

1) Analogical Reasoning over Natural Language Paragraphs (2022-2023)



Life is a Circus and We are the Clowns: Automatically Finding Analogies between Situations and Processes

Oren Sultan, Dafna Shahaf

A Base: Animal Cell

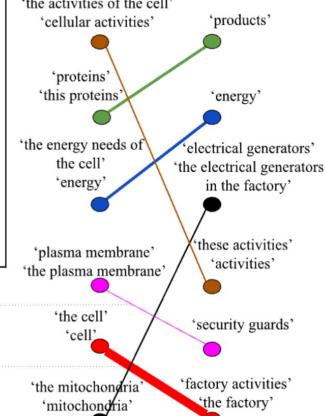
The plasma membrane encloses the animal cell. It controls the movement of materials into and out of the cell. The Nucleus controls the activities of the cell. These cellular activities require energy. The Mitochondria extract energy from food molecules to provide the energy needs of the cell. Animal cells must also synthesize a variety of proteins and other organic molecules necessary for growth and repair. Ribosomes produce these proteins. The cell may use these proteins or move them out of the cell for use in other cells. To move organic molecules, the cell contains a complex system of membranes that create channels within the cell. This system of membranes is called the endoplasmic reticulum.

B Target: Factory

Security guards monitor the doors of the factory. They control the movement of people into and out of the factory. Factory activities may be coordinated by a control center. These activities require energy. The electrical generators in the factory provide energy. The factory synthesizes products from raw materials using machines. The factory has hallways to move products through it.

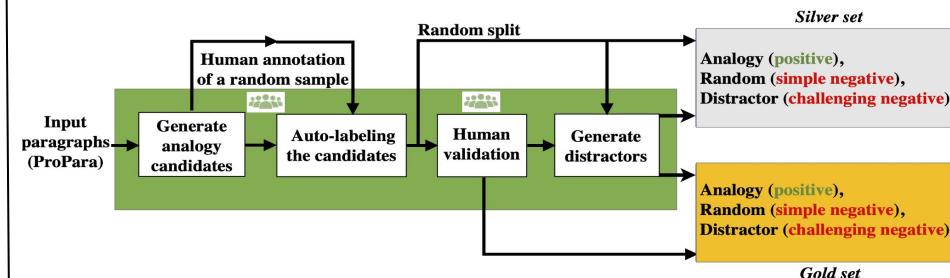
- What controls something? (B)
- Who controls something? (T)
- What synthesizes something? (B, T)
- What uses something? (B, T)
- What moves something? (B, T)

C Base (B) Target (T)



ParallelPARC: A Scalable Pipeline for Generating Natural-Language Analogies

Oren Sultan, Yonatan Bitton, Ron Yosef, Dafna Shahaf



Base	Target	Similar Relations
<p>Title: How does a solar panel work?</p> <p>Domain: Engineering</p> <p>Paragraph: solar energy powers an electric current within a solar panel. The photovoltaic cells within the panel convert the energy from the sun into electricity. The electrical wires then spread this power throughout the panel. The electric current is then used to power whatever the panel is connected to.</p>	<p>Title: How does photosynthesis occur?</p> <p>Domain: Natural Science</p> <p>Paragraph: Photosynthesis occurs when sunlight powers chemical reactions within the chloroplasts of a plant. The chloroplasts are able to transform the energy from the sunlight into usable energy for the plant. This energy is then used to produce nutrients for the plant, which are then distributed throughout the plant.</p>	<p>(solar energy, powers, electric current) (sunlight, powers, chemical reactions)</p> <p>(photovoltaic cells, convert, energy) (chloroplasts, transform, energy)</p> <p>(electrical wires, spread, power) (plants, distribute, nutrients)</p>

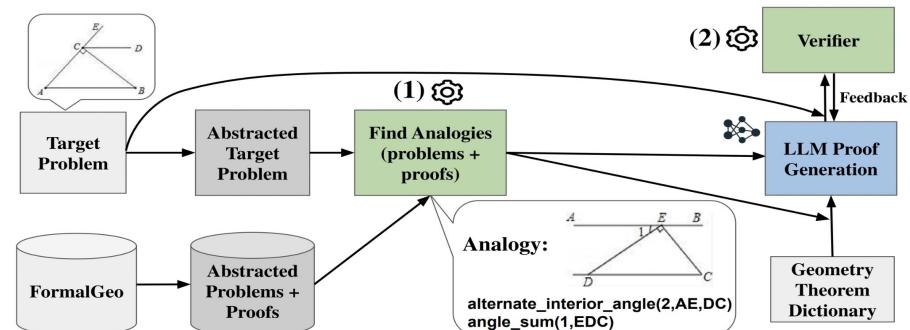
2) Combining LLMs with tools – A neuro-symbolic approach (2024-2025)



 ACL 2026
submission

Towards Reliable Proof Generation with LLMs: A Neuro-Symbolic Approach

Oren Sultan, Eitan Stern, Dafna Shahaf

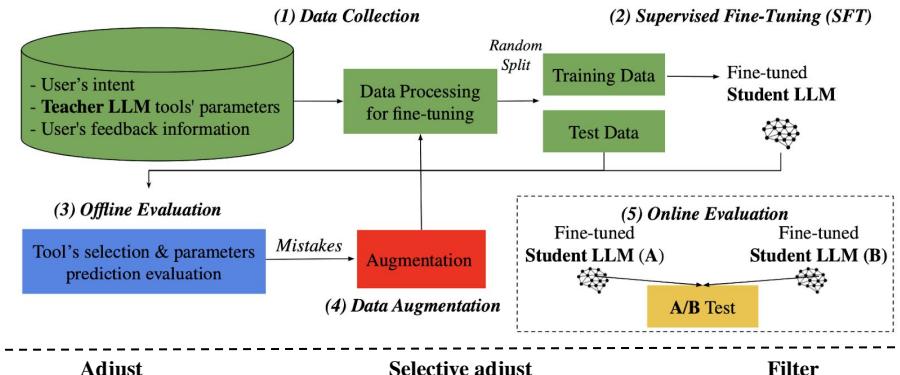


Problem:	Theorem Dictionary (GDL):
<p>Description: “$\angle ABC=40^\circ$, $CD \parallel AB$, BC is perpendicular to AC. Find the measure of $\angle ECD$”</p> <p>Construction: $Shape(CA,AB,BC)$, $Shape(EC,CD) \dots$</p> <p>Construction (extended): $Shape(AB,BC,CA)$, $Shape(BC,CA,AB) \dots$</p> <p>Conditions: $Equal(MeasureOfAngle(ABC),40)$, $ParallelBetweenLine(CD,AB) \dots$</p> <p>Goal: $Value(MeasureOfAngle(ECD))$</p> <p>Answer: 50</p> <p>Proof:</p> <p>1) $parallel_property_alternate_interior_angle(I,CD,AB)$ 2) $angle_addition(I,ECD,DCB)$ 3) $adjacent_complementary_angle(I,ECB,BCA)$</p>	<p>parallel_property_alternate_interior_angle(AB,CD):</p> <p>{</p> <p>1: {</p> <p> premises:</p> <p>$ParallelBetweenLine(AB,CD) \& Line(AD)$,</p> <p> conclusions: $Equal(MeasureOfAngle(BAD), \dots)$</p> <p>},</p> <p>2: { ... },</p> <p>...</p> <p>},</p> <p>angle_addition(ABC, CBD): {</p> <p>1: { ... },</p> <p>},</p> <p>...</p>

2) Combining LLMs with tools – A neuro-symbolic approach (2024)

Visual Editing with LLM-based Tool Chaining: An Efficient Distillation Approach for Real-Time Applications

Oren Sultan, Alex Khasin, Guy Shiran, Asnat Greenstein-Messica, Dafna Shahaf



Adjust

```
{  
    "exposure": 0,  
    "contrast": 10,  
    "brightness": 10,  
    "highlights": 20,  
    "shadows": -10,  
    "saturation": 15,  
    "vibrance": 15,  
    "temperature": 30,  
    "tint": 10,  
    "hue": 0,  
    "bloom": 0,  
    "sharpen": 0,  
    "structure": 0,  
    "linearOffset": 0  
}
```

Selective adjust

```
{  
    "red": {"saturation": 20, "luminance": 10},  
    "orange": {"saturation": 30, "luminance": 20},  
    "yellow": {"saturation": 40, "luminance": 30},  
    "green": {"saturation": -20, "luminance": 0},  
    "cyan": {"saturation": -20, "luminance": 0},  
    "blue": {"saturation": 0, "luminance": 0}  
}
```

Filter

```
{  
    "name": "faded_HighNoon",  
    "intensity": 40  
}
```

“Golden hour”



3) Code World Models (CWM) and the Halting Problem (2025)



CWM: An Open-Weights LLM for Research on Code Generation with World Models

Meta FAIR CodeGen Team, Oren Sultan

Meta FAIR CodeGen Team

```
<|trace_context_start|>  
  
def count(s, t):  
    n = 0  
    for c in s:  
        n += int(c == t)  
    return n  
  
count("strawberry", "r") # << START_OF_TRACE  
  
<|frame_sep|> { "s": "'strawberry'", "t": "'r'" } <|action_sep|> def count(s, t):  
<|frame_sep|>  
<|line_sep|> { "s": "...", "t": "..." } <|action_sep|> n = 0  
<|frame_sep|>  
<|line_sep|> { "s": "...", "t": "...", "n": "0" } <|action_sep|> for c in s:  
<|frame_sep|>  
<|line_sep|> { "s": "...", "t": "...", "n": "...", "c": "'s'" } <|action_sep|> n += int(c == t)  
...  
<|frame_sep|>  
<|return_sep|> <|action_sep|> return n <|arg_sep|> "3"  
<|frame_sep|>
```



LLMs versus the Halting Problem: Revisiting Program Termination Prediction

Oren Sultan, Jordi Armengol-Estabé, Pascal Kesseli, Julien Vanegue, Dafna Shahaf, Peter O'Hearn, Yossi Adi

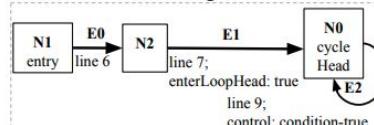
Meta FAIR CodeGen Team

C code example:

```
1: typedef enum {false,true} bool;  
2:  
3: extern int __VERIFIER_nondet_int(void);  
4:  
5: int main()  
6: {  
7:     int i;  
8:     i = __VERIFIER_nondet_int();  
9:     while (i >= -5 && i <= 5) {  
10:         if (i > 0) {  
11:             i = i-1;  
12:         }  
13:         if (i < 0) {  
14:             i = i+1;  
15:         }  
16:     }  
17:  
18:     return 0;  
19: }
```

Verdict: Non-termination

LLM witness prediction:



UAutomizer (Witness Validator)