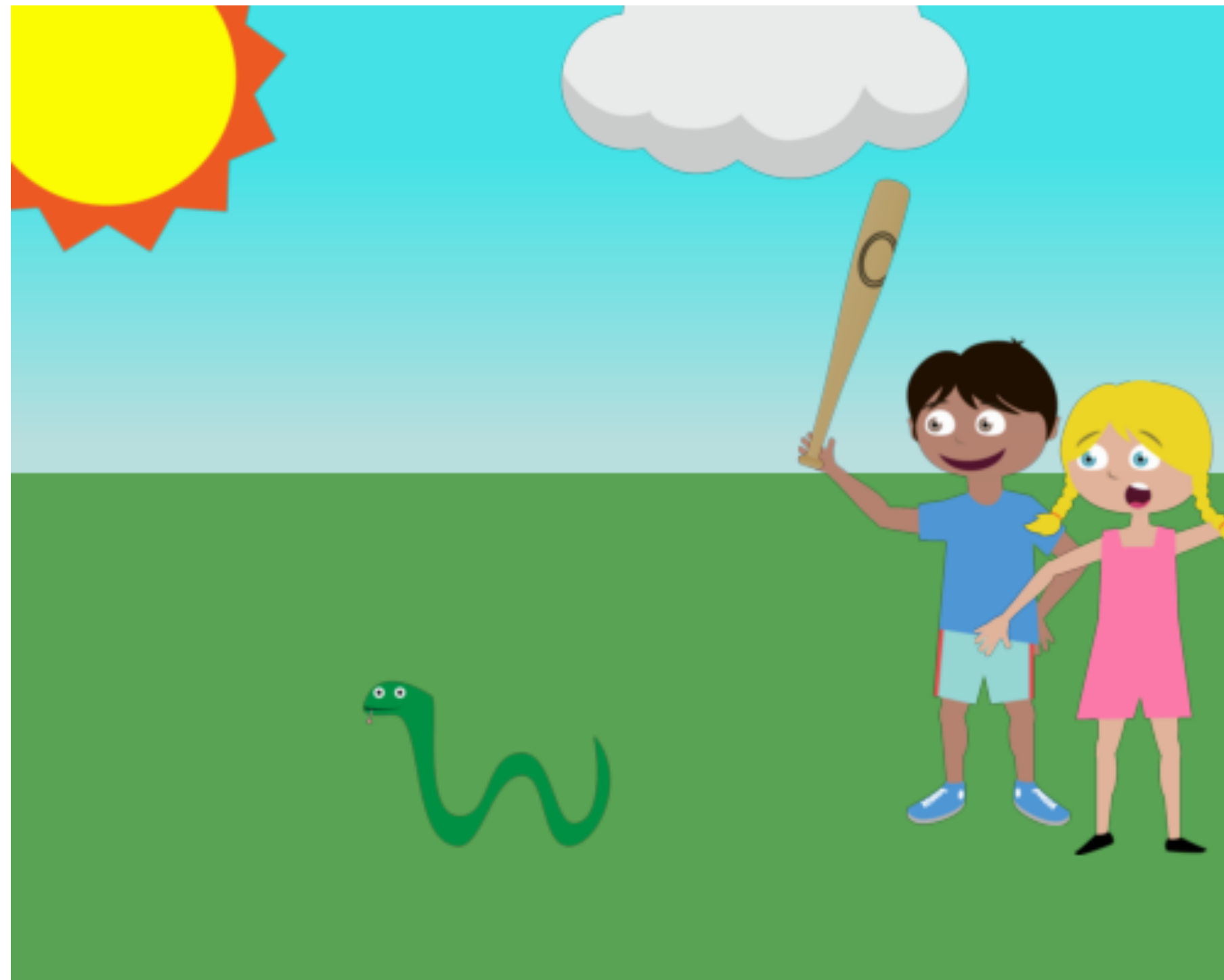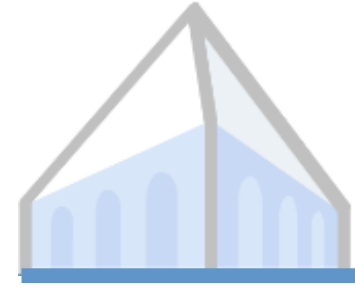# Reasoning about pragmatics
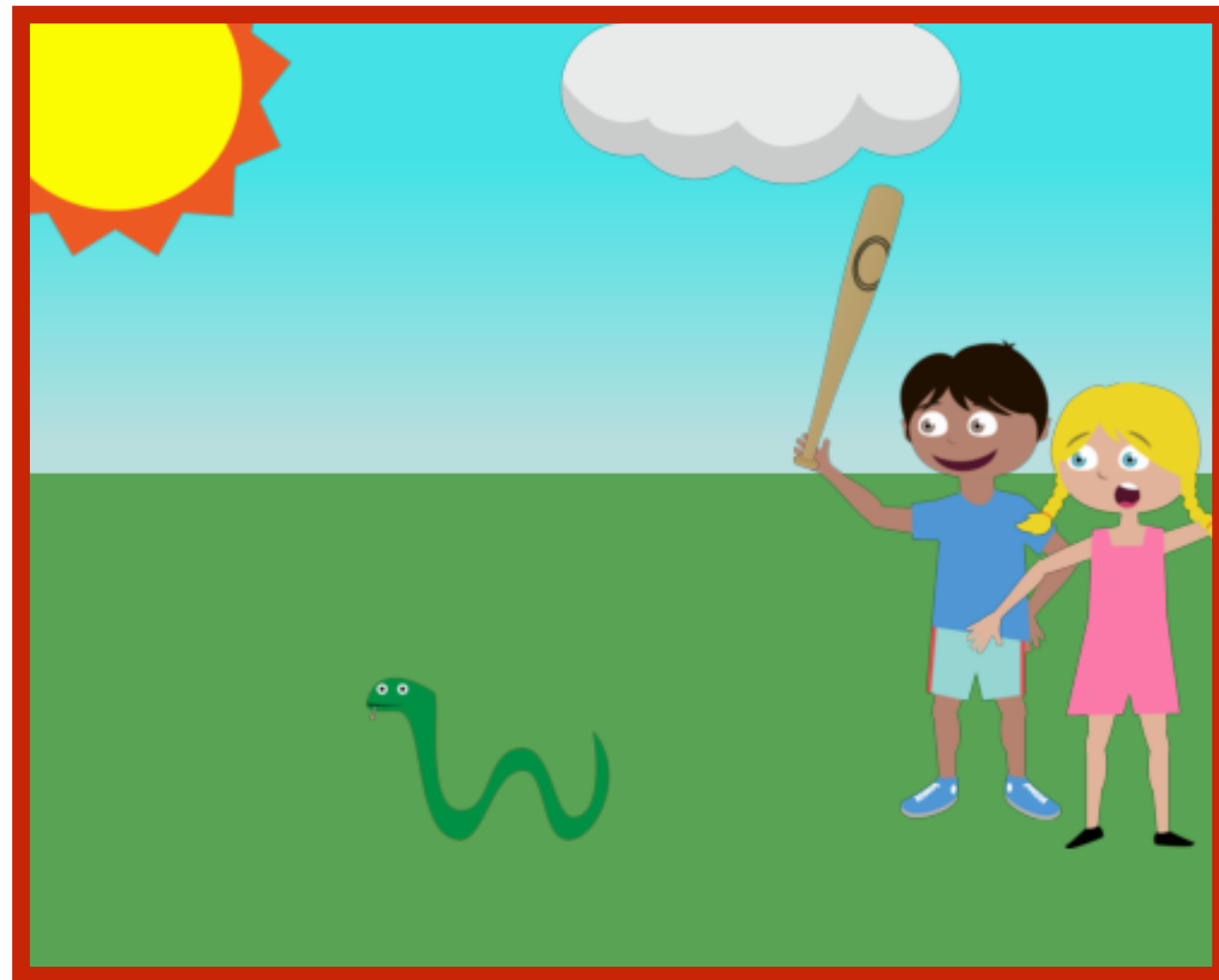# with neural listeners and speakers

Jacob Andreas and Dan Klein
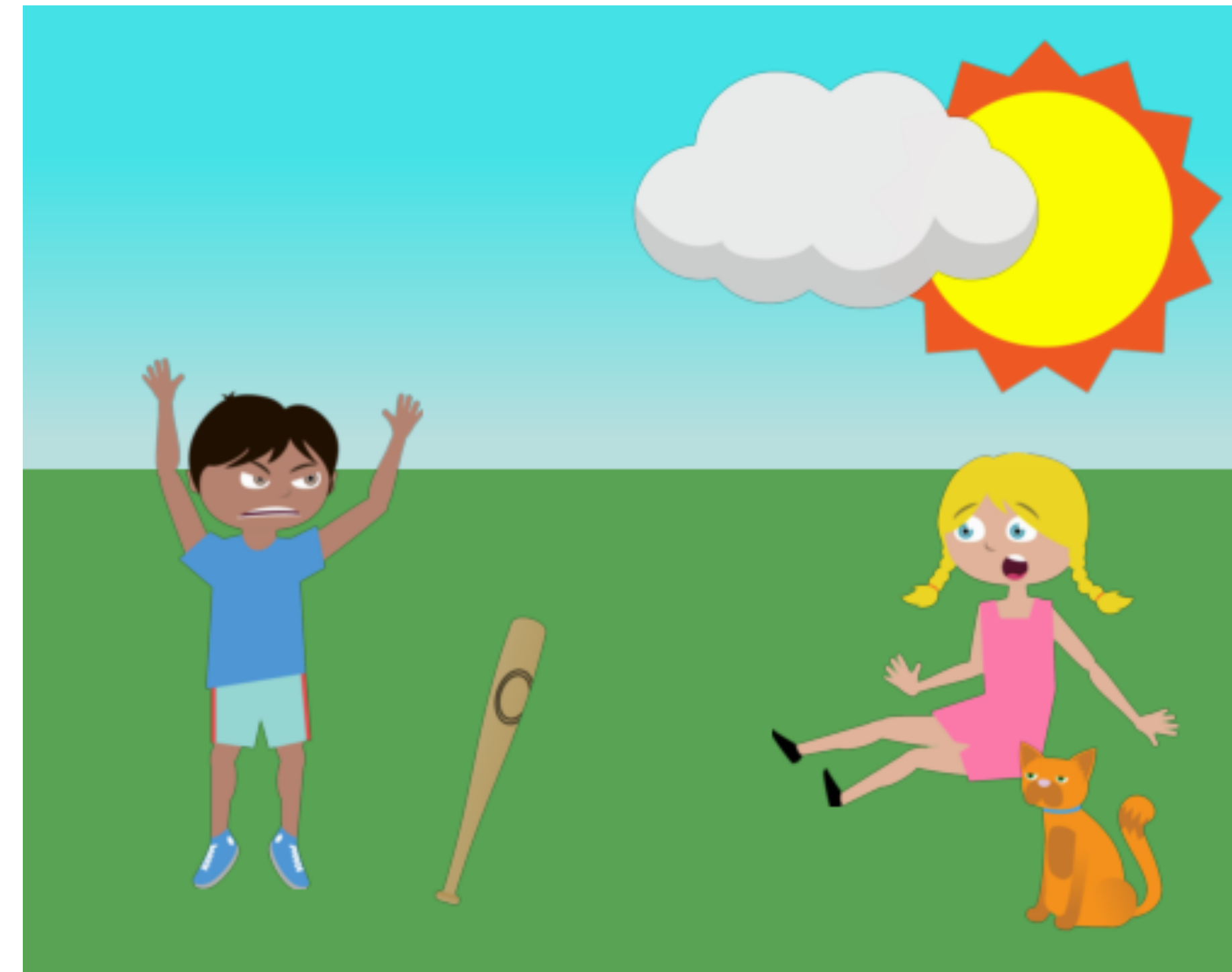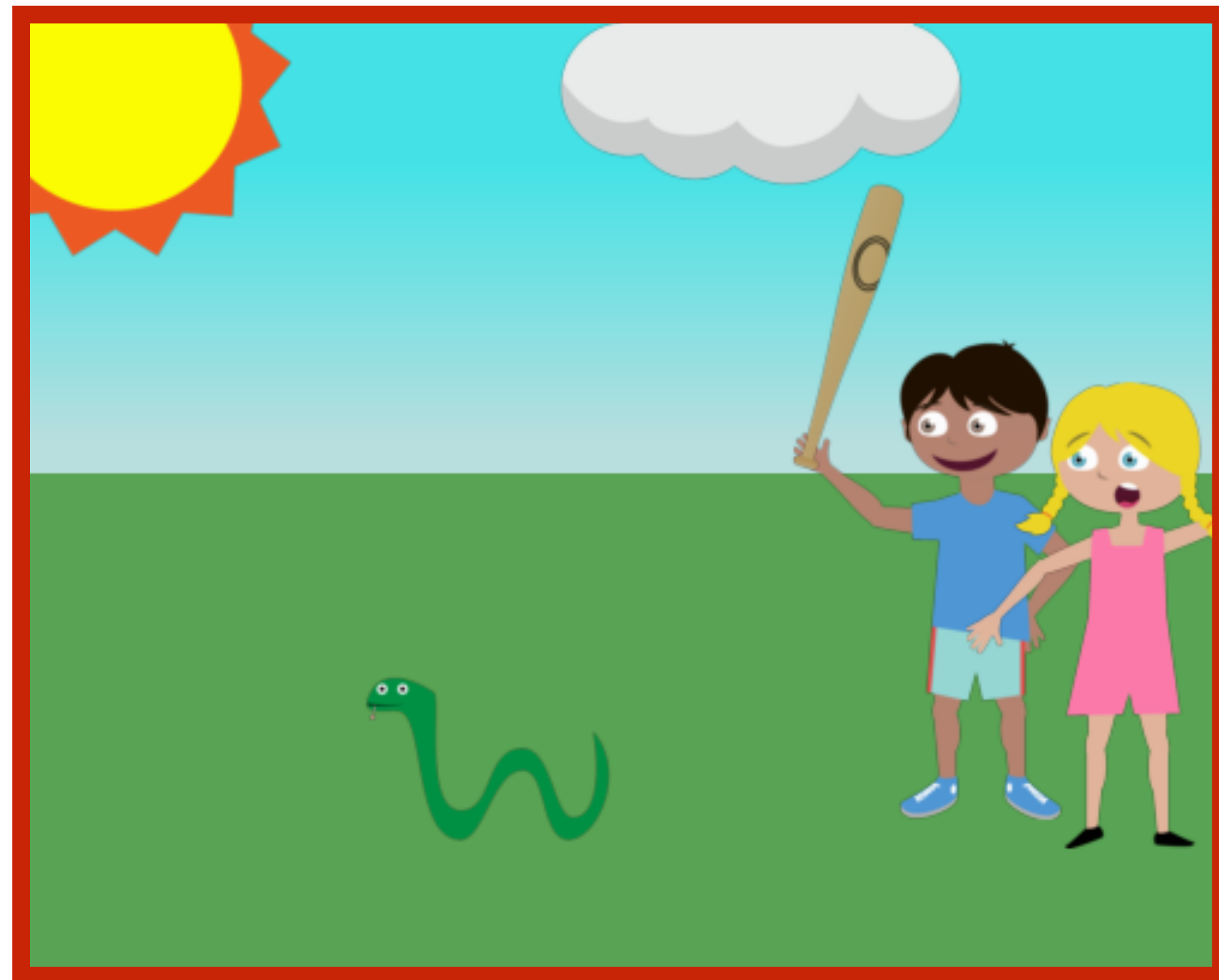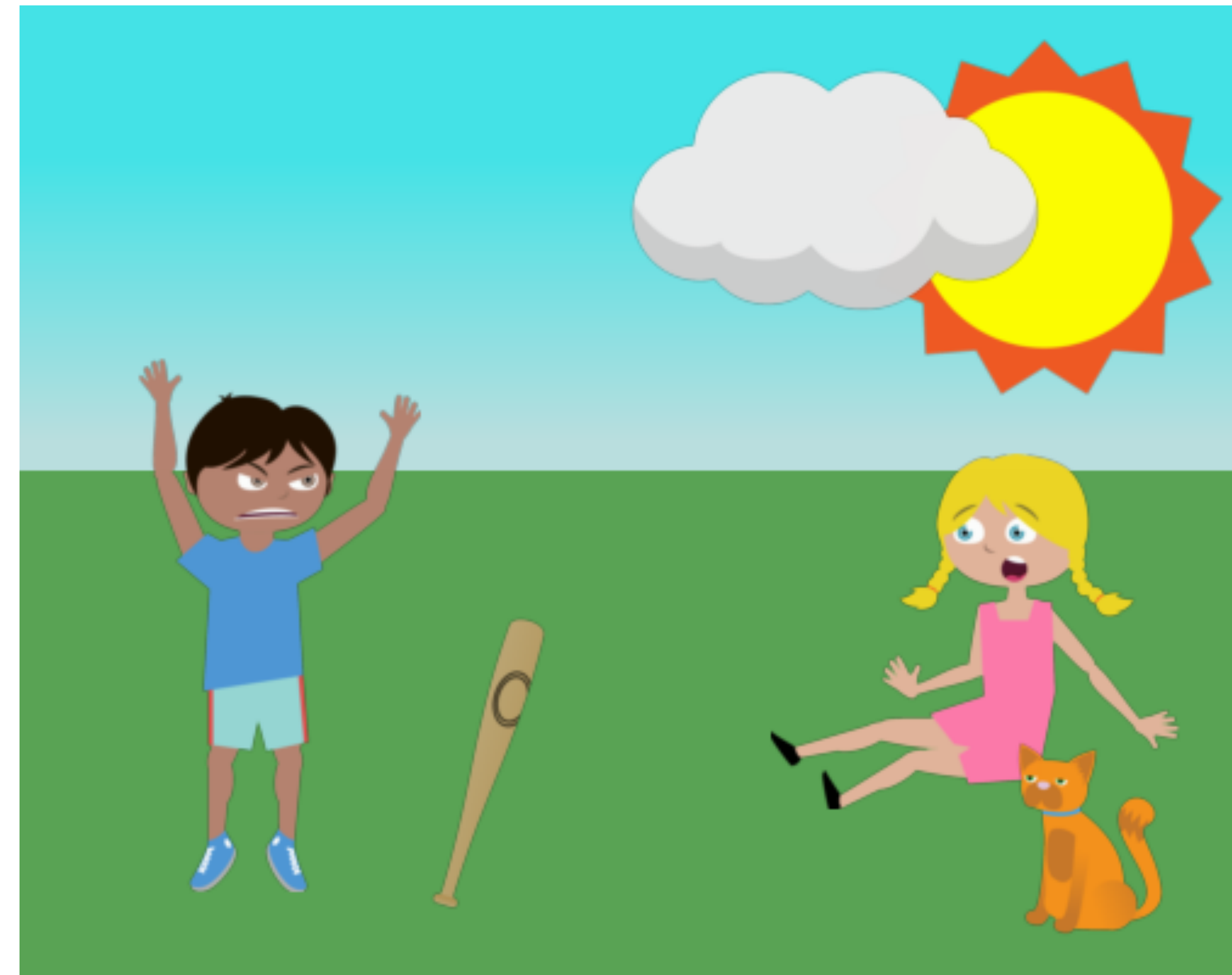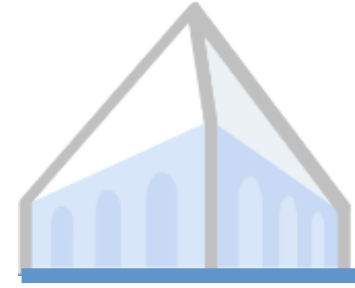
The reference game

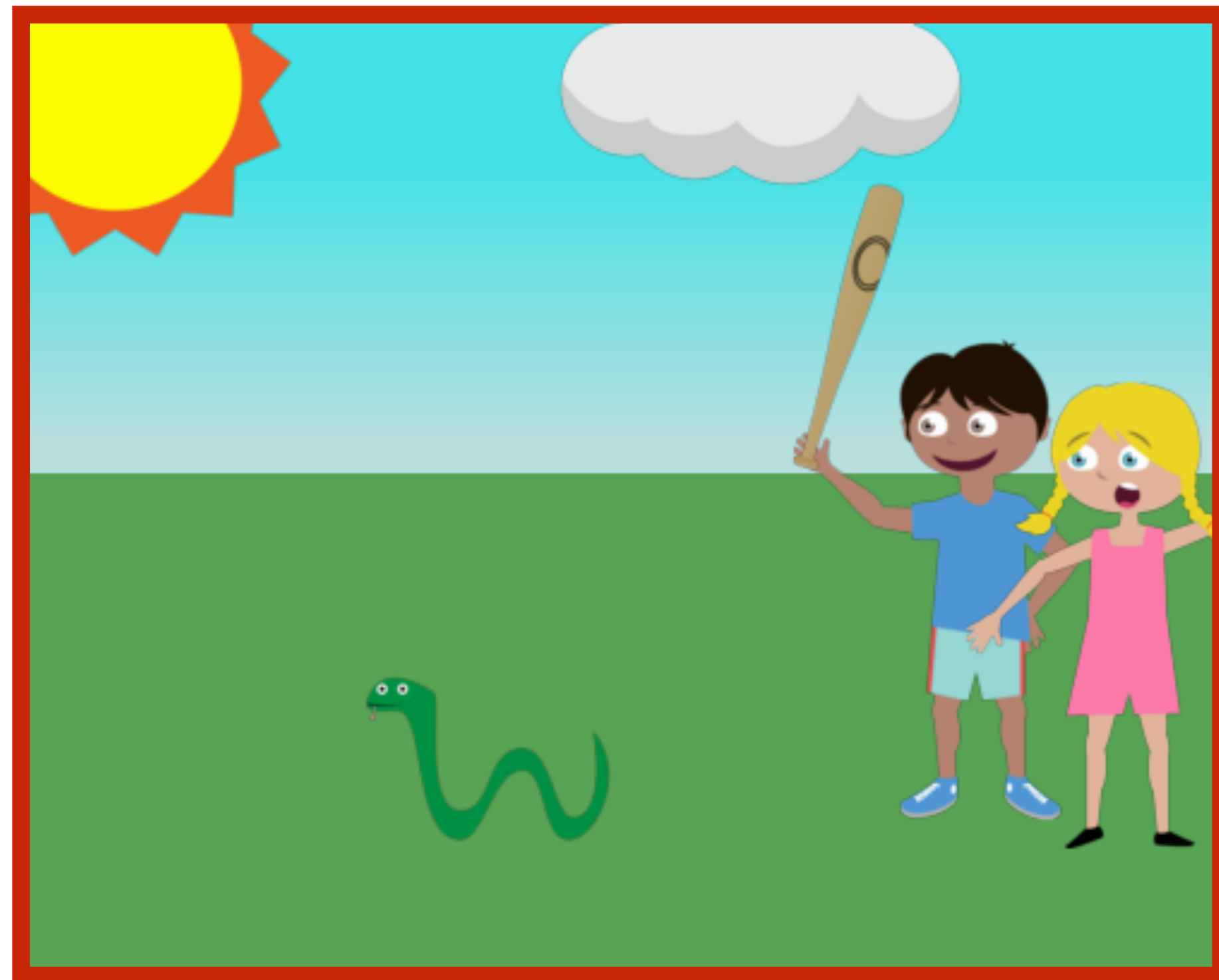# The reference game

The reference game
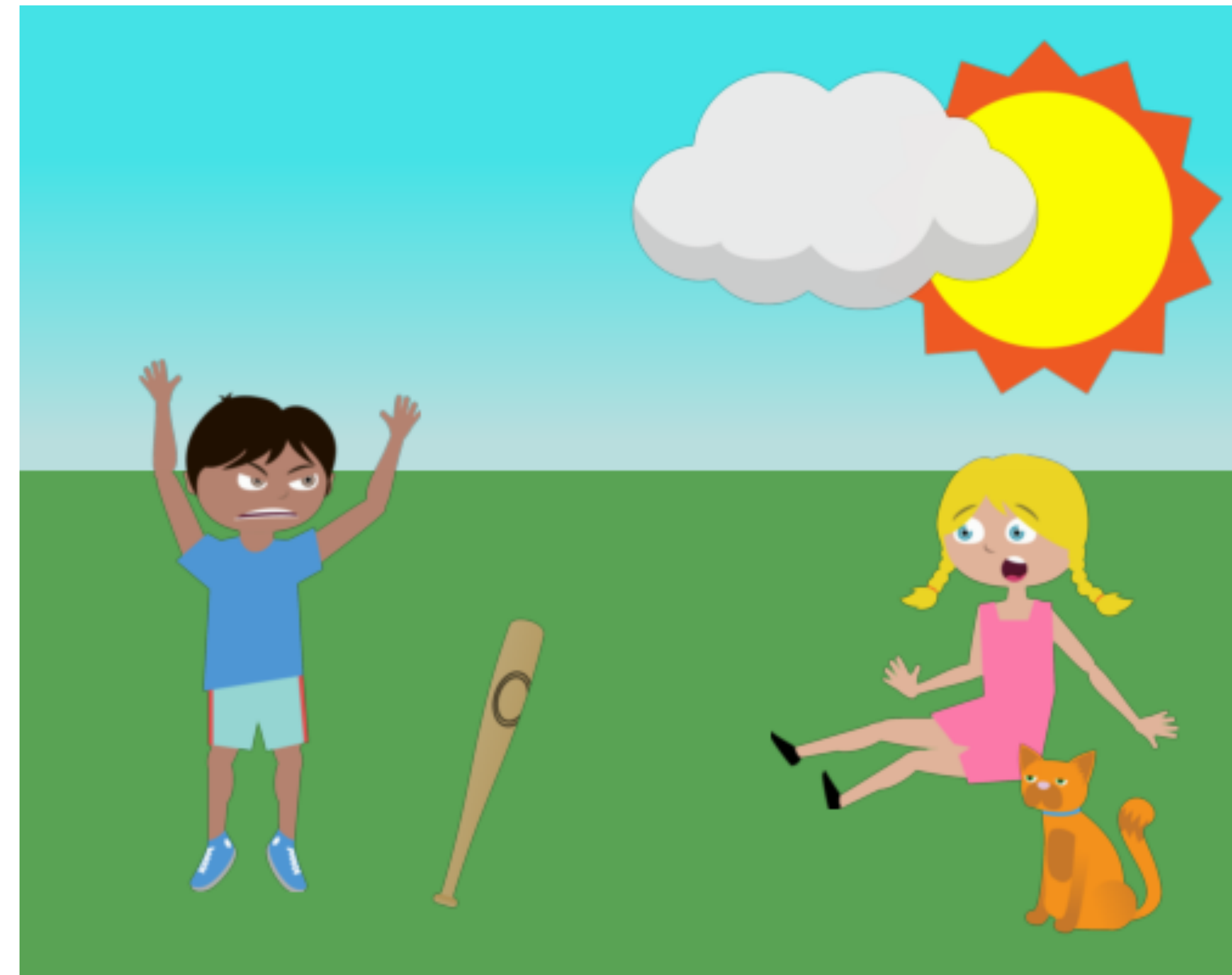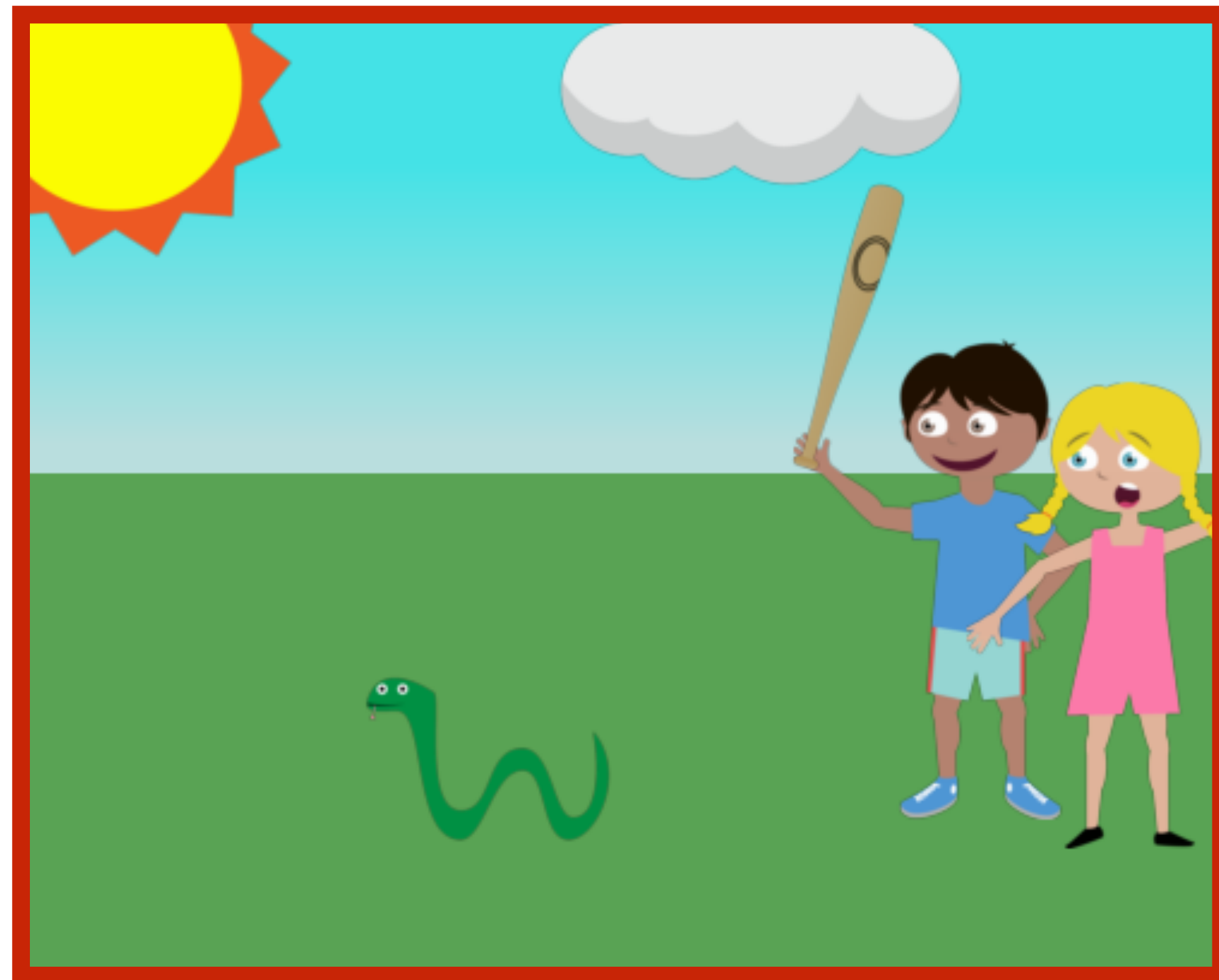
The one with the snake

The reference game

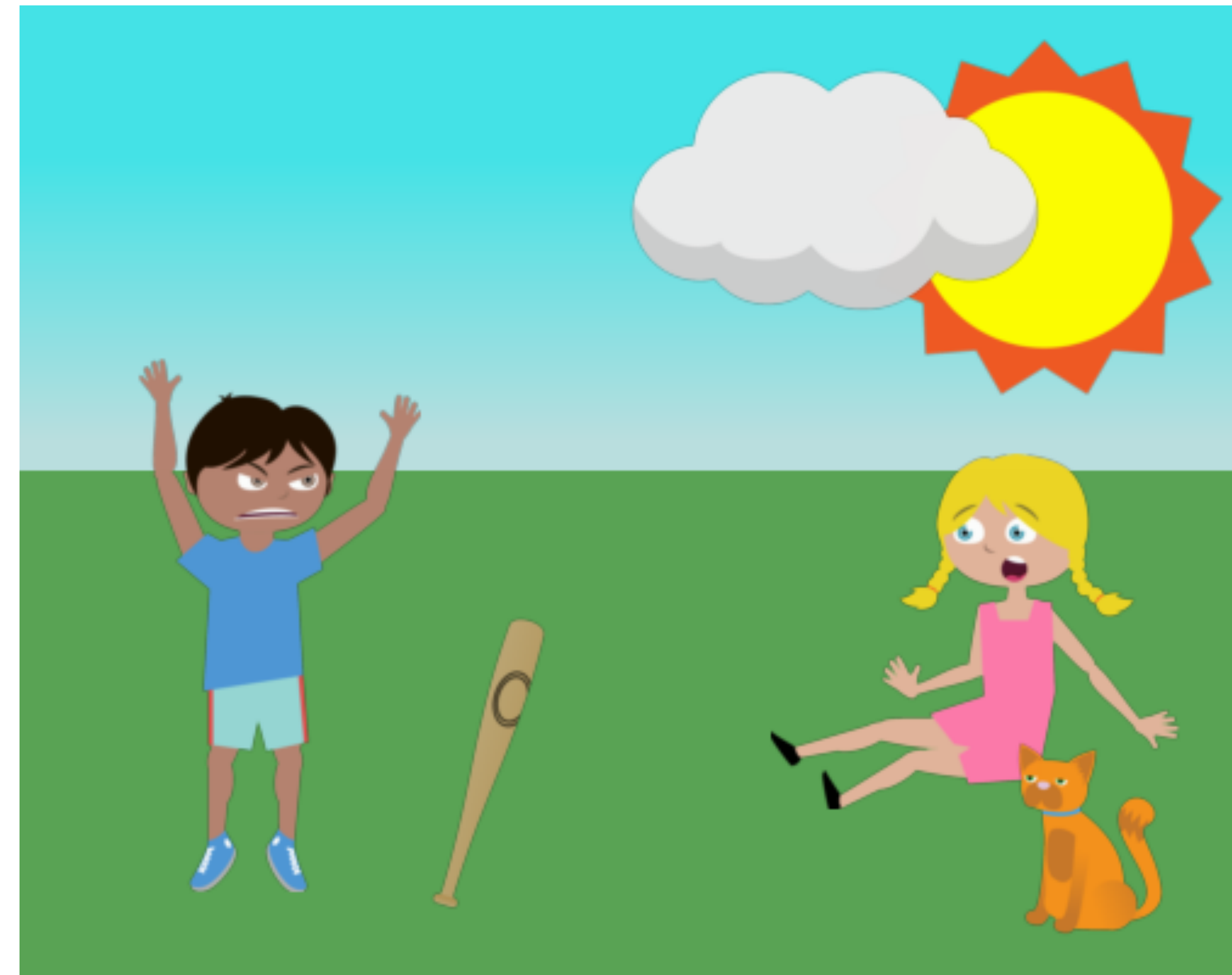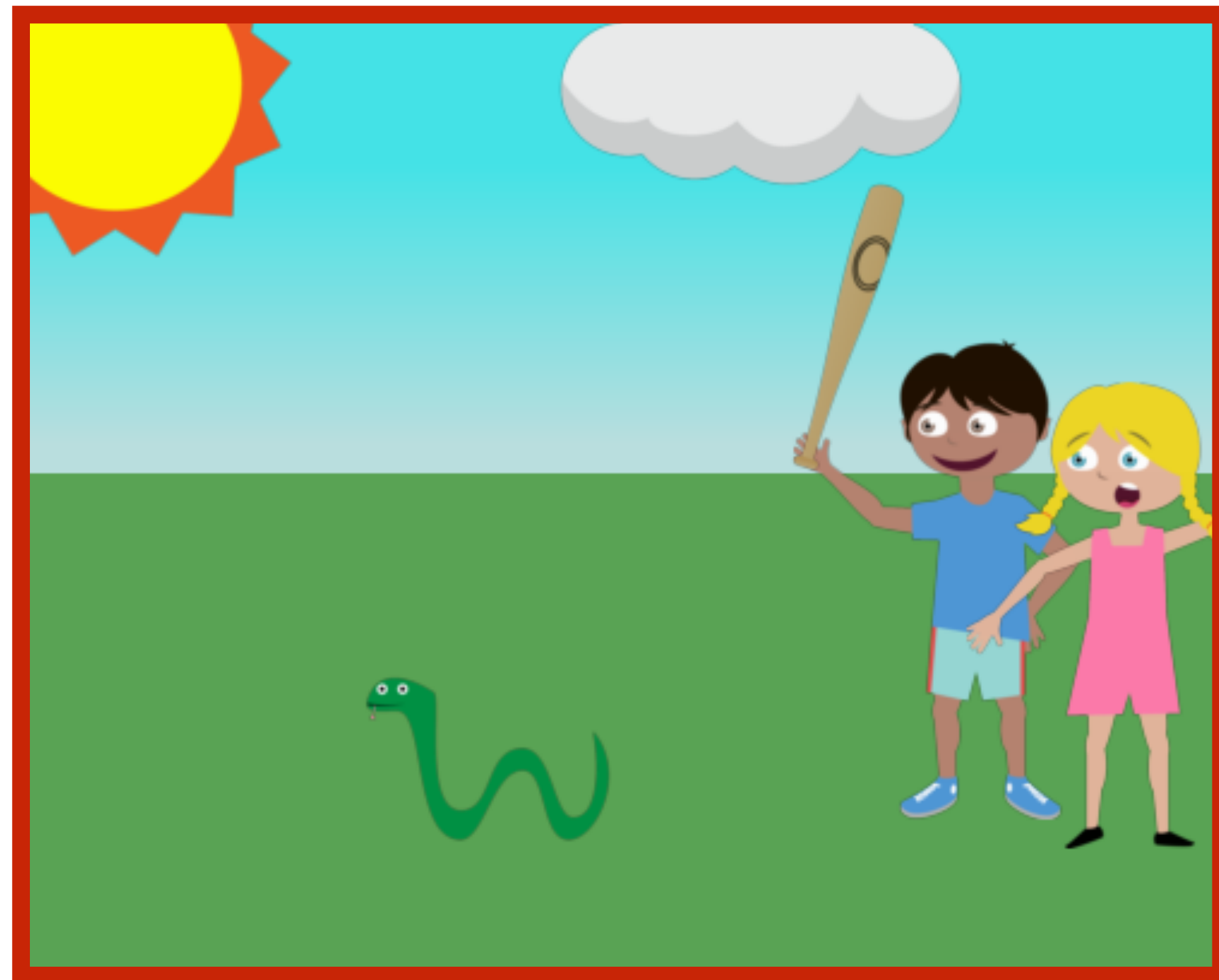*Mike is holding a baseball bat*

The reference game

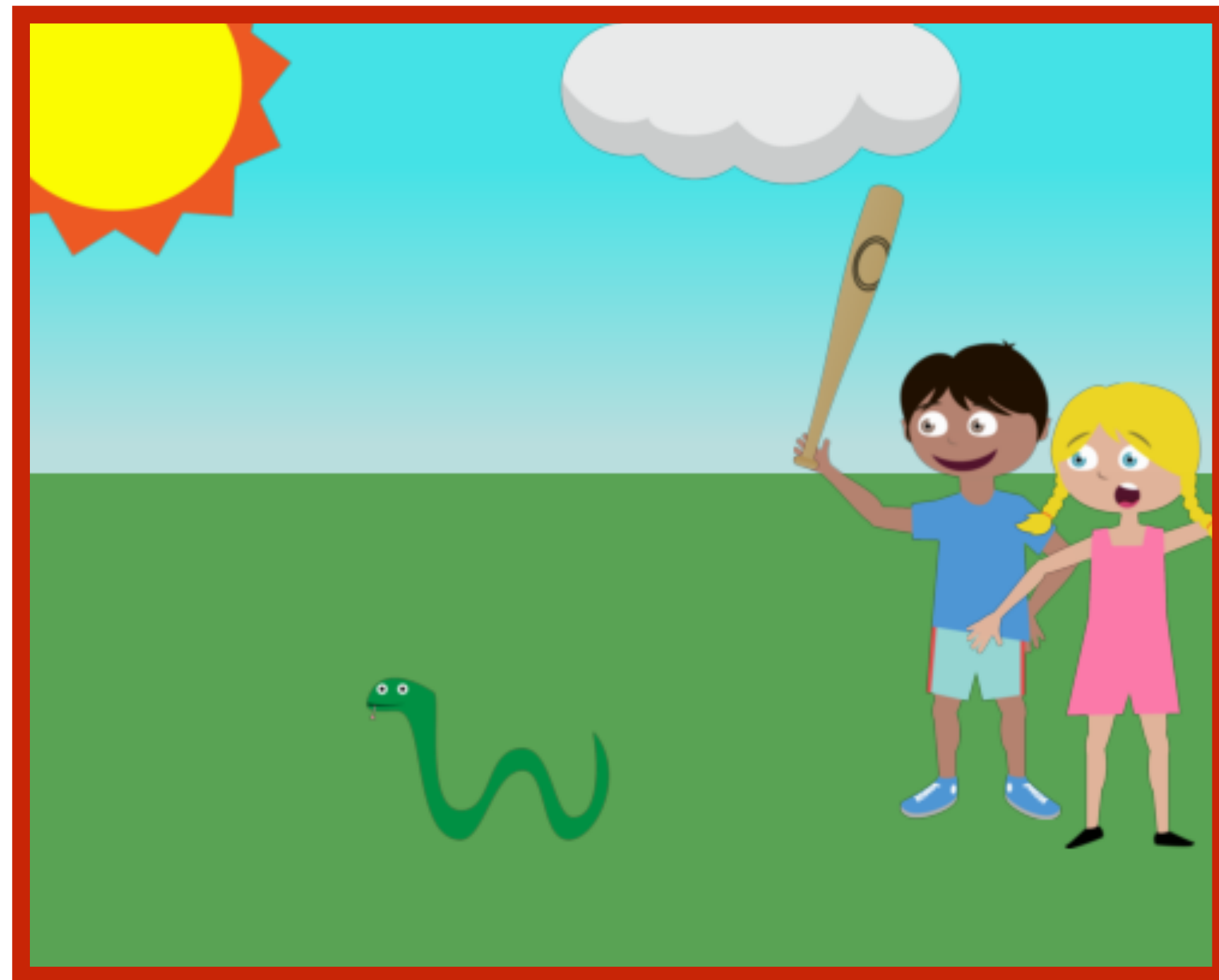*bat a is holding Mike baseball*

The reference game

They are sitting by a picnic table

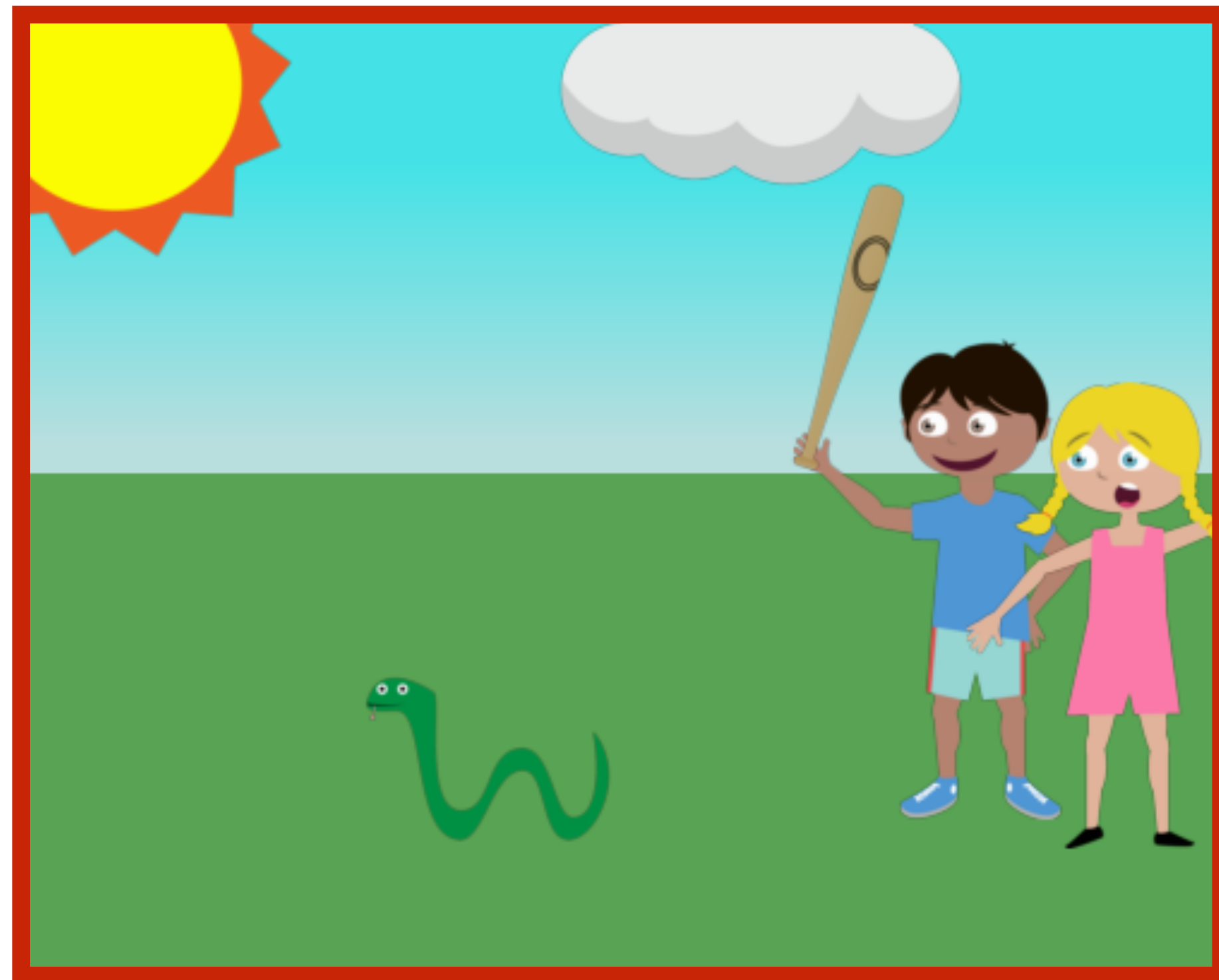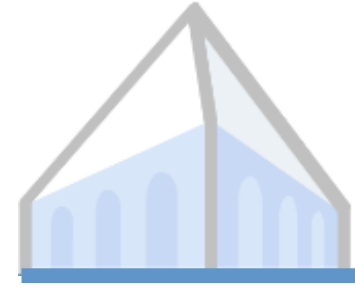# The reference game

There is a bat

The reference game
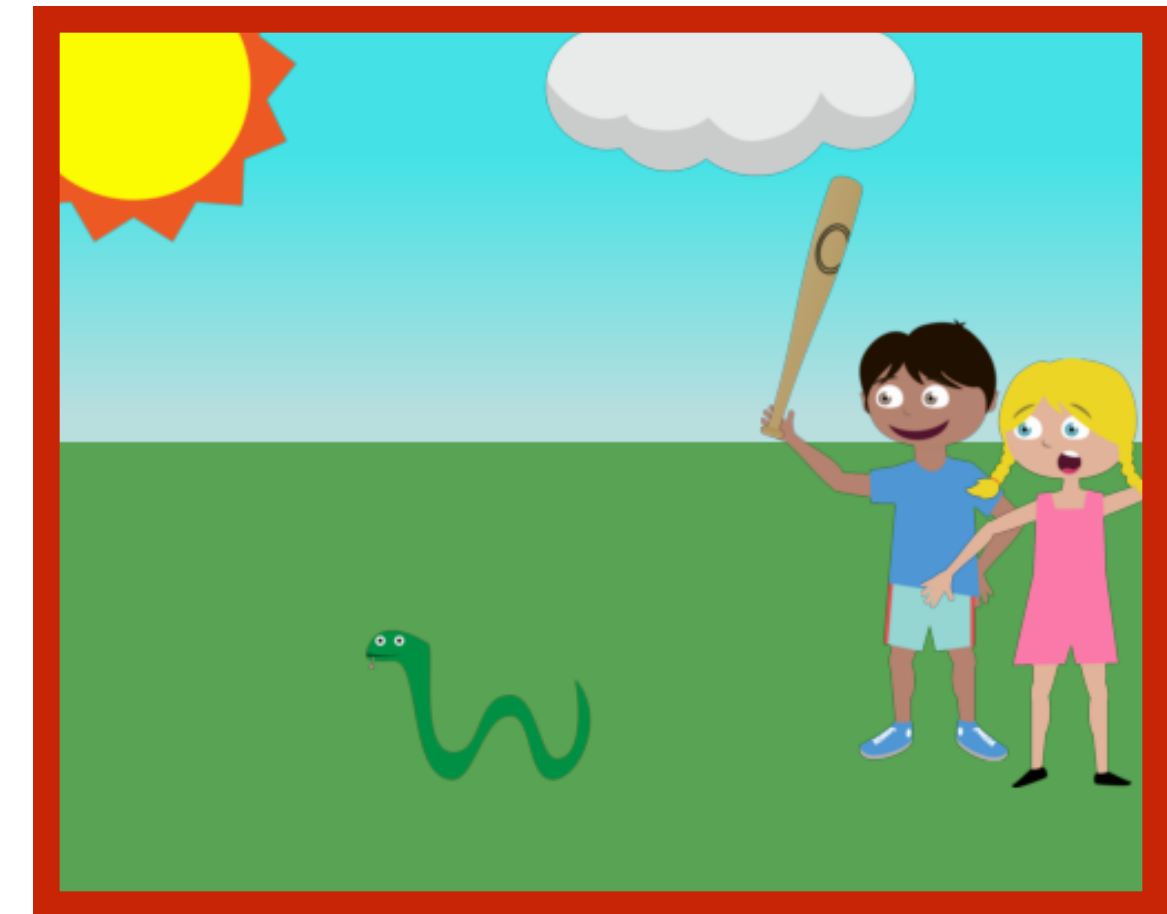
There is a bat

The reference game

Why do we care about this game?

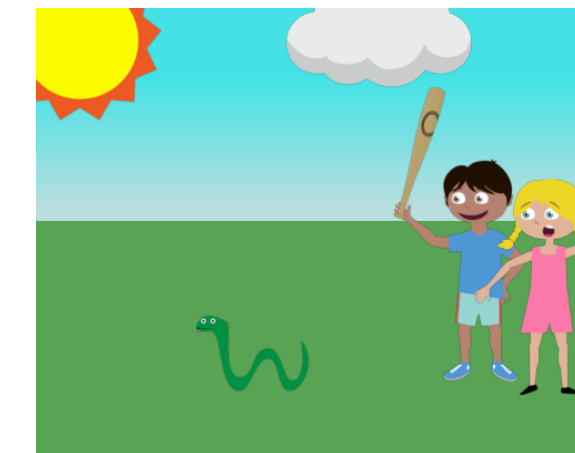*Don't you think it's a little cold in here?*
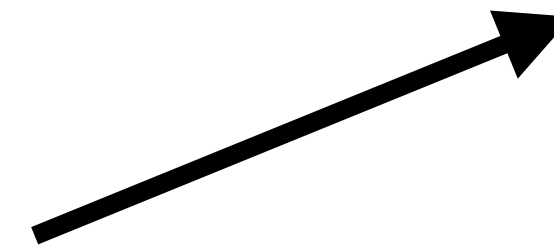
Do you know what time it is?

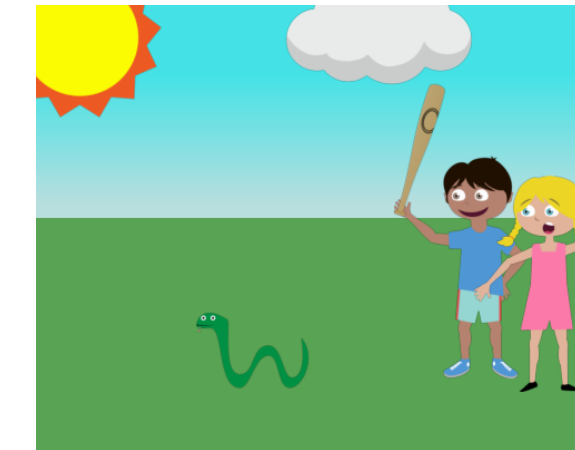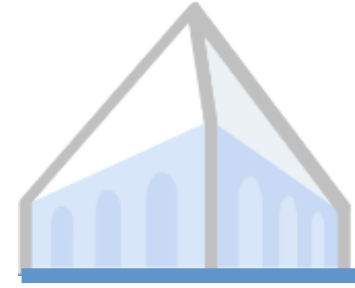*Some of the children played in the park.*
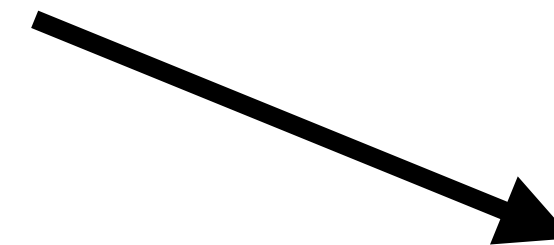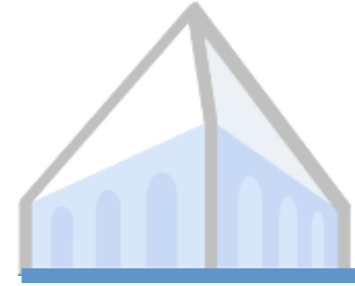
Deriving pragmatics from reasoning

Mike is holding a baseball bat
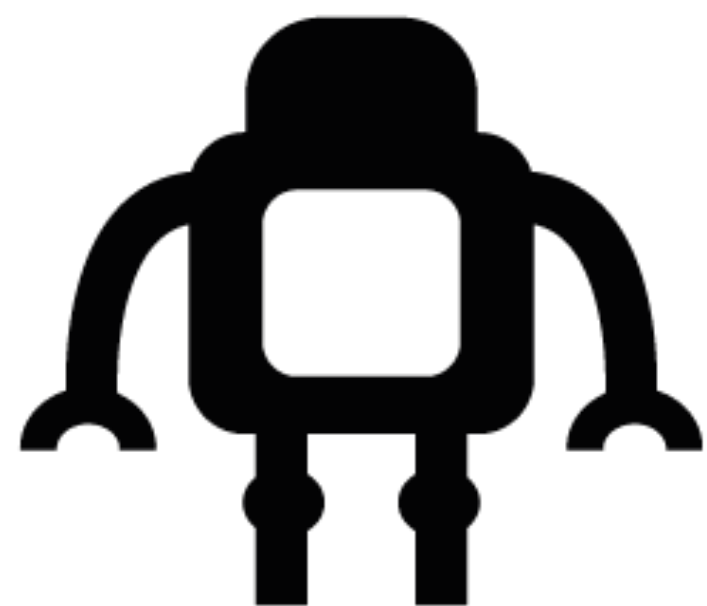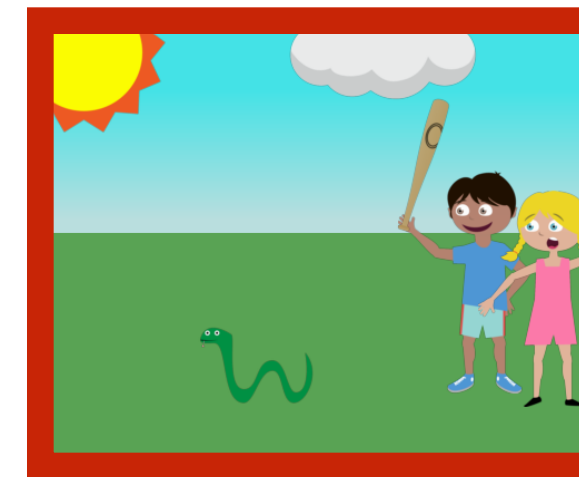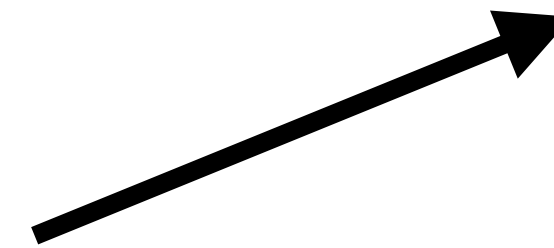
Deriving pragmatics from reasoning

Jenny is running from the snake →

Deriving pragmatics from reasoning
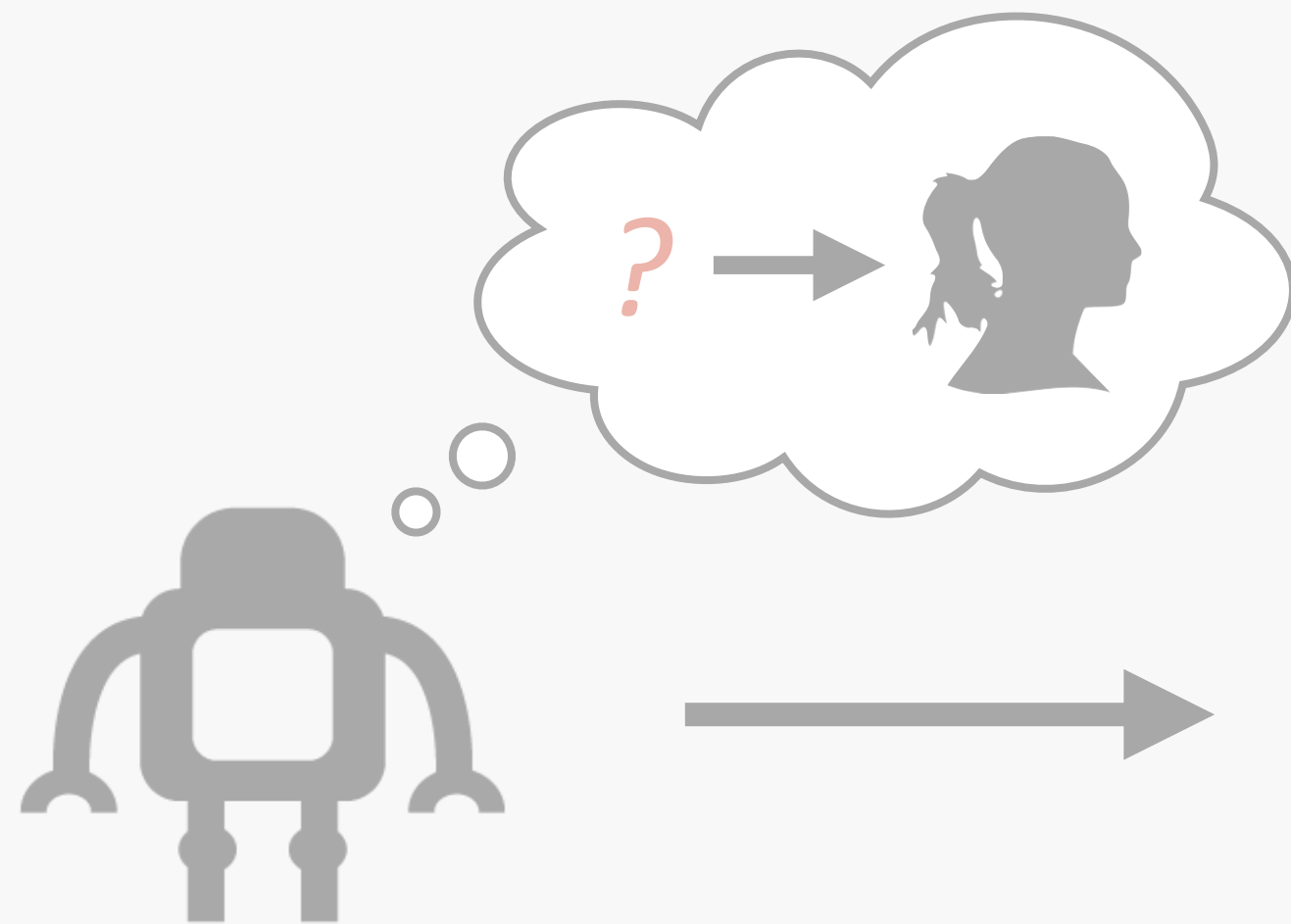
Mike is holding a baseball bat

How to win

DERIVED STRATEGY:
Reason about listener beliefs
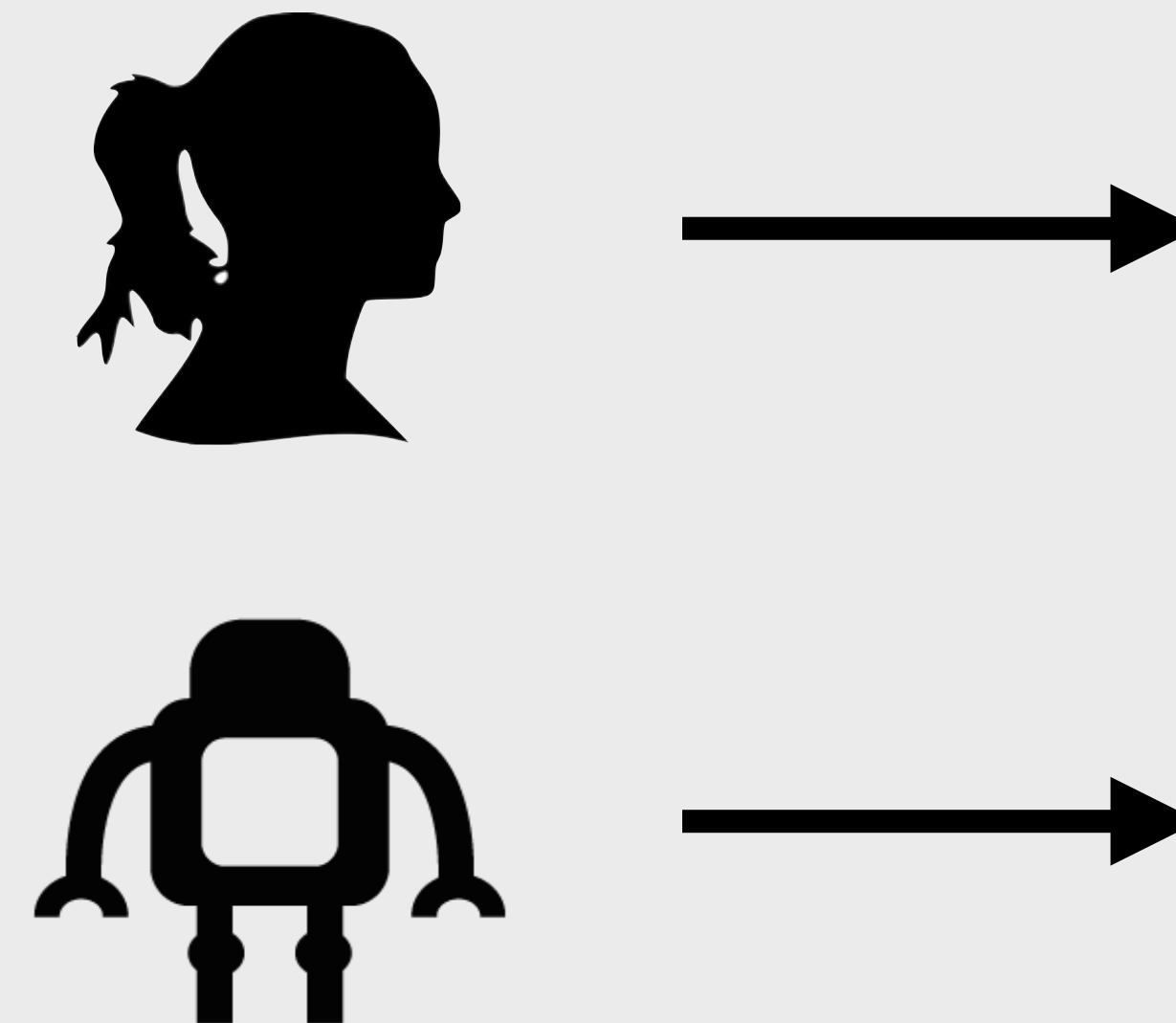
There is a snake

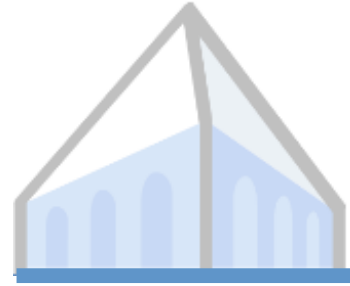**DIRECT STRATEGY**:
Imitate successful human play

There is a snake

There is a snake

How to win

DERIVED STRATEGY:
Reason about listener beliefs

[Monroe and Potts, 2015]

[Smith et al. 2013]

[Vogel et al. 2013]

[Golland et al. 2010]

DIRECT STRATEGY:
Imitate successful human play

[Mao et al. 2015]

[Kazemzadeh et al. 2014]

[Fitzgerald et al., 2013]

How to win

DERIVED STRATEGY:
Reason about listener beliefs

PRO: pragmatics "for free"

CON: past work needs
    hand-engineering

DIRECT STRATEGY:
Imitate successful human play

PRO: domain repr "for free"

CON: past work needs
    targeted data

How to win

DERIVED STRATEGY:
Reason about listener beliefs

Learn base models for interpretation & generation without pragmatic context

DIRECT STRATEGY:
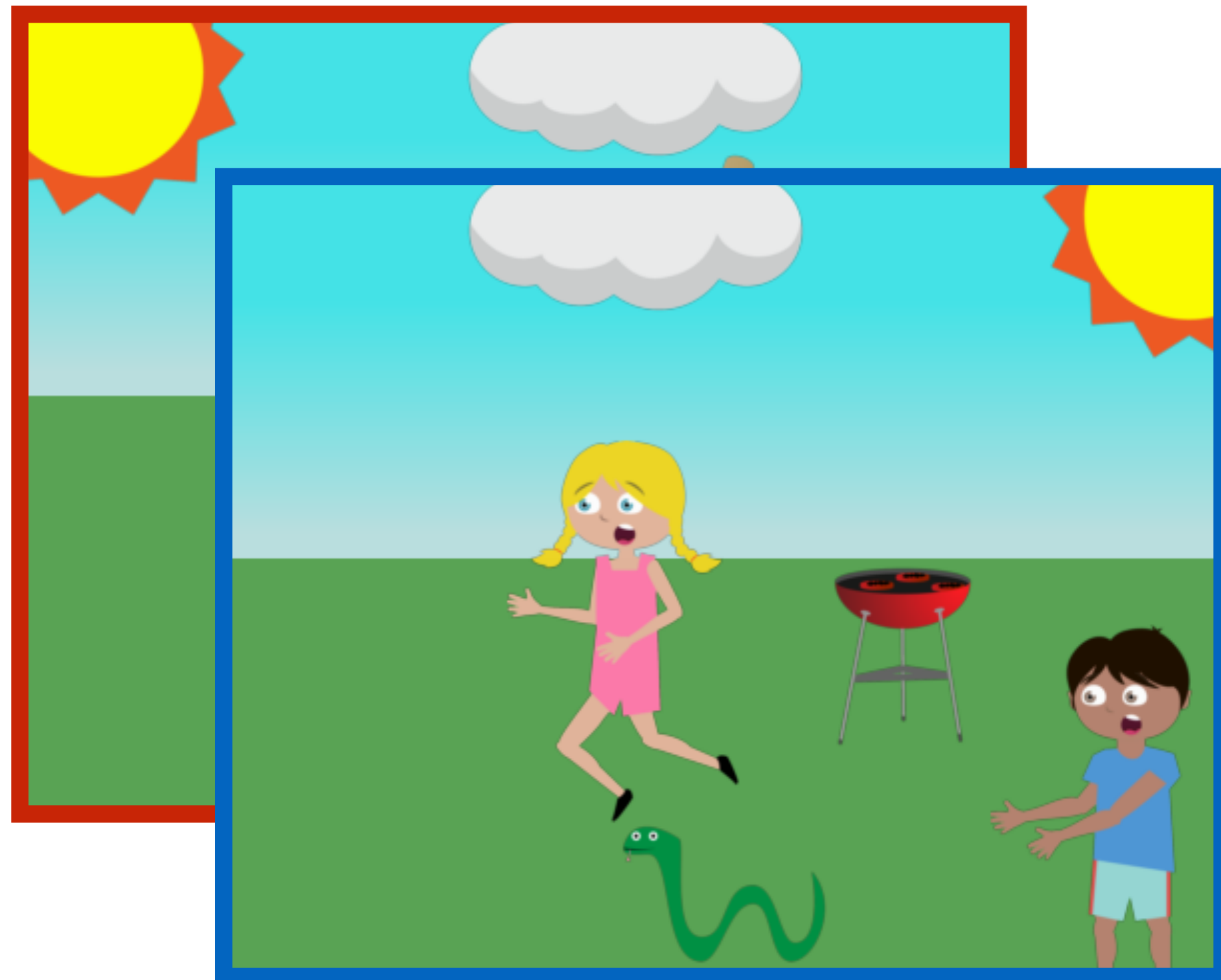Imitate successful human play

Explicitly **reason** about base models to get novel behavior
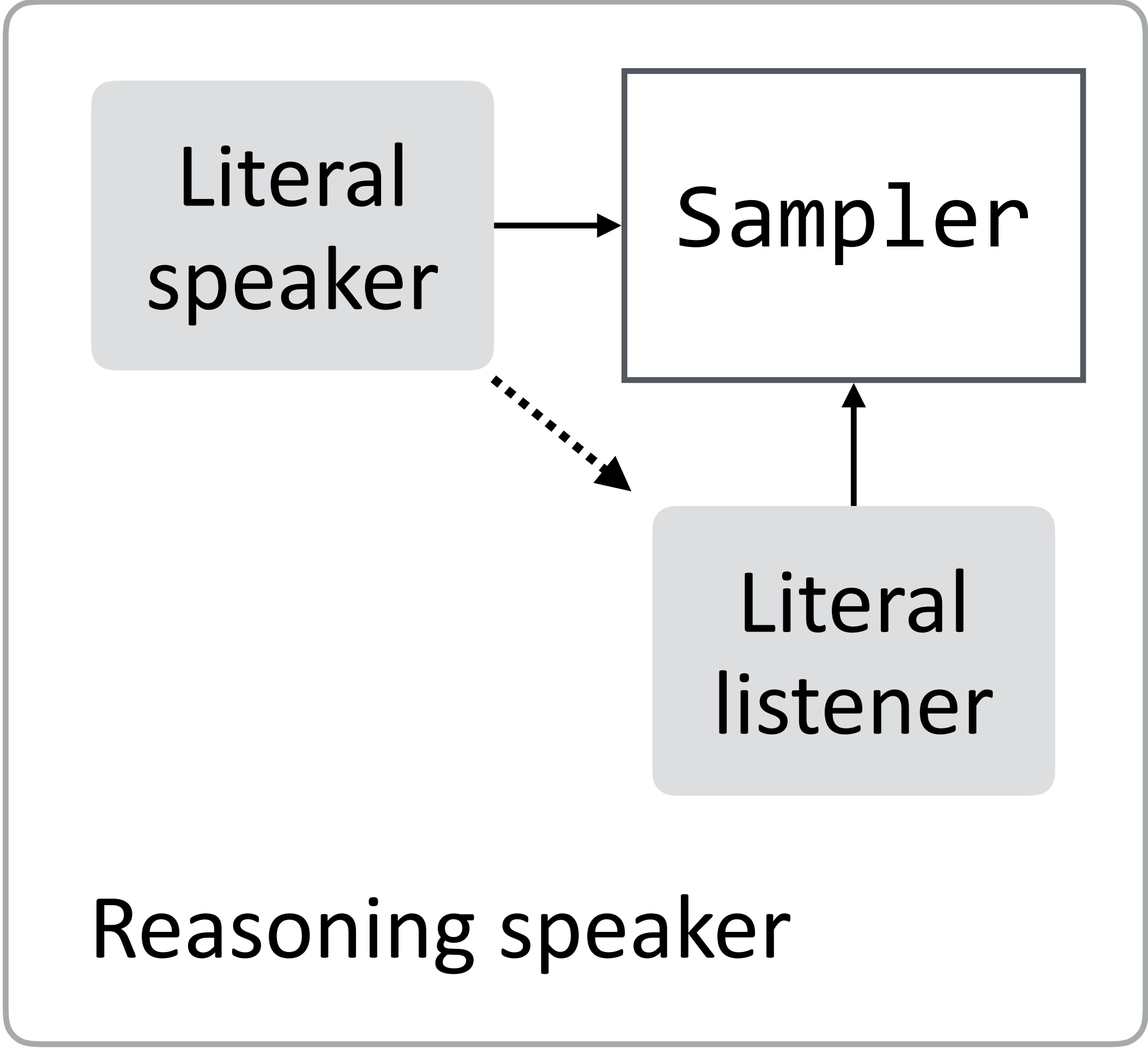
Data

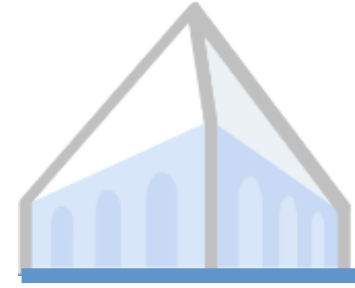**Abstract Scenes Dataset**
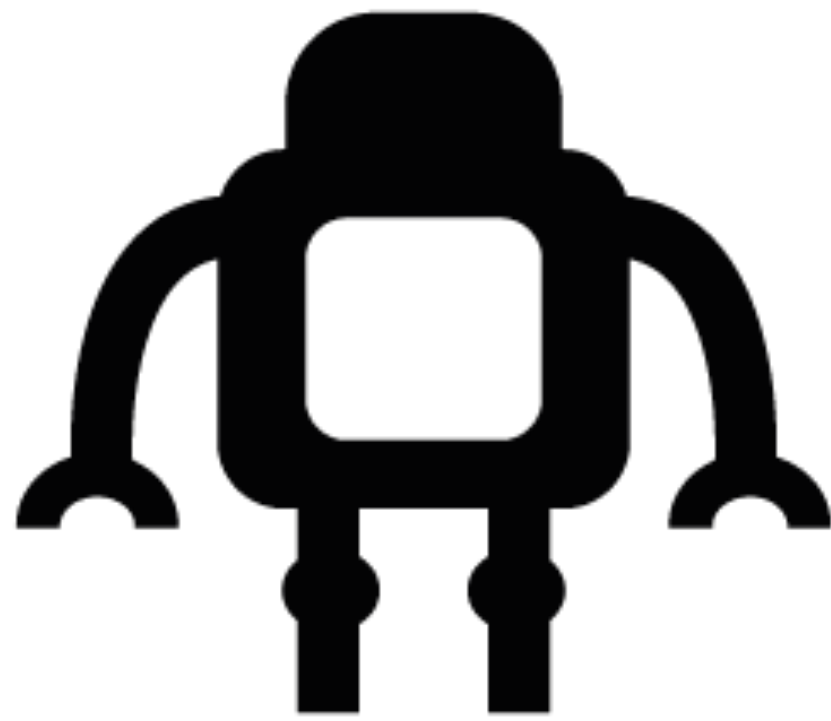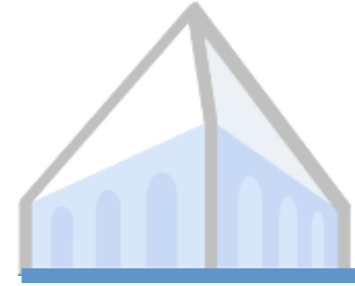
1000 scenes

10k sentences

Feature representations

Approach

A literal speaker (S0)
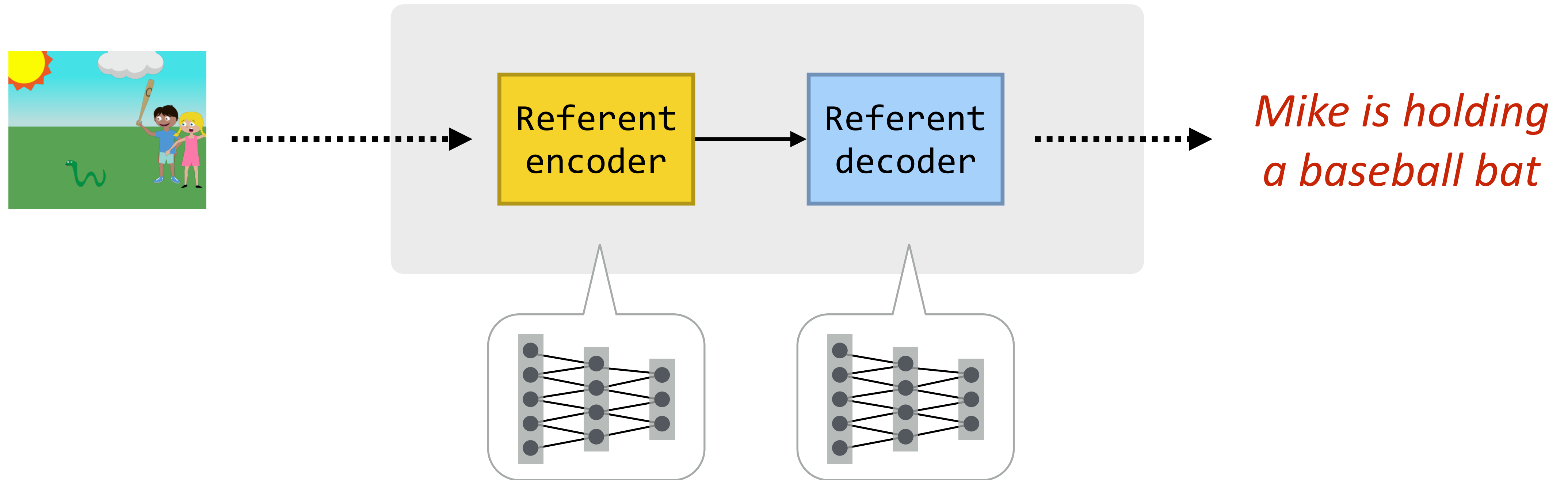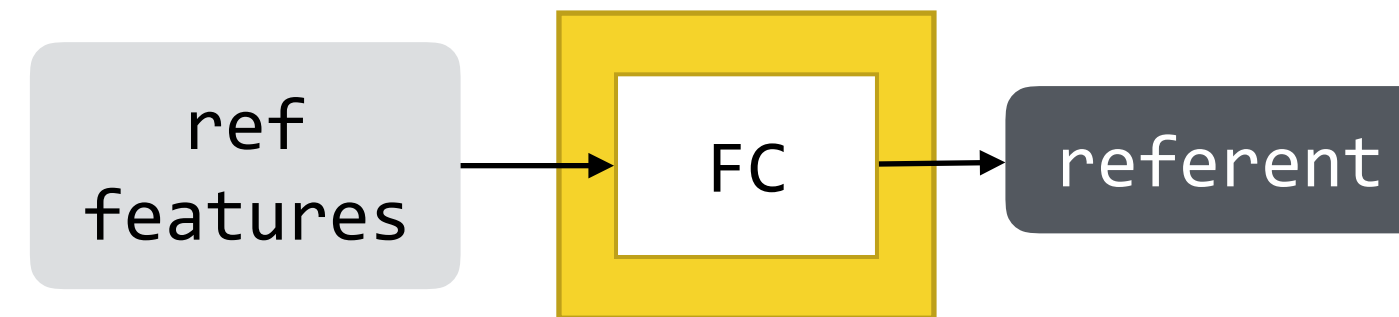

Mike is holding a baseball bat

A literal speaker (S0)

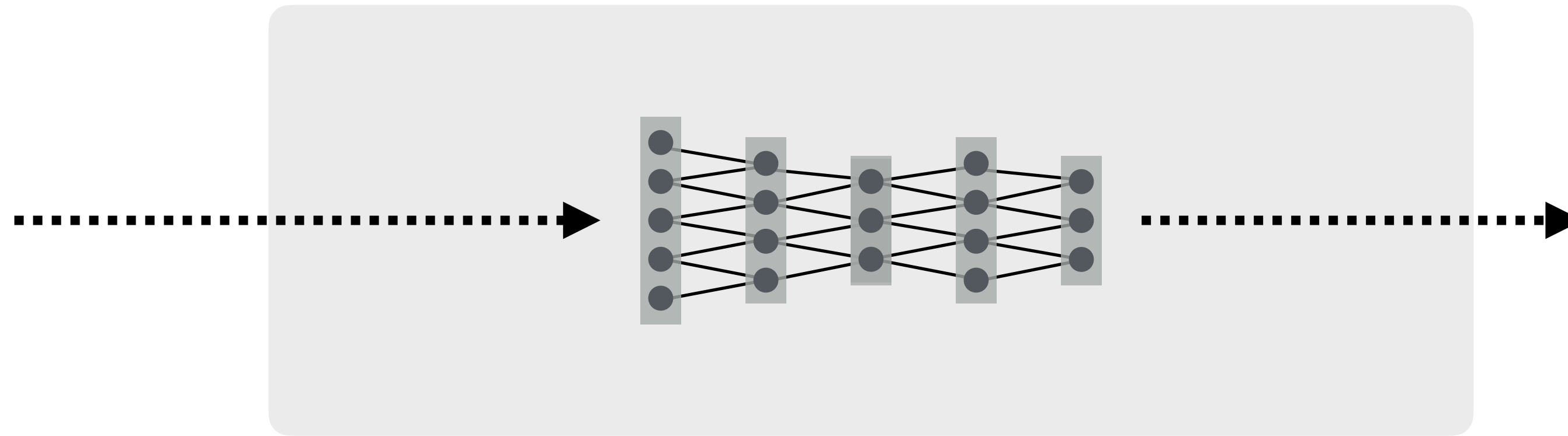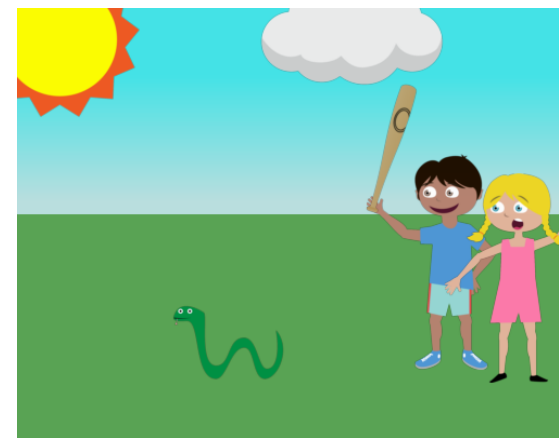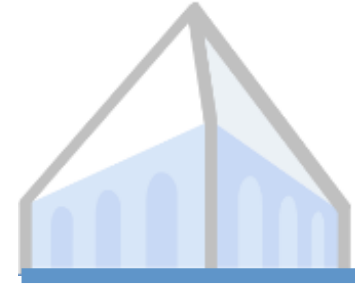# Module architectures

Referent encoder

```
ref        →  [ FC ]  →  referent
features
```

Referent decoder

```
word_n  ──┐
          ↓
word_<n ──→ [ FC → ReLU → FC → Softmax ] → word_{n+1}
          ↑
referent ─┘
```

Training S0

A literal speaker (S0)

S0

Mike is holding a baseball bat

The sun is in the sky

Jenny is standing next to Mike

Mike is holding a baseball bat
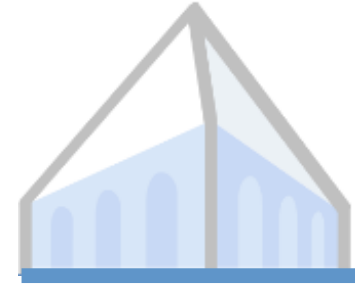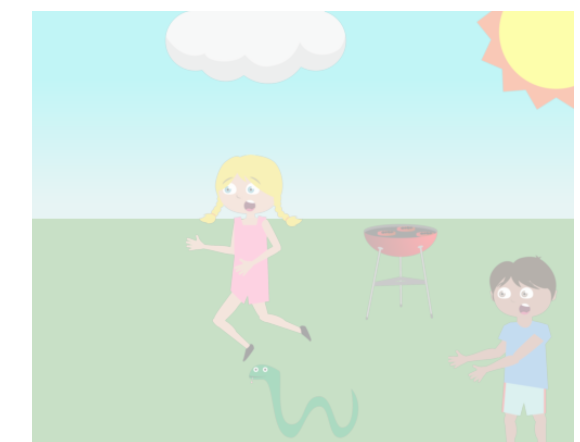
A literal listener (L0)
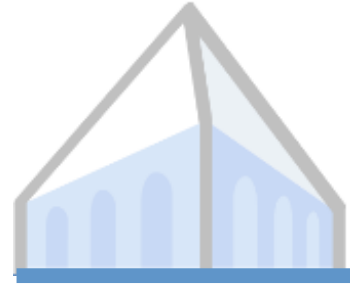
Module architectures

Referent encoder

```
ngram        →  FC  →  desc
features
```

Referent decoder

```
desc  ⟍
        Sum → ReLU → FC → Softmax → choice
referent ⟋
```

Mike is holding a baseball bat

(**random** distractor)
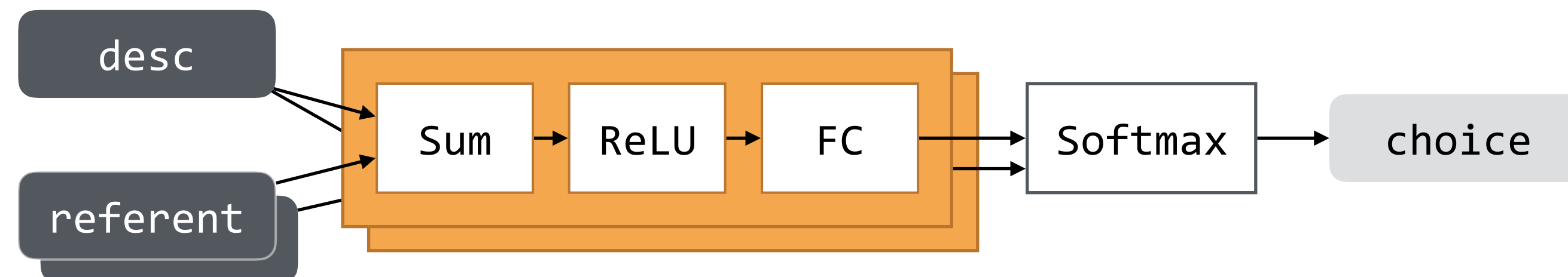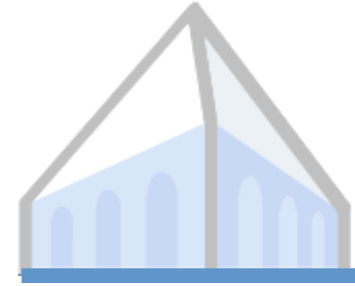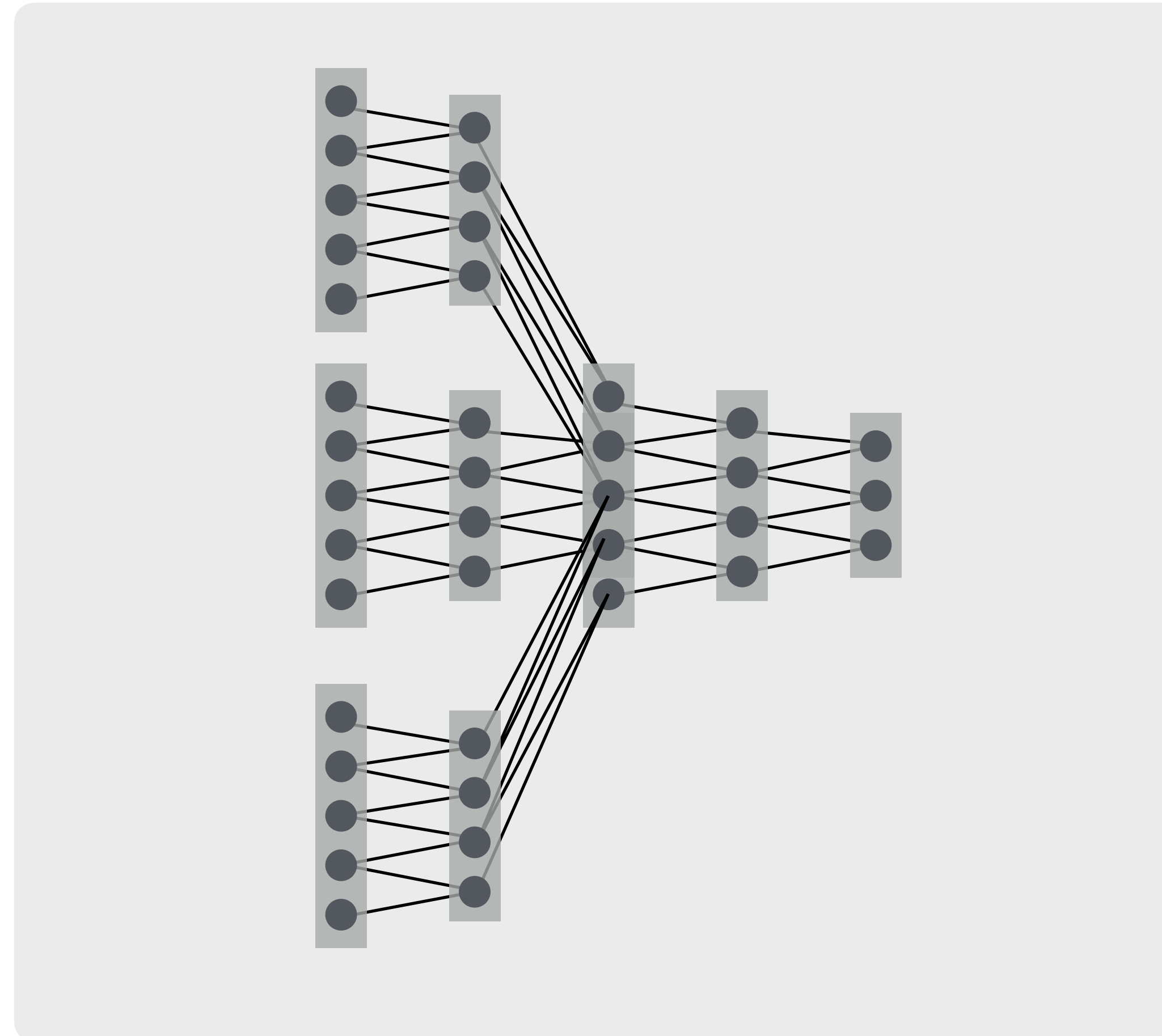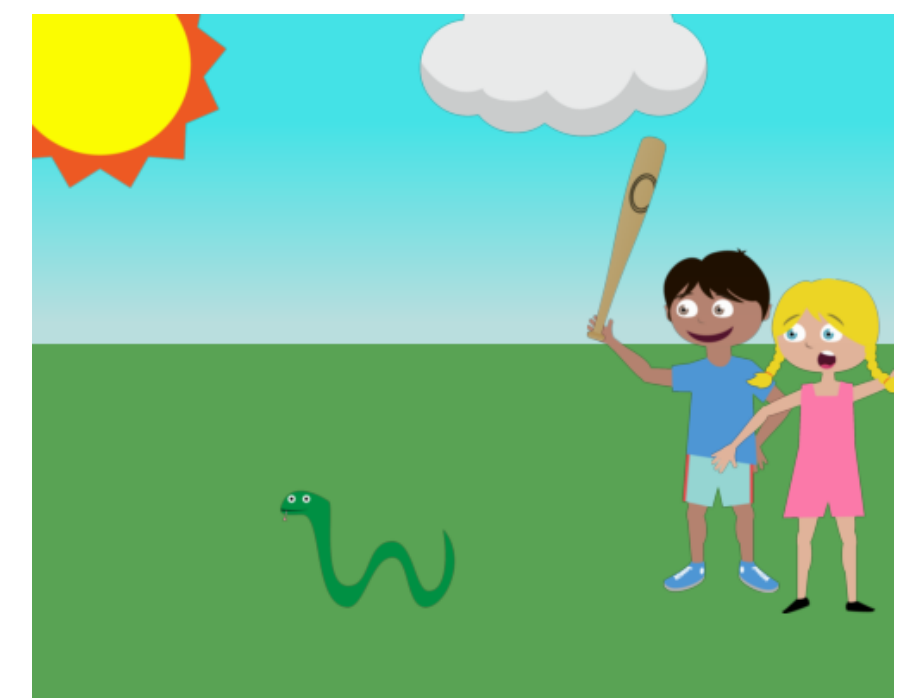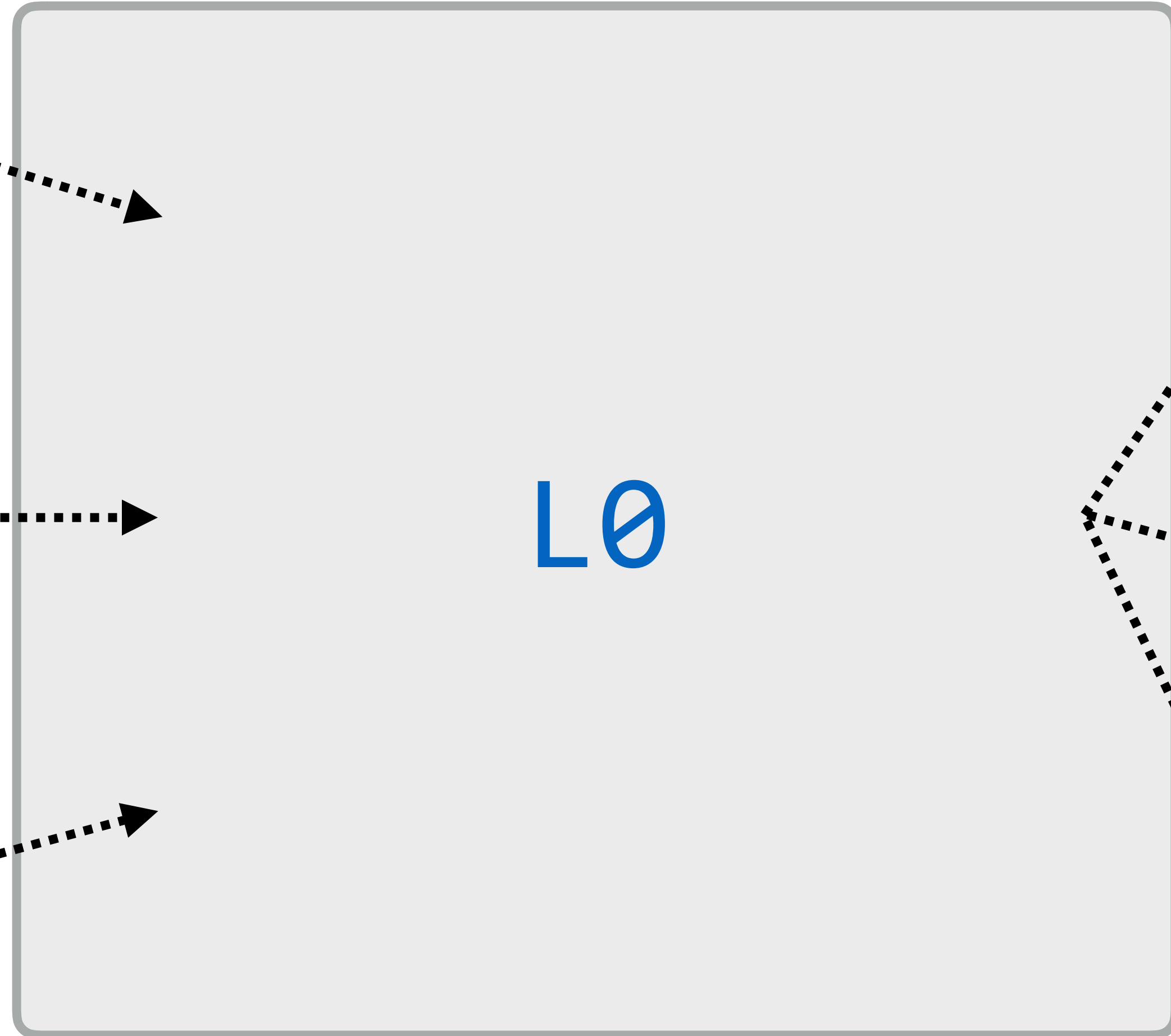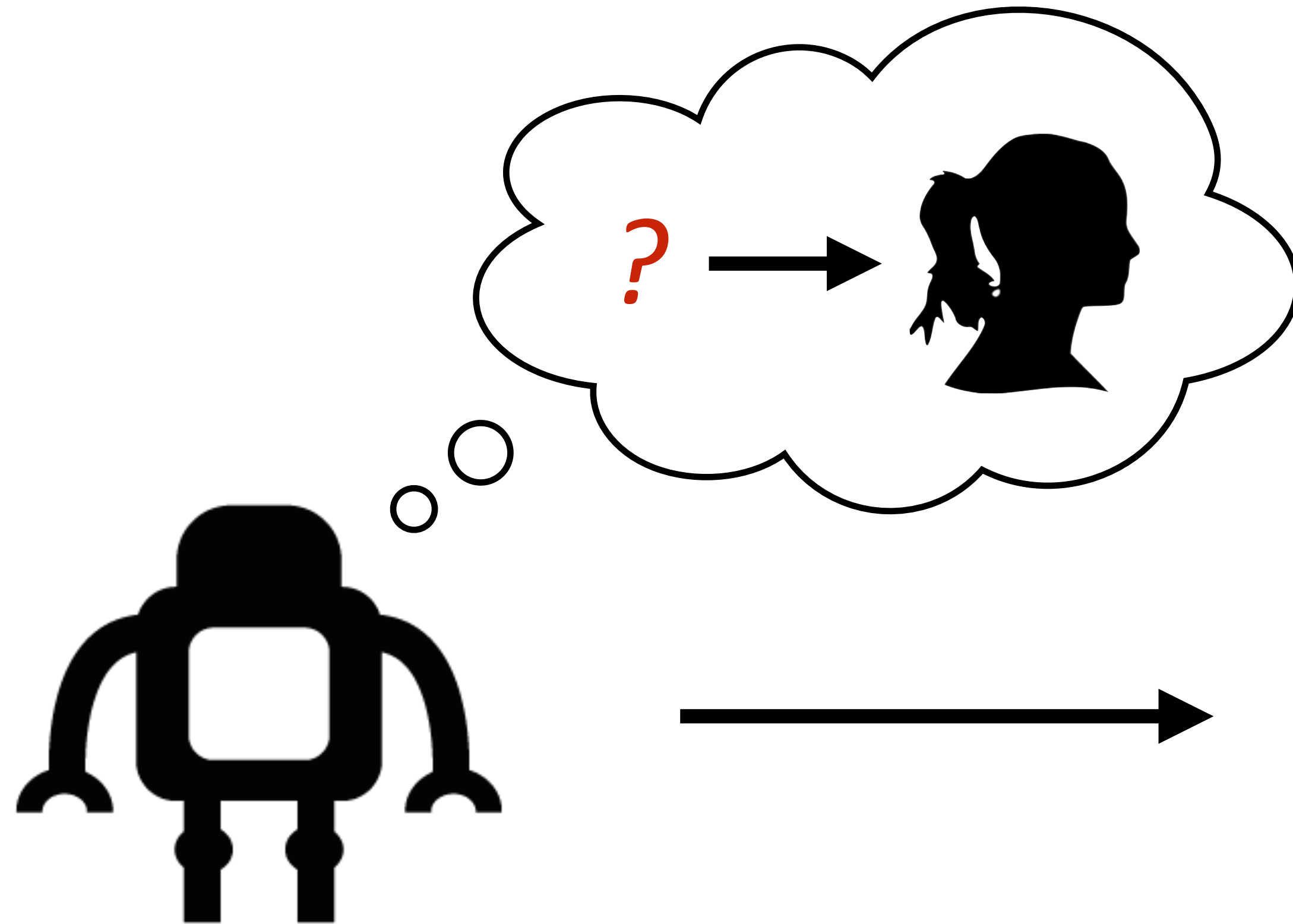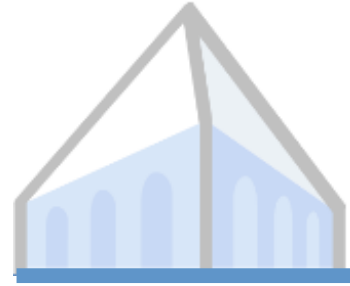
0.87

A literal listener (L0)

Mike is holding a baseball bat

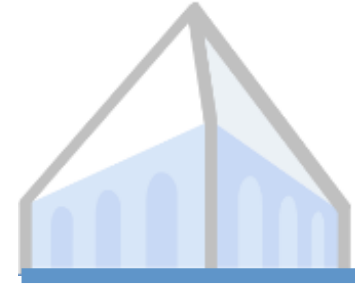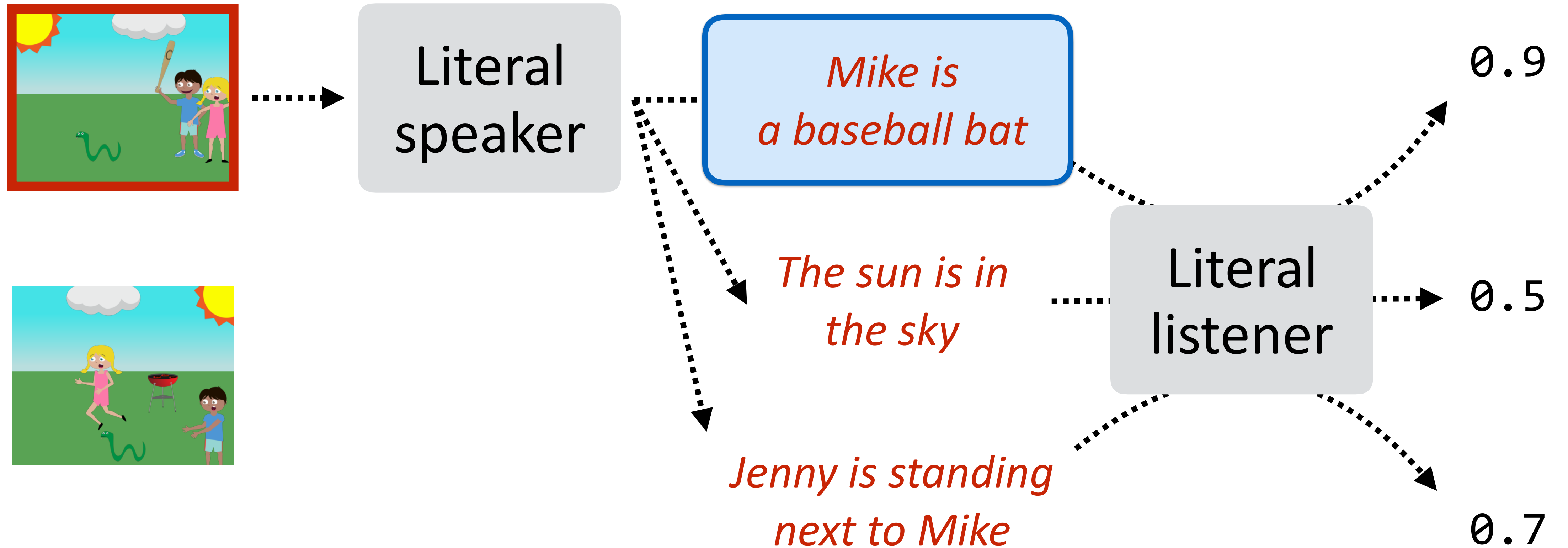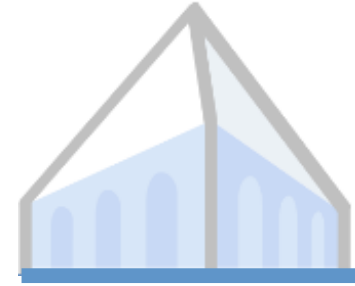# A reasoning speaker (S1)

A reasoning speaker (S1)
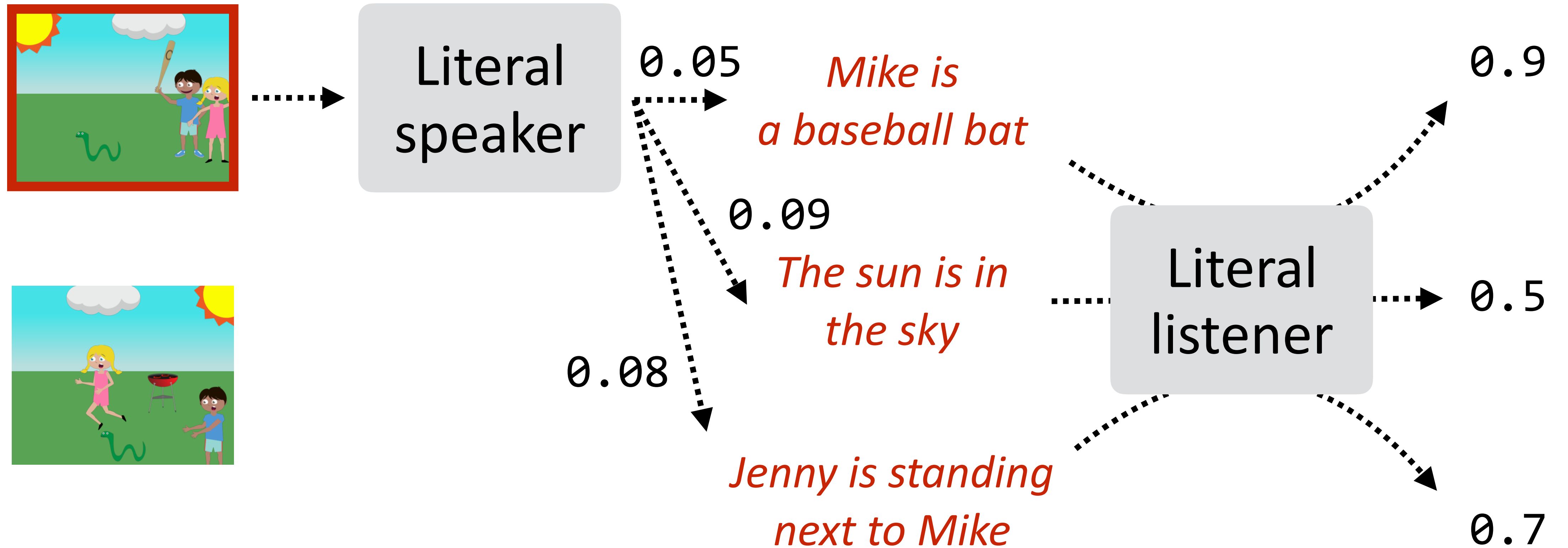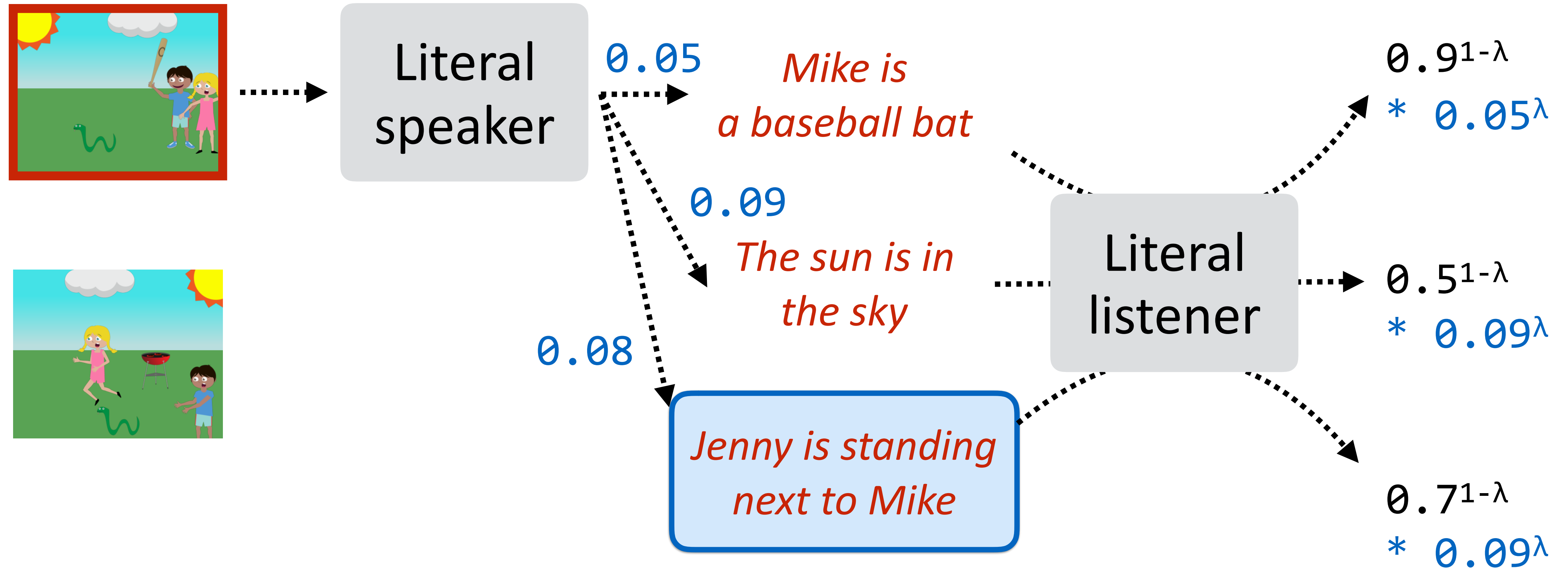
Literal speaker

0.05 → *Mike is a baseball bat*

0.09 → *The sun is in the sky*

0.08 → *Jenny is standing next to Mike*

Literal listener

0.9

0.5

0.7

A reasoning speaker (S1)

Literal speaker

0.05 → *Mike is a baseball bat*

0.09 → *The sun is in the sky*

0.08 → *Jenny is standing next to Mike*

Literal listener

$0.9^{1-\lambda}$ * 0.05^{λ}

$0.5^{1-\lambda}$ * 0.09^{λ}

$0.7^{1-\lambda}$ * 0.09^{λ}

Experiments
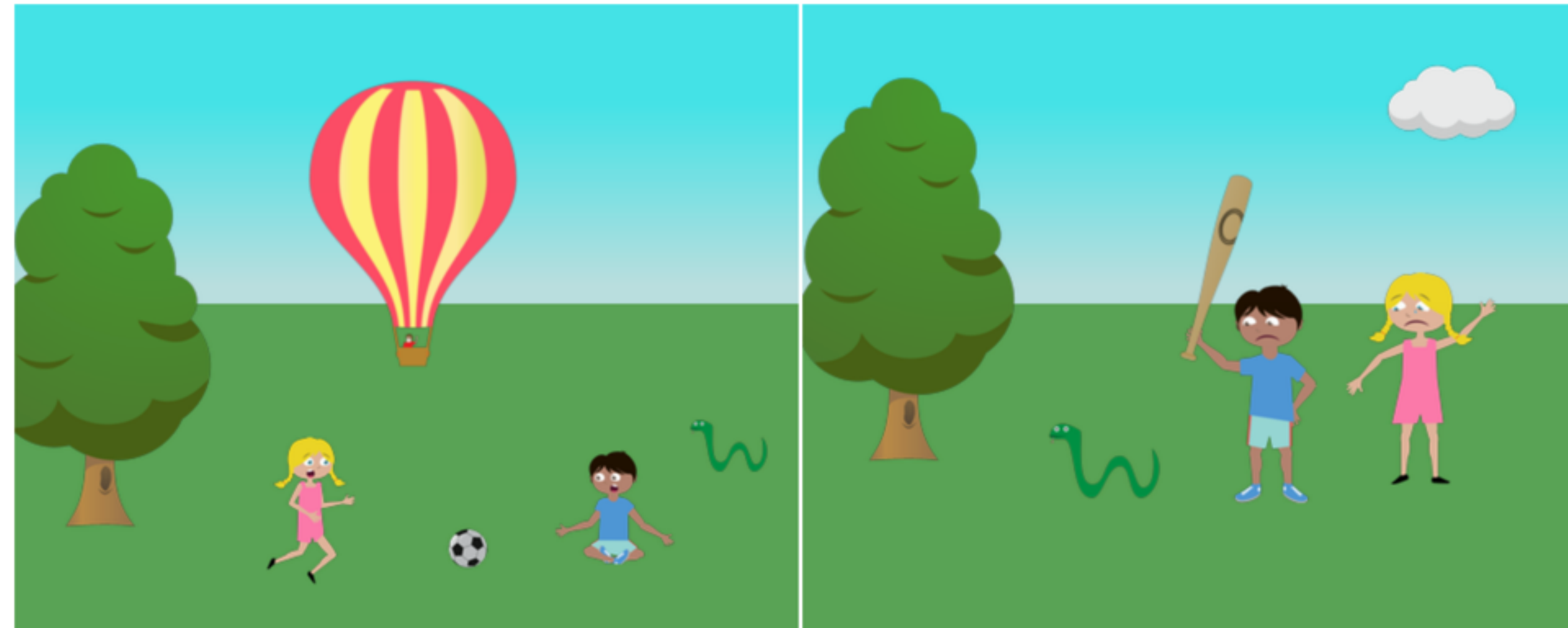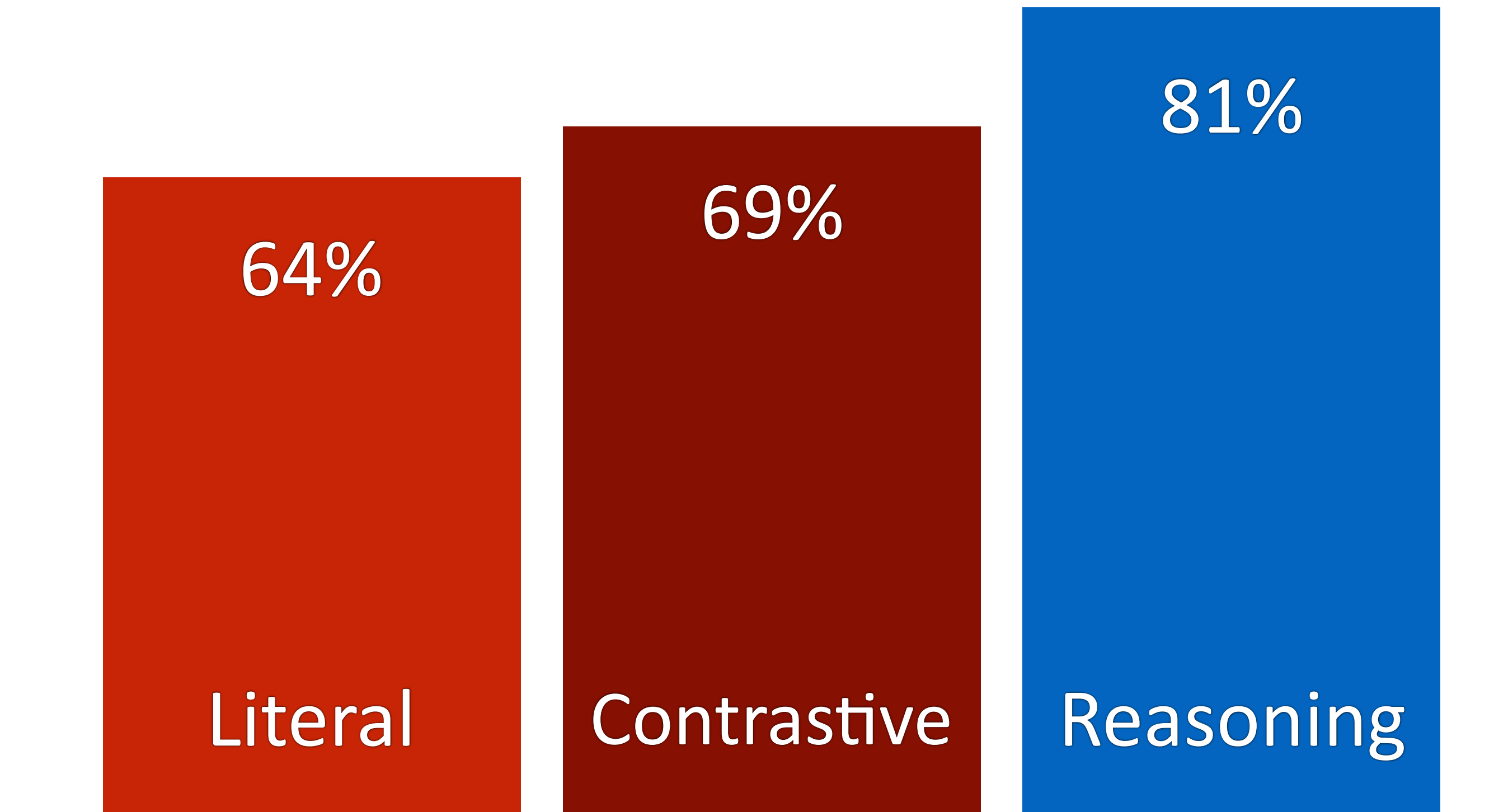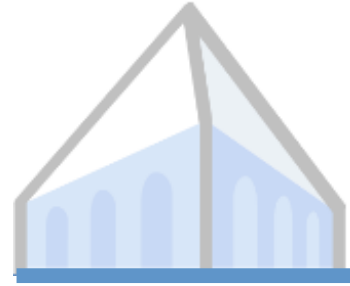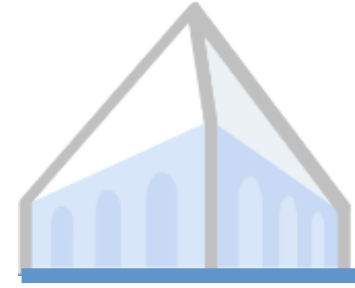
Baselines

- **Literal**: the L0 model by itself

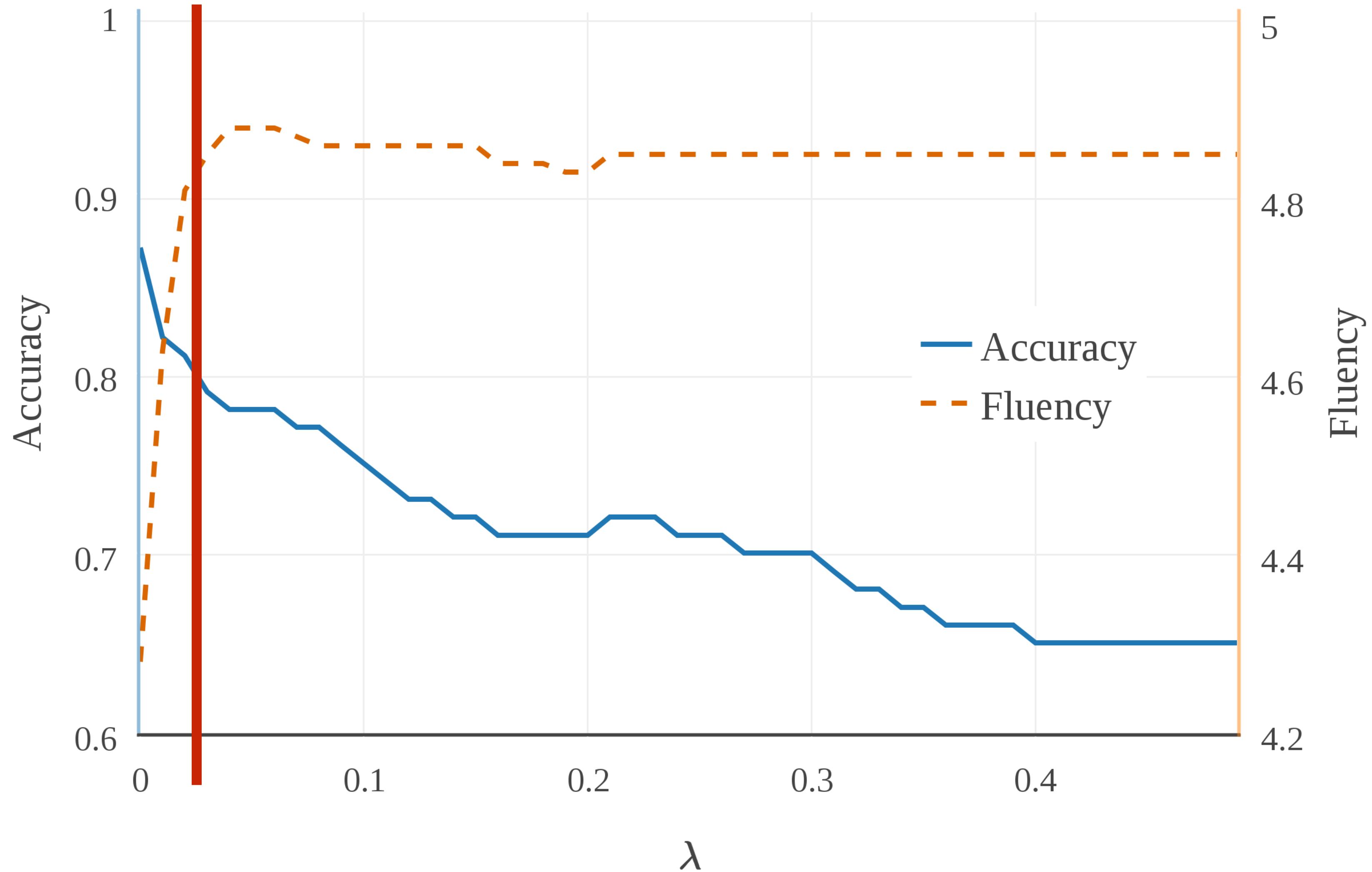- **Contrastive**: a conditional LM trained on both the target image and a random distractor [Mao et al. 2015]
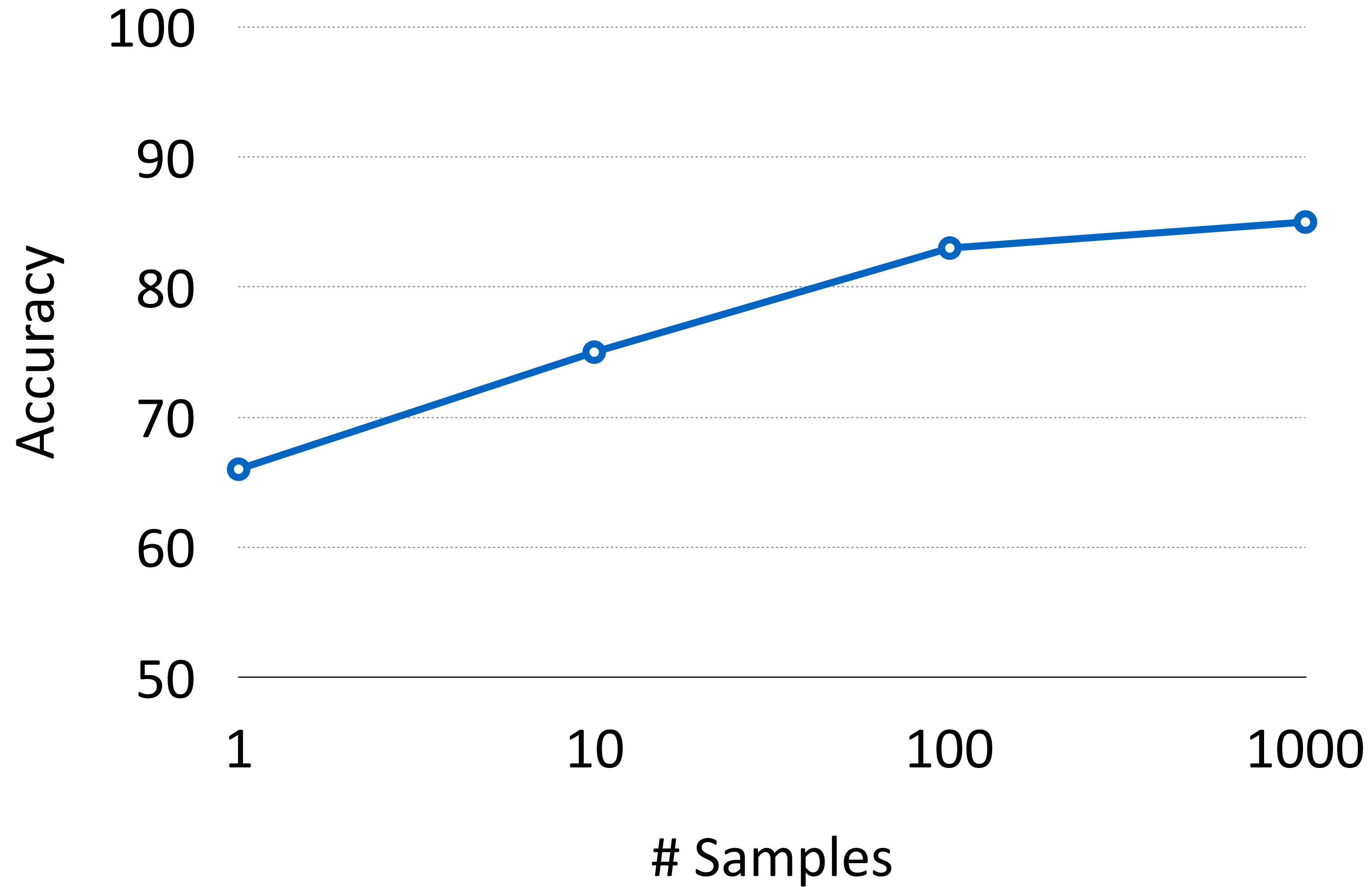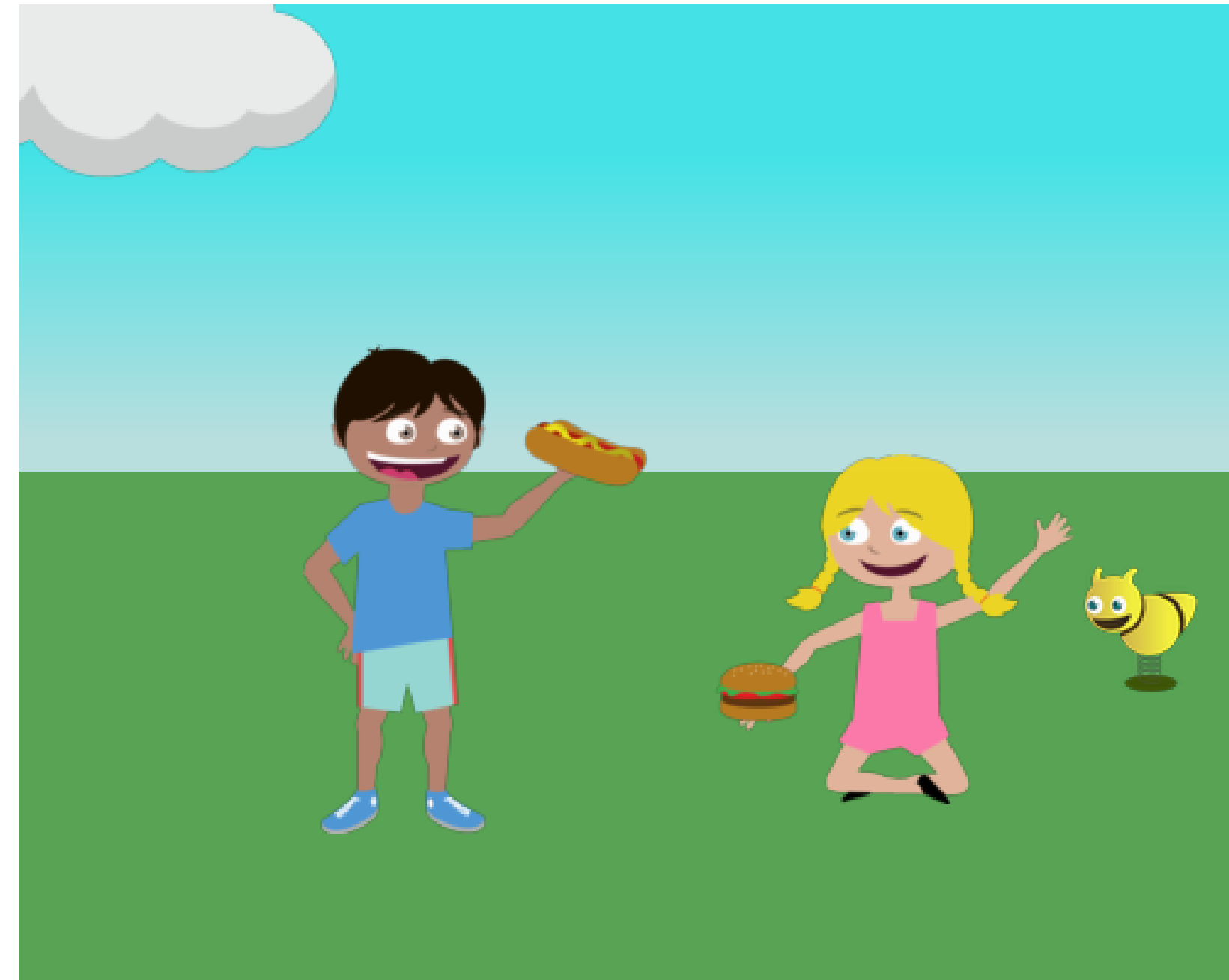
Results (test)

Accuracy and fluency

How many samples?
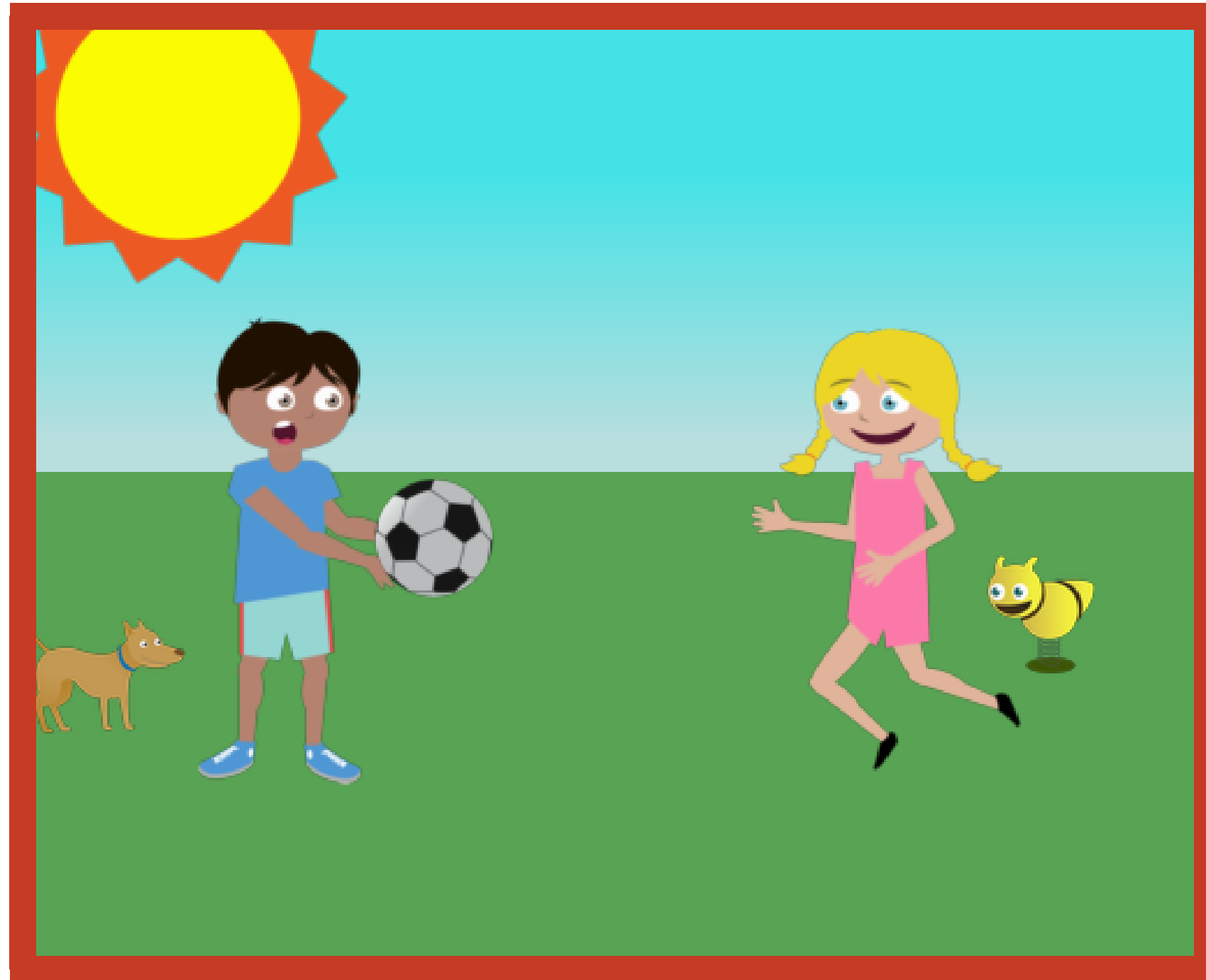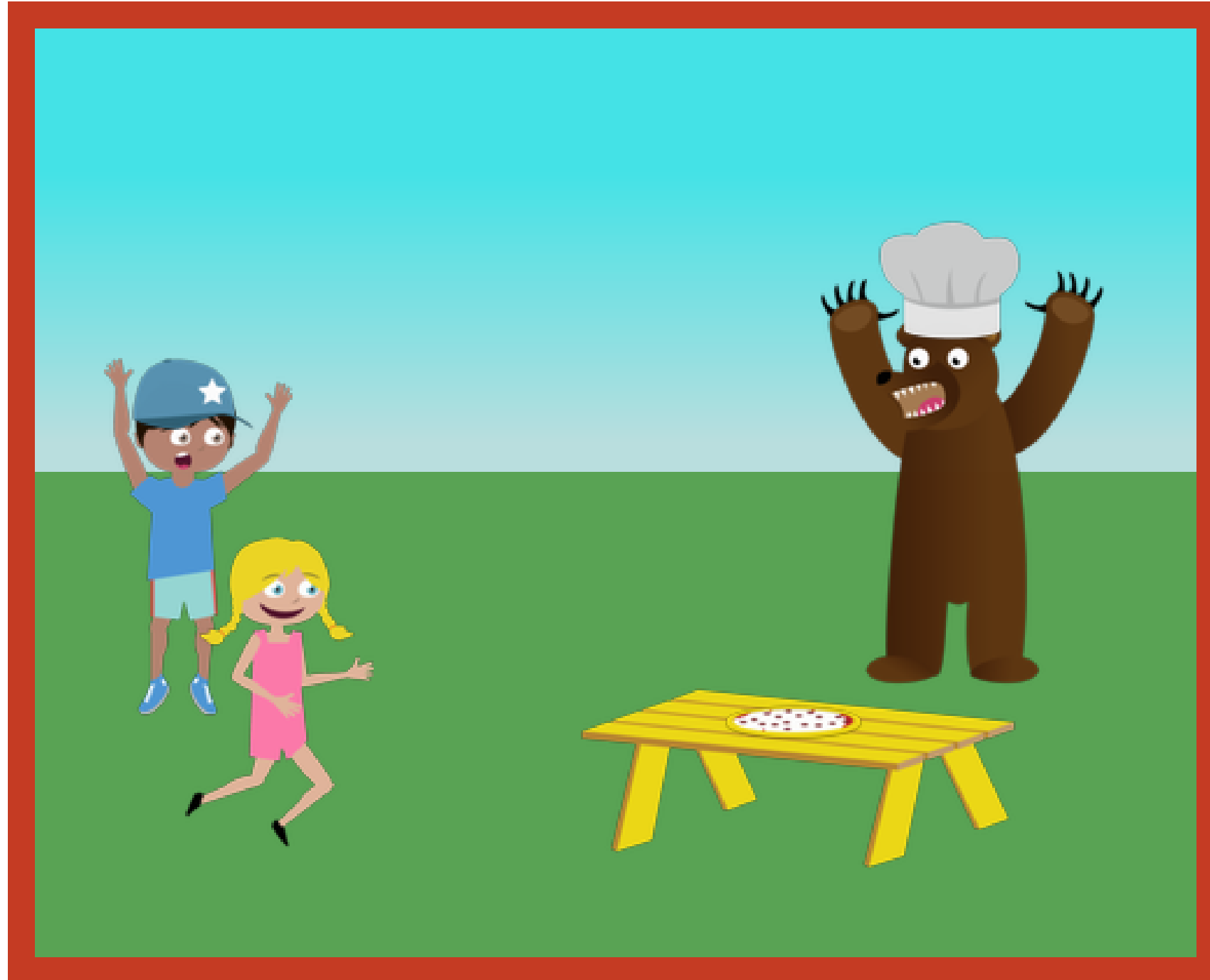
Examples

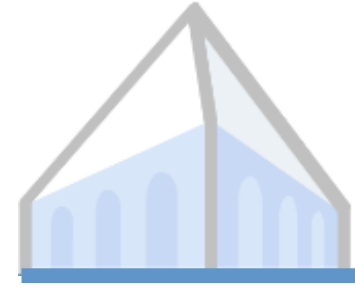*the sun is in the sky*

the dog is standing beside jenny
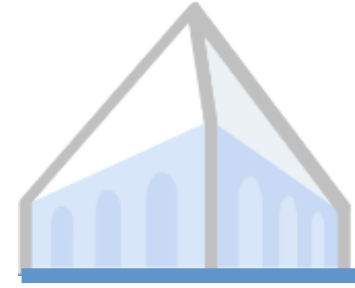
Examples

mike is wearing a chef's hat

Conclusions

- Standard neural kit of parts for base models

- Probabilistic reasoning for high-level goals
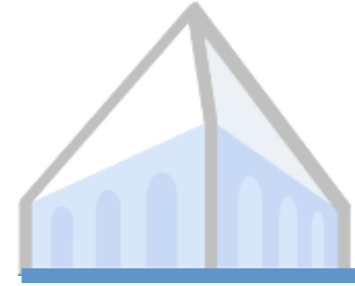
- A little bit of structure goes a long way!

Thank you!

"Compiling" the reasoning model

What if we train the contrastive model on the output of the reasoning model?

Results (dev)