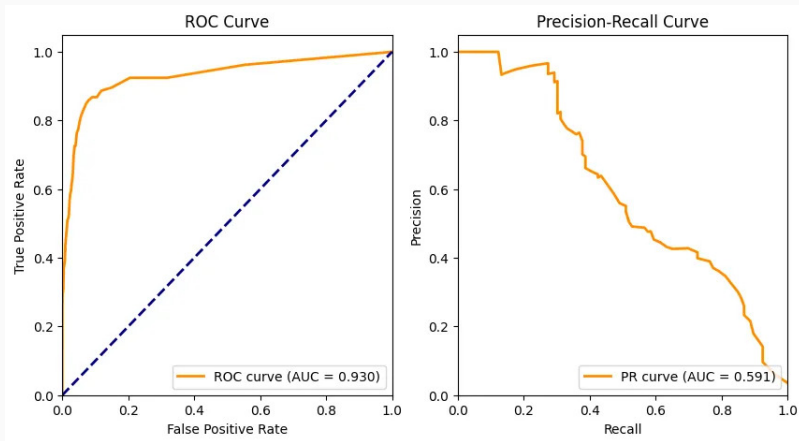
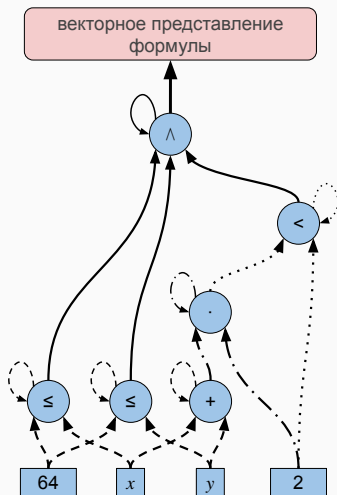


Рис. 2: Площадь под ROC-кривой и площадь под PR-кривой [5].

# Архитектура

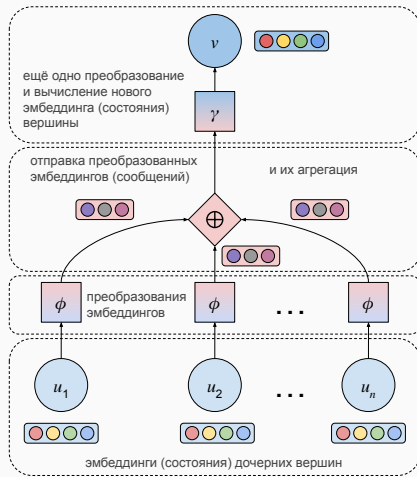
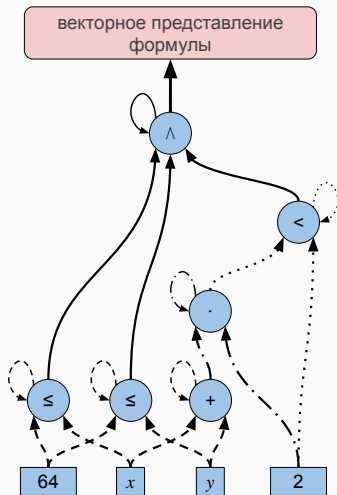


В качестве графа вычислений берём AST формулы, превращённое в ориентированный ациклический граф.

Слева пример для формулы  
 $(64 \leq x) \wedge (64 \leq y) \wedge ((x + y) \cdot 2 < 2)$

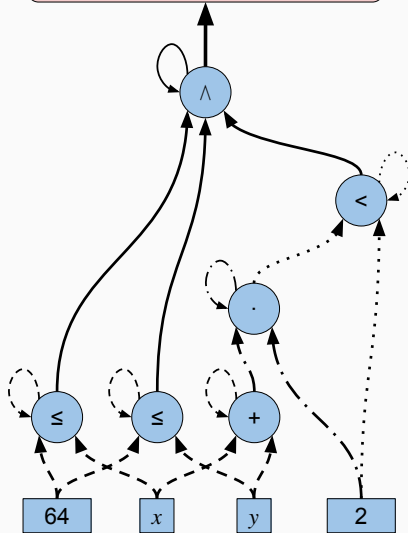


# Архитектура

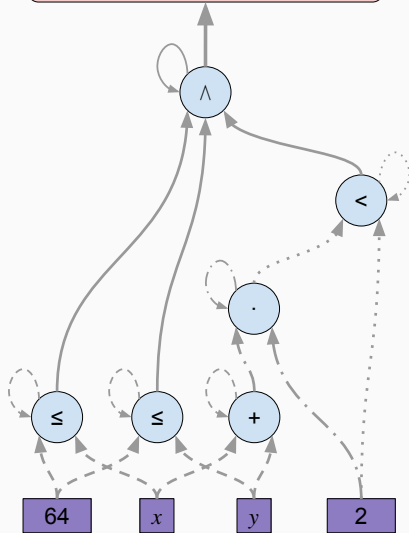




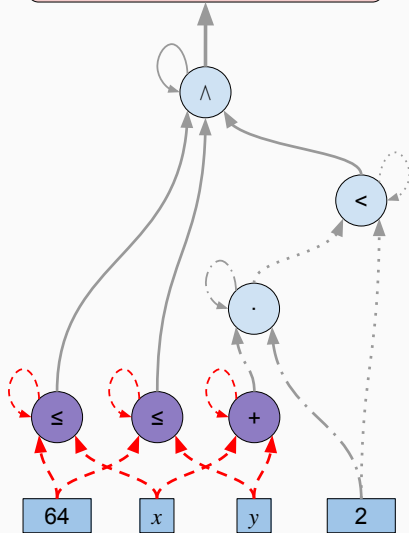
векторное представление  
формулы



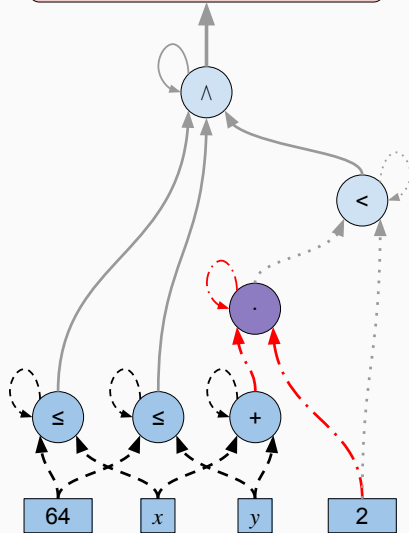
векторное представление  
формулы



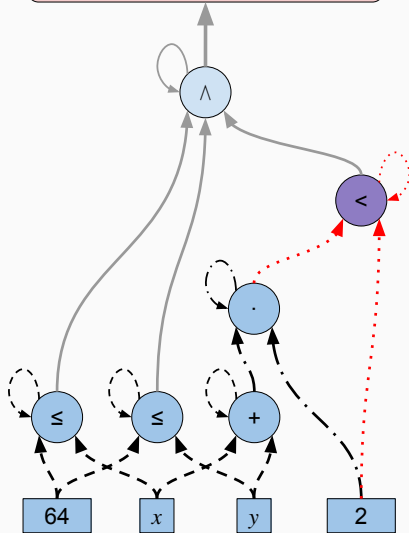
векторное представление  
формулы



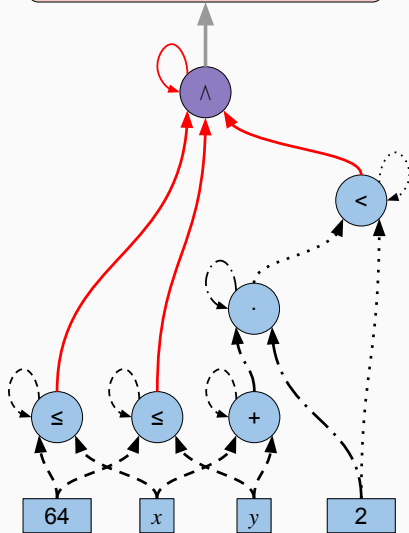
векторное представление  
формулы



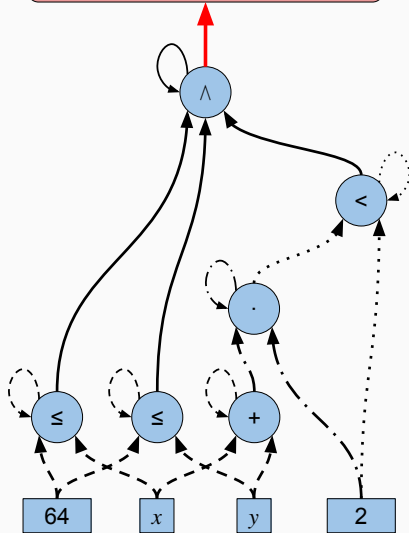
векторное представление  
формулы



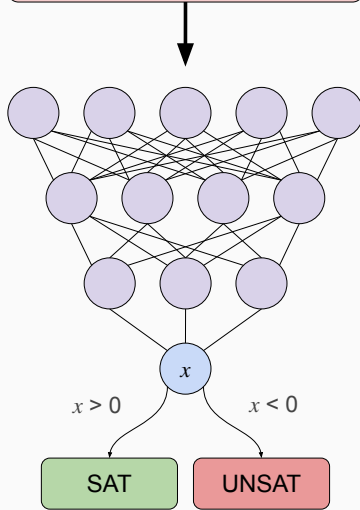
векторное представление  
формулы



векторное представление  
формулы



векторное представление  
формулы





# Как собрать детей по-быстрому?

SAGE Convolution [6]:

$$x'_v = \sigma \left( W_3 \cdot \left( W_1 \cdot x_v + W_2 \cdot \text{MEAN}_{u \in \mathcal{N}(v)} x_u \right) + b \right)$$



# Как собрать детей по-быстрому?

Transformer Convolution [7]:

$$\alpha_u = \text{SOFTMAX}_{u \in \mathcal{N}(v)} \left[ \frac{(W_3 \cdot x_v)^T (W_4 \cdot x_u)}{\sqrt{d}} \right]$$

$$x'_v = W_1 \cdot x_v + \sum_{u \in \mathcal{N}(v)} \alpha_u W_2 \cdot x_u$$



# Начальные состояния

Проблема — откуда взять начальные состояния?

- Замена  $m, k, n \rightarrow \text{Var}[\text{Int}]$ ;  $2.8, 2.4, 3.0 \rightarrow \text{Val}[\text{Real}]$ ;
- 

*Для переменных:*

- Случайные выученные векторы;
- Позиционное кодирование;

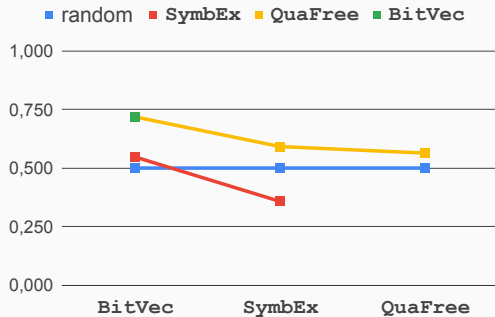
*Для констант:*

- Векторы с учётом групп (бинов) [8];
- Преобразование с Fourier Feature Mapping [8];

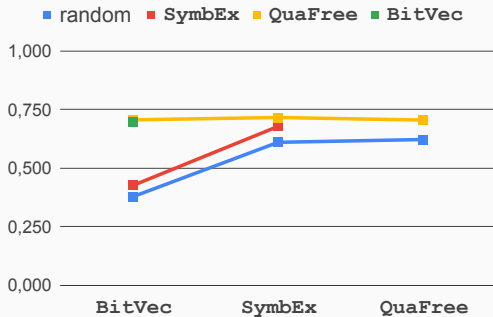


# Результаты SAGE Convolution (SMT - COMP)

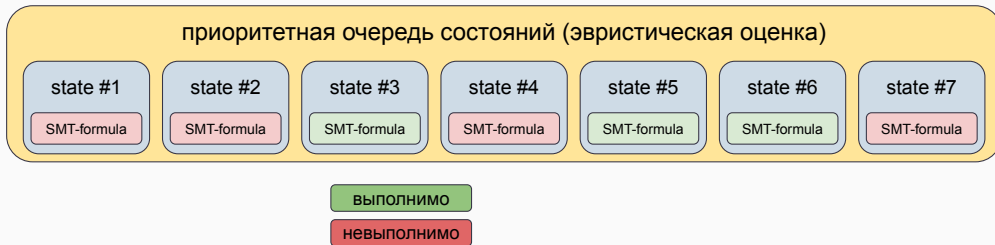
## ROC-AUC



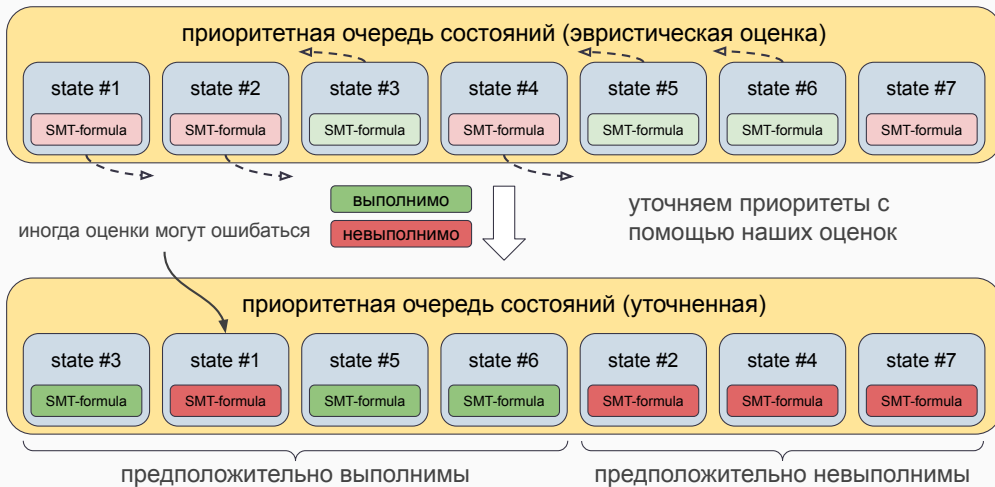
## Avg. Precision



# Символьное исполнение

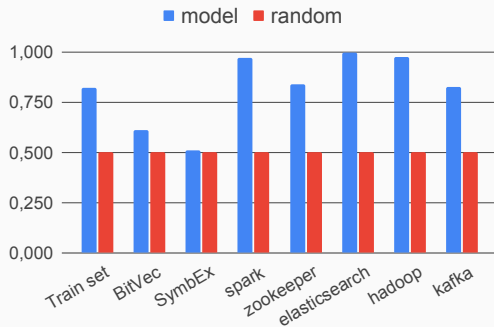


# Символьное исполнение

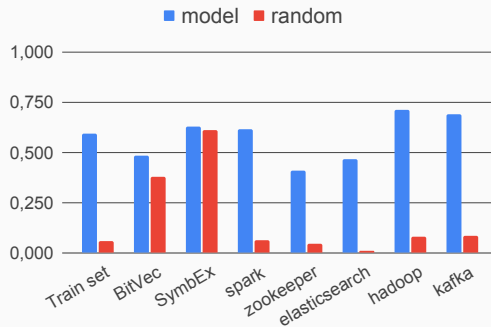


# Результаты SAGE Convolution (USVM)

ROC-AUC



Avg. Precision



# Дальнейшее развитие

- Подбор гиперпараметров.
- Архитектурные улучшения.
- Аугментации данных.
- Постановка задачи ранжирования.
- Эффективная реализация и внедрение.





# Итоги работы

1. Спроектирована, обучена и протестирована первая нейросеть для предсказания выполнимости SMT-формул.
  2. Качество решения общей задачи плохое, но в случае одного из возможных практических применений результаты выглядят многообещающе.
  3. Есть много идей для дальнейшего развития проекта.
- 

Степан Остапенко. tg: @flaax,  
stepanos2002@gmail.com



<https://qrco.de/bf9EBt>



## Источники i

- [1] SC<sup>2</sup>: Satisfiability Checking Meets Symbolic Computation – Scientific Figure on ResearchGate. Available from:  
[https://www.researchgate.net/figure/The-functioning-of-SMT-solvers\\_fig1\\_305214266](https://www.researchgate.net/figure/The-functioning-of-SMT-solvers_fig1_305214266) (дата  
обр. 11.06.2024).
- [2] SMT-COMP 2023 benchmarks. URL: <https://smt-comp.github.io/2023/benchmarks.html>  
(дата обр. 20.05.2024).

## Источники ii

- [3] USVM on github. URL:  
<https://github.com/UnitTestBot/usvm> (дата обр.  
10.06.2024).
- [4] Поспелов Сергей Андреевич. «Проектирование и  
разработка универсальной символьной виртуальной  
машины». *Бакалаврская выпускная квалификационная  
работа*. URL: <http://hdl.handle.net/11701/42155>.  
СПбГУ, 2023.

## Источники iii

- [5] URL: <https://juandelacalle.medium.com/how-and-why-i-switched-from-the-roc-curve-to-the-precision-recall-curve-to-analyze-my-imbalanced-6171da91c6b8> (дата обр. 27.05.2024).
- [6] William L. Hamilton et al. «Inductive Representation Learning on Large Graphs». *ArXiv abs:1706.02216* (2017).

## Источники iv

- [7] Yunsheng Shi et al. «Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification». *ArXiv abs:2009.03509* (2021).
- [8] Yury Gorishniy, Ivan Rubachev and Artem Babenko. «On Embeddings for Numerical Features in Tabular Deep Learning». *Neural Information Processing Systems (NeurIPS '2022)*.

## Источники v

- [9] Alessandro Sperduti and Antonina Starita. «Supervised neural networks for the classification of structures». In *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 714–735, May 1997. IEEE, 1997.
- [10] Paolo Frasconi, Marco Gori and Alessandro Sperduti. «A general framework for adaptive processing of data structures». In *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 768–786, Sept. 1998. IEEE, 1998.

## Источники vi

- [11] Mislav Balunovic, Pavol Bielik and Martin T. Vechev. «Learning to Solve SMT Formulas». *Advances in Neural Information Processing Systems 31* (2018).
- [12] Jan Hůla, David Mojžíšek and Mikoláš Janota. «Graph Neural Networks for Scheduling of SMT Solvers». *IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI '2021)*, pp. 447–451. IEEE, 2021.

## Источники vii

- [13] Daniel Selsam, Matthew Lamm, Benedikt Bünz, Percy Liang, Leonardo Mendonça de Moura and David L. Dill. «Learning a SAT Solver from Single-Bit Supervision». *ArXiv abs:1802.03685* (2018).
- [14] Валеев Азат Рустамович. «Добавление признаков композиций в функцию потерь модели MusicTransformer для настраиваемой генерации музыки». *Бакалаврская выпускная квалификационная работа*. СПбГУ, 2023.

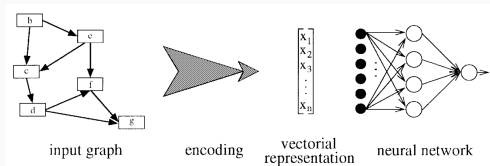
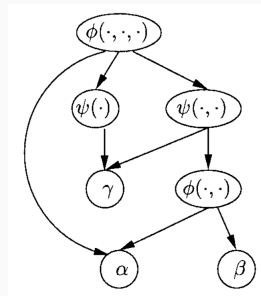
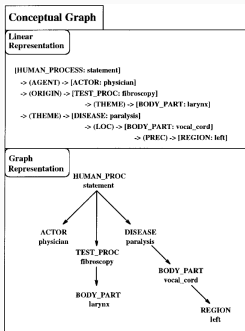


# Параметры обучения

размерность эмбединга	32
функция потерь	<i>Cross-Entropy</i>
оптимизатор	<i>AdamW</i>
шаг обучения	$10^{-4}$
штраф за веса	$10^{-3}$
расписание	<i>Reduce LR On Plateau</i>
val выборка	15%
test выборка	10%
количество эпох	100

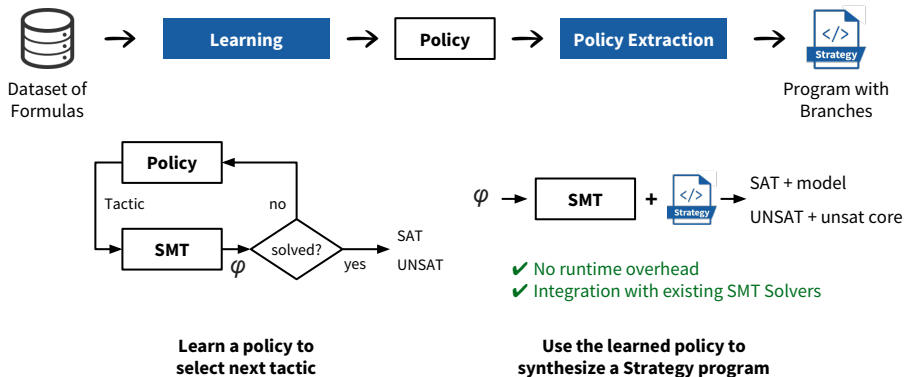


# RvNN [9] [10]



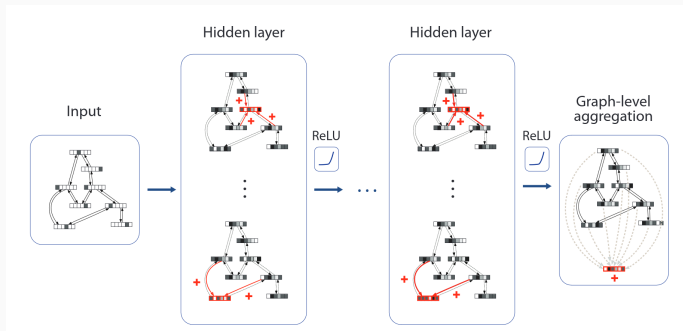
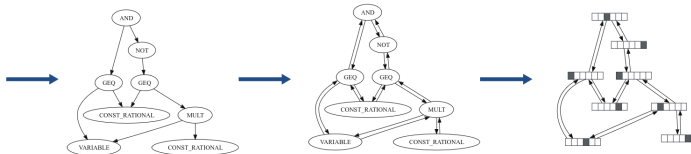
# FastSMT [11]

## Learning to Solve SMT Formulas

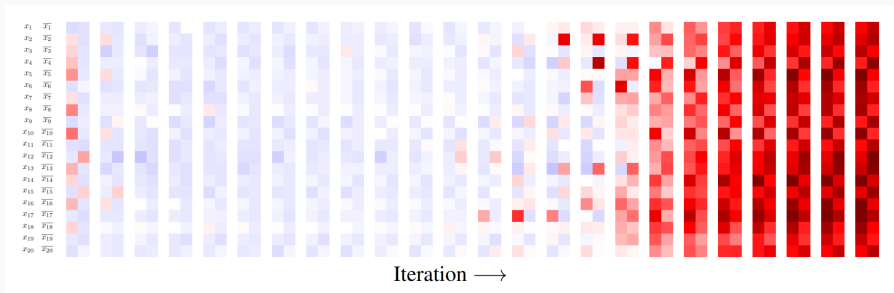
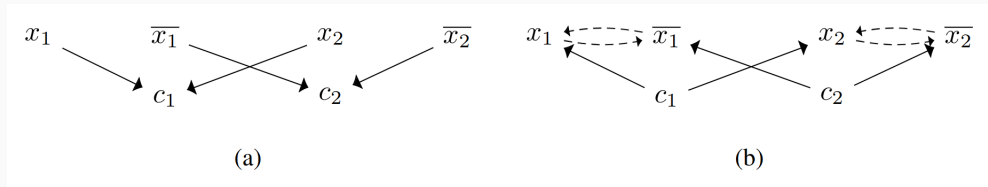


# GNN for Scheduling of SMT Solvers [12]

```
(set-info :smt-lib-version 2.6)
(set-logic QF_NRA)
(set-info :category "crafted")
(set-info :status sat)
(declare-fun a () Real)
(assert (and (>= a 3)
  (not (>= (* a 2) 3))))
(check-sat)
(exit)
```



# NeuroSAT [13]



# Параметры датасетов с SMT-COMP

Датасет	Количество формул	Средний размер формулы	Средняя глубина формулы	Доля выполнимых формул
BitVec	33 797	1181.92	85.45	0.378
SymbEx	85 078	669.46	49.24	0.610
QuaFree	123 396	965.16	48.71	0.622

**Таблица 1:** Параметры датасетов, полученных из данных с соревнования SMT-COMP 2023 [2].



## Параметры датасетов, собранных с USVM

Датасет	Количество формул	Средний размер формулы	Средняя глубина формулы	Доля выполнимых формул
u-test	153 778	522.84	5.18	0.038
the-alg	181 633	277.03	23.94	0.066
u-core	192 744	179.42	12.19	0.066

**Таблица 2:** Параметры тренировочных датасетов, собранных в процессе работы символьного движка USVM [3] [4].

