# Math Word Problems Solving

2021.01.16 许晓丹

# 从数据集看任务分类

- Algebra word problem——不涉及方程的应用题

  例：买一套住房，售价28万元，每年付2万元，多少年后才能付清?

  代表数据集：Math23K

- Number word problem

  例：有两个数，其中一个数是另一个数的两倍，且相差10，这两个数分别是多少?

  代表数据集：Dolphin1878

- 应用的方程组问题

  代表数据集：Dolphin18K（有错误）/HMWP

# Algebra word problem——Tree Decoder

Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)

## A Goal-Driven Tree-Structured Neural Model for Math Word Problems

**Zhipeng Xie**[*†] and **Shichao Sun**[†]
Shanghai Key Laboratory of Data Science, Fudan University
School of Computer Science, Fudan University
{xiezp, scsun17}@fudan.edu.cn

# Motivation

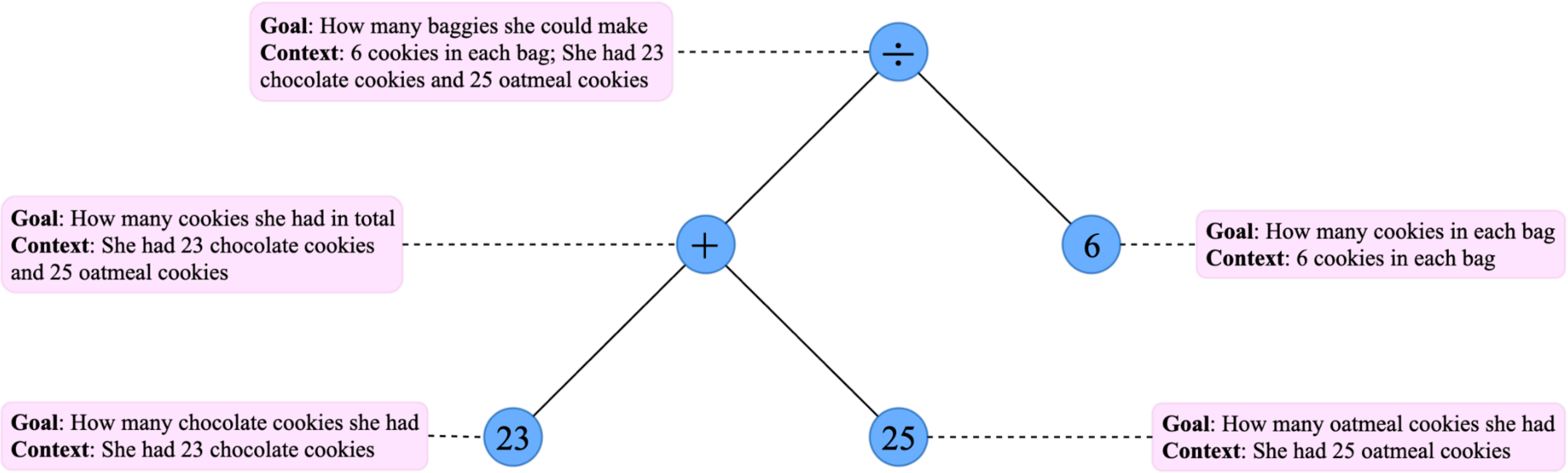



Figure 1: The expression tree of the problem in Table 1

模仿人的思维方式：当人们解决数学问题时，先有一个总目标，然后逐级降解

# Model——prefix

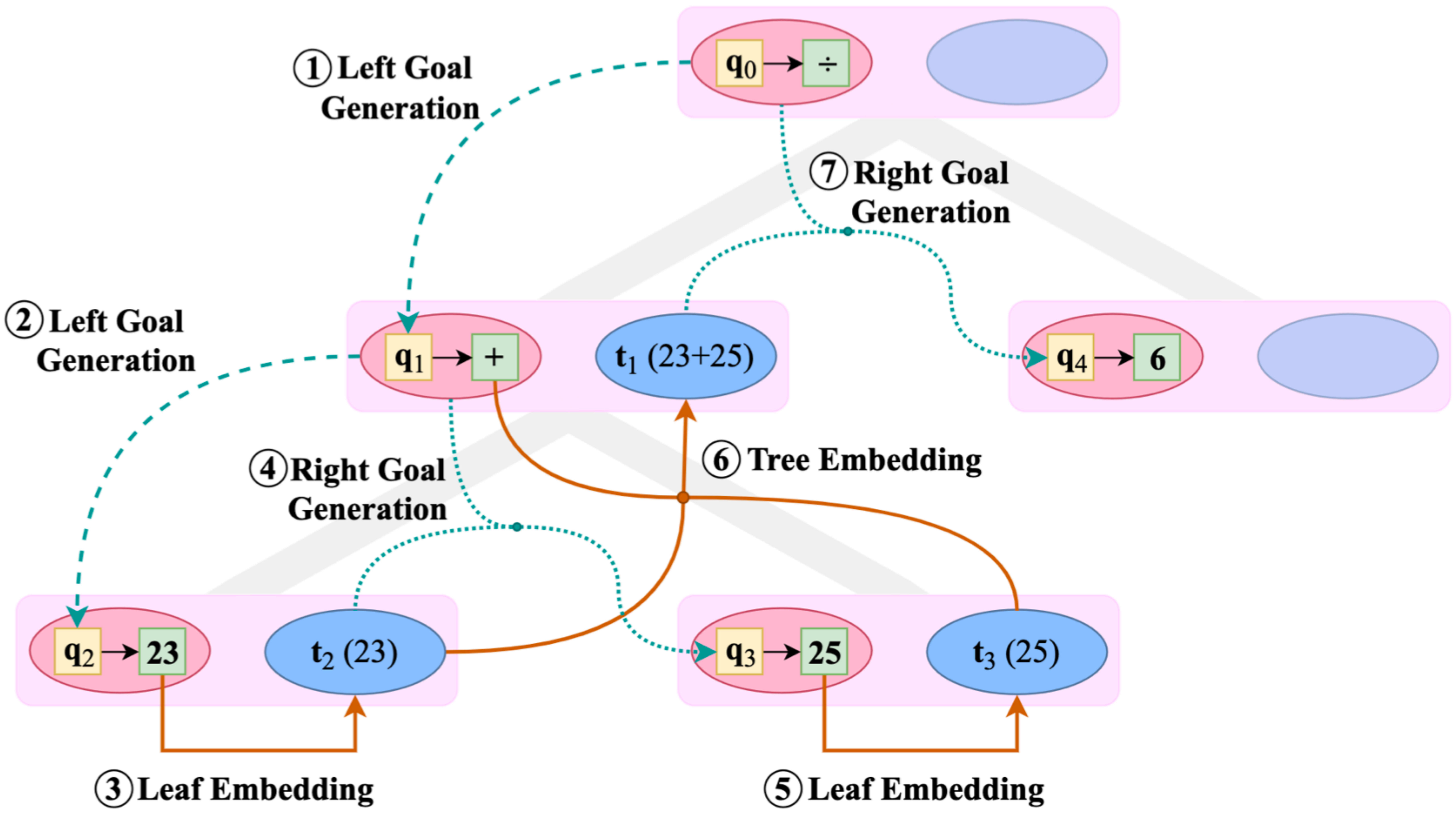



Figure 2: Goal-driven Tree-structured Model

# Model

表达树中的每个节点由三部分组成：

a goal vector q, a token yˆ, and a subtree embedding t of n.

$$\mathrm{s}(y|\mathbf{q},\mathbf{c},P) = \mathbf{w}_n^\top \tanh\left(\mathbf{W}_s\left[\mathbf{q},\mathbf{c},\mathbf{e}(y|P)\right]\right)$$

$$\mathrm{prob}(y|\mathbf{q},\mathbf{c},P) = \frac{\exp\left(\mathrm{s}\left(y|\mathbf{q},\mathbf{c},P\right)\right)}{\sum_i \exp\left(\mathrm{s}\left(y_i|\mathbf{q},\mathbf{c},P\right)\right)}$$

# Left/Right Sub-Goal Generation——自顶向下

$$
\begin{aligned}
o_l &= \sigma\left(\mathbf{W}_{ol}\left[\mathbf{q}, \mathbf{c}, \mathbf{e}(\hat{y}|P)\right]\right) \\
C_l &= \tanh\left(\mathbf{W}_{cl}\left[\mathbf{q}, \mathbf{c}, \mathbf{e}(\hat{y}|P)\right]\right) \\
\mathbf{h}_l &= o_l \odot C_l \\
g_l &= \sigma\left(\mathbf{W}_{gl}\mathbf{h}_l\right) \\
Q_{le} &= \tanh\left(\mathbf{W}_{le}\mathbf{h}_l\right) \\
\mathbf{q}_l &= g_l \odot Q_{le}
\end{aligned}
$$

$$
\begin{aligned}
o_r &= \sigma\left(\mathbf{W}_{or}\left[\mathbf{q}, \mathbf{c}, \mathbf{e}(\hat{y}|P)\right]\right) \\
C_r &= \tanh\left(\mathbf{W}_{cr}\left[\mathbf{q}, \mathbf{c}, \mathbf{e}(\hat{y}|P)\right]\right) \\
\mathbf{h}_r &= o_r \odot C_r \\
g_r &= \sigma\left(\mathbf{W}_{gr}\left[\mathbf{h}_r, \mathbf{t}_l\right]\right) \\
Q_{re} &= \tanh\left(\mathbf{W}_{re}\left[\mathbf{h}_r, \mathbf{t}_l\right]\right) \\
\mathbf{q}_r &= g_r \odot Q_{re}
\end{aligned}
$$

# Subtree Embedding via Recursive Neural Network——自下而上

$$\mathbf{t} = \begin{cases} comb(\mathbf{t}_l, \mathbf{t}_r, \hat{y}) & \text{if } \hat{y} \in V_{op} \\ \mathbf{e}(\hat{y}|P) & \text{if } \hat{y} \in n_P \cup V_{con} \end{cases}$$

$$\begin{aligned} comb(\mathbf{t}_l, \mathbf{t}_r, \hat{y}) &= g_t \odot C_t \\ g_t &= \sigma\left(\mathbf{W}_{gt}\left[\mathbf{t}_l, \mathbf{t}_r, \mathbf{e}(\hat{y}|P)\right]\right) \\ C_t &= \tanh\left(\mathbf{W}_{ct}\left[\mathbf{t}_l, \mathbf{t}_r, \mathbf{e}(\hat{y}|P)\right]\right) \end{aligned}$$

| Model | Accuracy(%) |
|---|---|
| Hybrid model w/ SNI [Wang *et al.*, 2017] | 64.7 |
| Ensemble model w/ EN [Wang *et al.*, 2018a] | 68.4 |
| GTS model w/o Subtree Embedding | 70.0 |
| **GTS model** | **74.3** |

Table 2: Model comparison on answer accuracy

# Algebra word problem——Tree Decoder

## A Knowledge-Aware Sequence-to-Tree Network for Math Word Problem Solving

**Qinzhuo Wu, Qi Zhang*, Jinlan Fu, Xuanjing Huang**
Shanghai Key Laboratory of Intelligent Information Processing,
School of Computer Science, Fudan University, Shanghai, China
(qzwu17,qz,fujl16,xjhuang)@fudan.edu.cn

EMNLP 2020

# Motivation

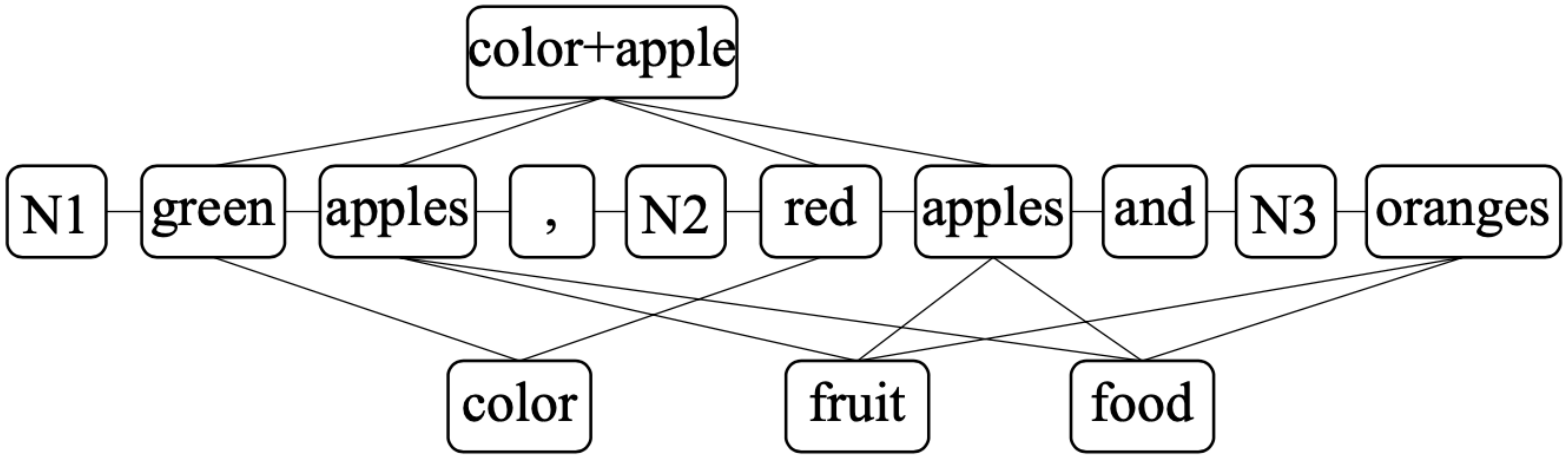
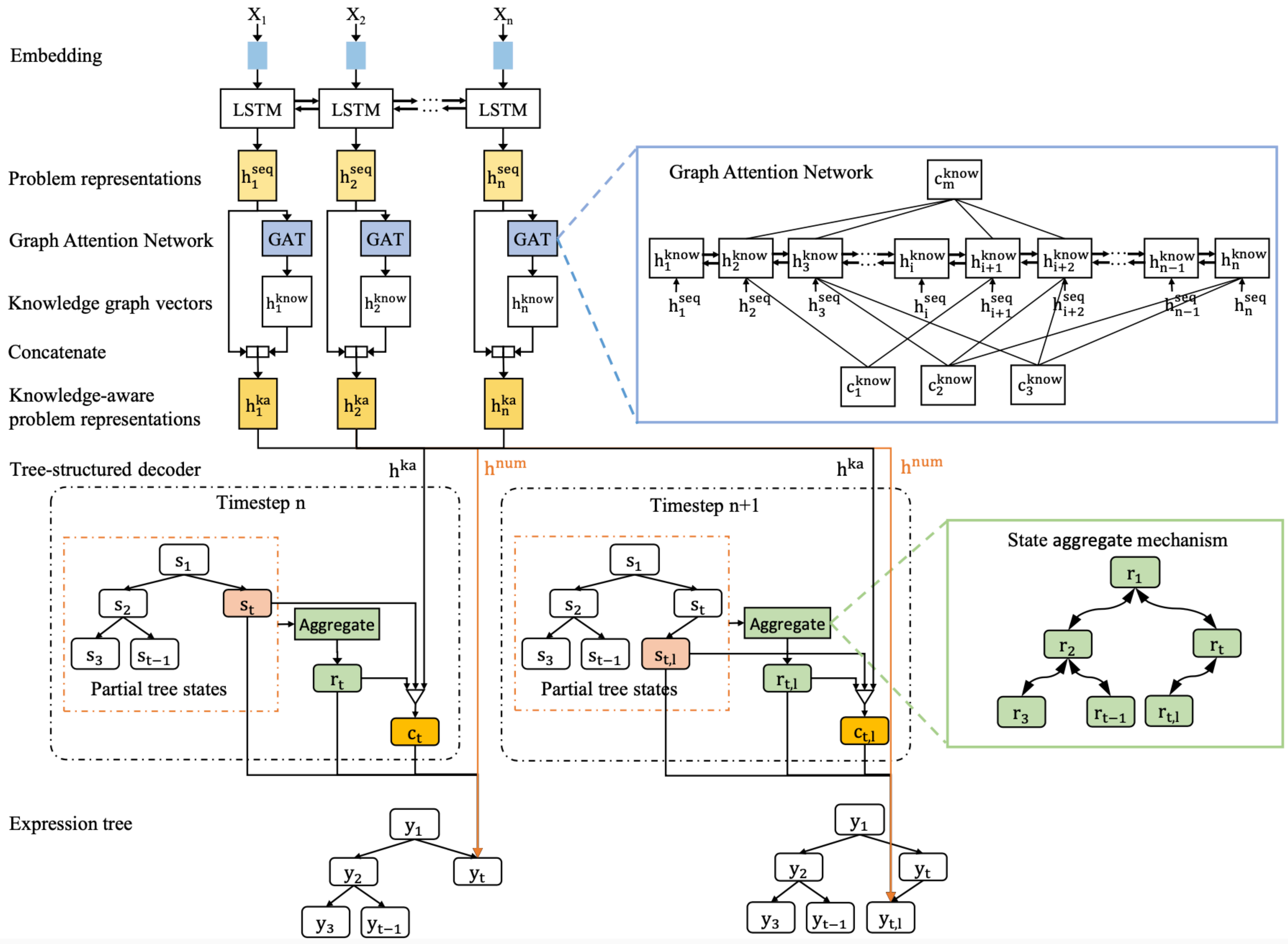



Figure 3: An example of an entity graph

# Model——prefix



# Model——State Aggregation Mechanism

$$\mathbf{s_{t,l}} = \sigma(\mathrm{W}_{\mathrm{left}}[\mathbf{s_t} : \mathbf{c_t} : \mathbf{r_t} : (\mathbf{e}(y_t)]),$$

$$\mathbf{s_{t,r}} = \sigma(\mathrm{W}_{\mathrm{right}}[\mathbf{s_t} : \mathbf{c_t} : \mathbf{r_t} : (\mathbf{e}(y_t)]),$$

| Models | Accuracy |
|---|---|
| DNS (Wang et al., 2017) | 58.1% |
| DNS+Retrieval (Wang et al., 2017) | 64.7% |
| Bi-LSTM (Wang et al., 2018a) | 66.7% |
| ConvS2S (Wang et al., 2018a) | 64.2% |
| Transformer (Wang et al., 2018a) | 62.3% |
| Ensemble (Wang et al., 2018a) | 68.4% |
| RecursiveNN (Wang et al., 2019) | 68.7% |
| Tree-Decoder (Liu et al., 2019) | 69.0% |
| GTS (Xie and Sun, 2019) | 74.3% |
| **KA-S2T (Our)** | **76.3%** |

Table 1: Answer accuracy of our model and other state-of-the-art models on Math23K dataset.

# Algebra word problem——StackDecoder

**Semantically-Aligned Equation Generation for Solving and Reasoning Math Word Problems**

**Ting-Rui Chiang Yun-Nung Chen**
National Taiwan University, Taipei, Taiwan
r07922052@csie.ntu.edu.tw y.v.chen@ieee.org

NAACL 2019

# Motivation

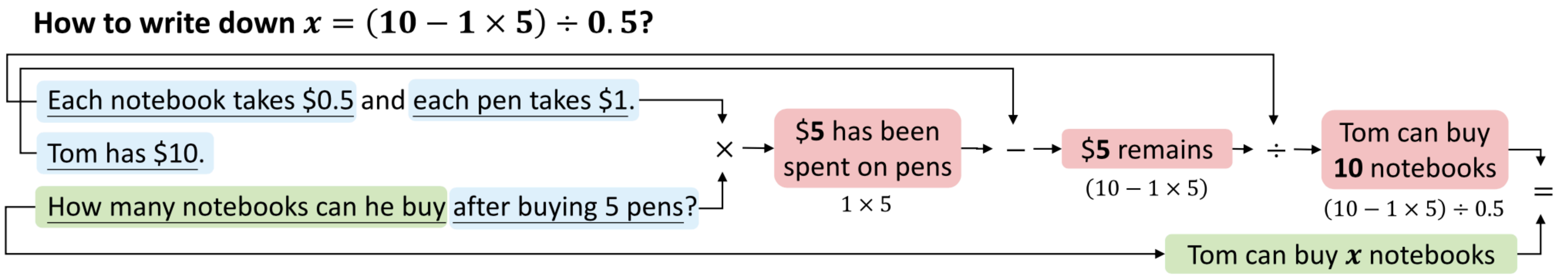



Figure 1: The solving process of the math word problem "*Each notebok takes $0.5 and each pen takes $1. Tom has $10. How many notebook can he buy after buying 5 pens?*" and the associated equation is $x = (10 - 1 \times 5) \div 0.5$. The associated equation is $x = (10 - 1 \times 5) \div 0.5$.

将生成equation的过程看作是连接semantic和symbolic的桥梁；

人们其实给每个number附加语义信息，然后根据number的语义信息选择operator

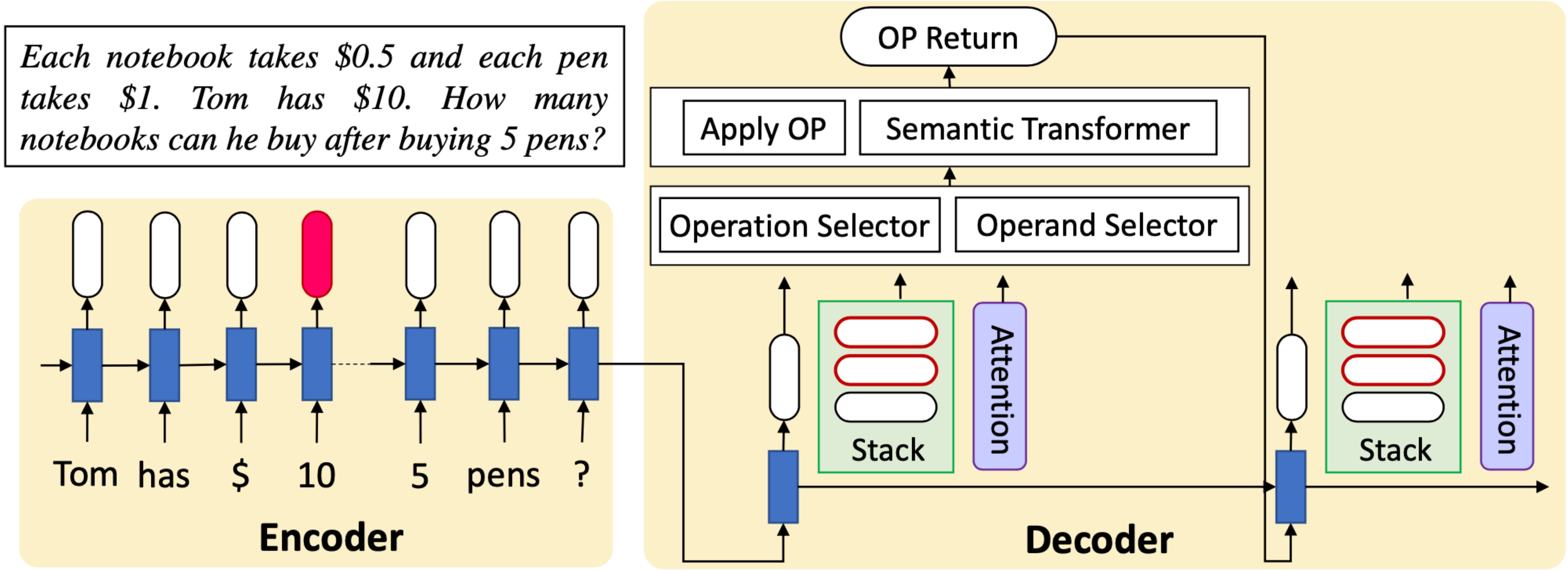



Figure 2: The encoder-decoder model architecture of the proposed neural solver machine.

# Model— —postfix

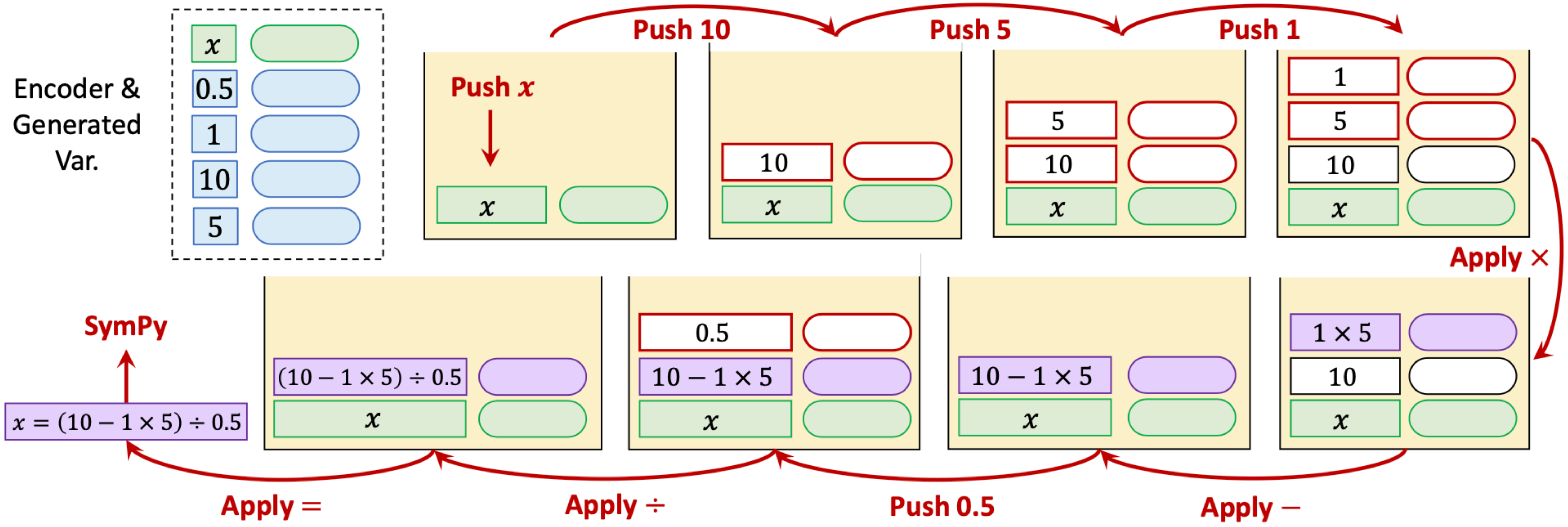



Figure 3: Illustration of the inference process. The purple round blocks denote the transformed semantics, while the green ones are generated by the variable generator.

# Equation Set Problem

## Semantically-Aligned Universal Tree-Structured Solver for Math Word Problems

**Jinghui Qin**[1], **Lihui Lin**[2], **Xiaodan Liang**[1,2,*], **Rumin Zhang**[2], **Liang Lin**[1,2]
[1] Sun Yat-sen University
[2] Dark Matter AI Inc.
qinjingh@mail2.sysu.edu.cn, linlh23@mail2.sysu.edu.cn,
xdliang328@gmail.com, rm_zhang@foxmail.com,
linliang@ieee.org

# Motivation

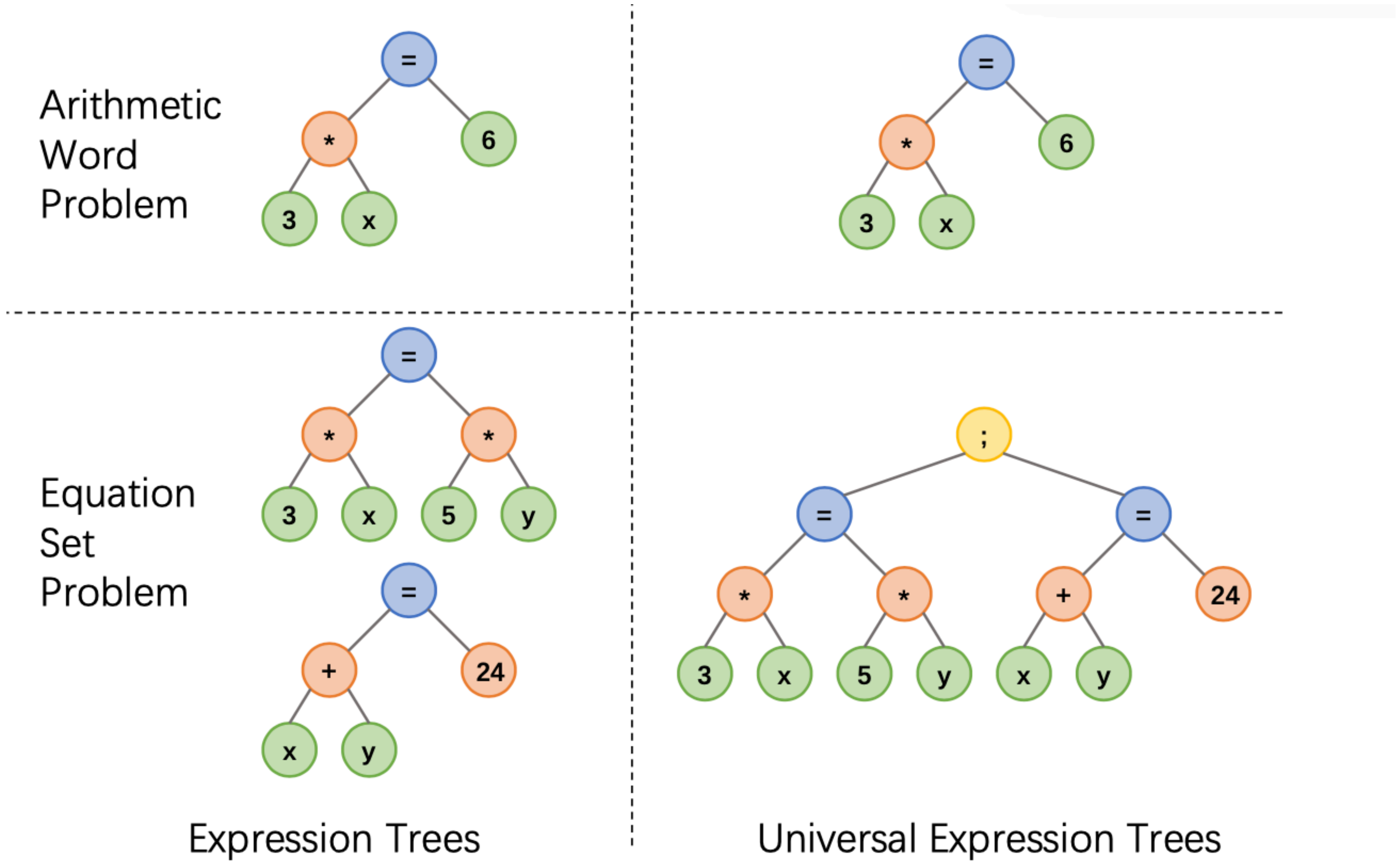
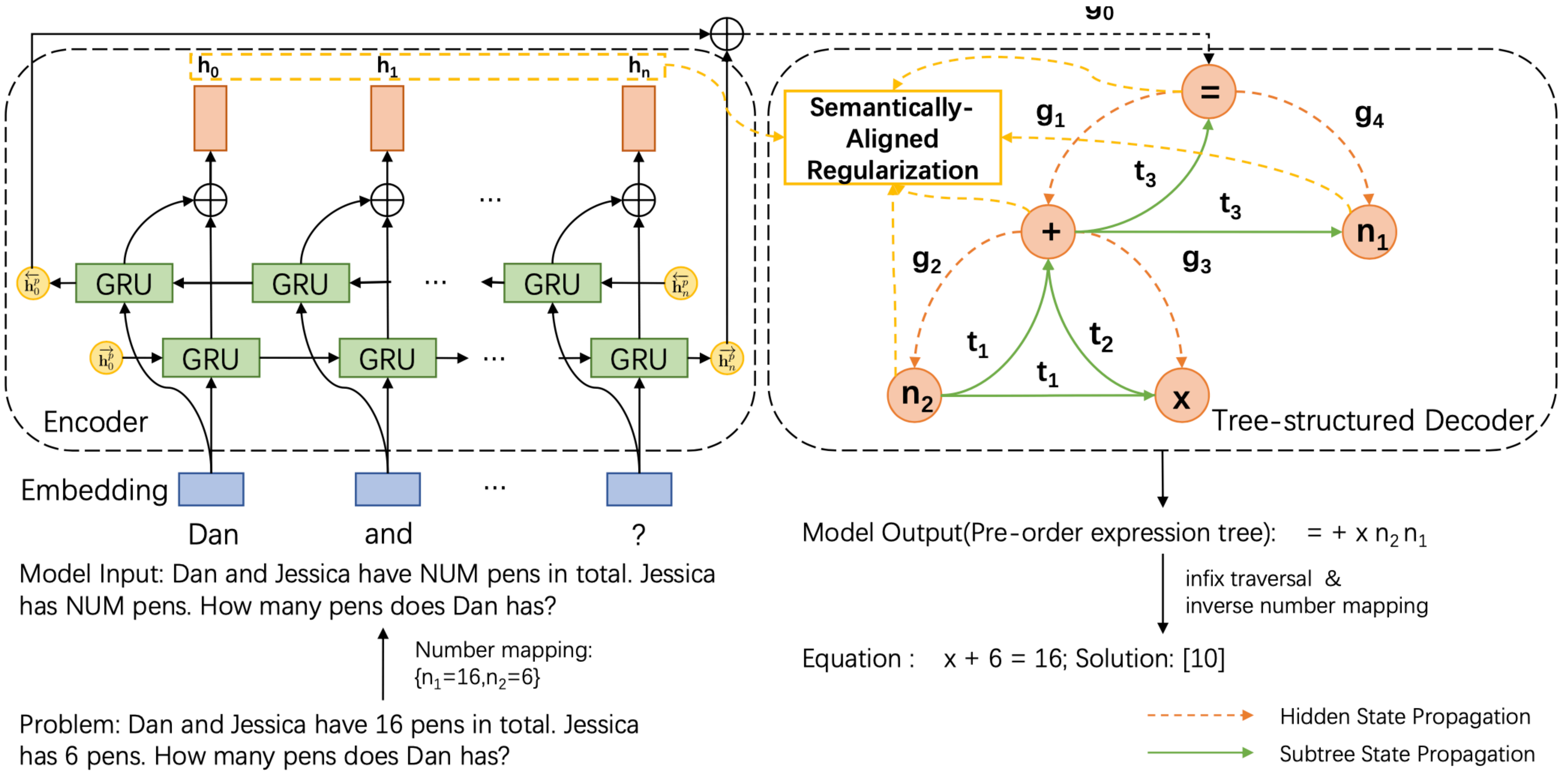



Figure 1: Universal Expression Trees (UET). In our UET representation, multiple expression trees underlying a MWP will be integrated as an universal expression tree (UET) via symbol extension. UET can enable a solver to handle multiple types of MWPs in an unified manner like a single expression tree of an equation.

# Model



g0
h0
h1
hn
Semantically-Aligned Regularization
g1
g4
t3
t3
+
=
n1
g2
g3
t1
t2
t1
n2
x
GRU
GRU
GRU
GRU
GRU
GRU
Encoder
Tree-structured Decoder
Embedding
Dan
and
?
Model Input: Dan and Jessica have NUM pens in total. Jessica has NUM pens. How many pens does Dan has?
Number mapping: {n1=16,n2=6}
Problem: Dan and Jessica have 16 pens in total. Jessica has 6 pens. How many pens does Dan has?
Model Output(Pre-order expression tree): = + x n2 n1
infix traversal & inverse number mapping
Equation : x + 6 = 16; Solution: [10]
Hidden State Propagation
Subtree State Propagation