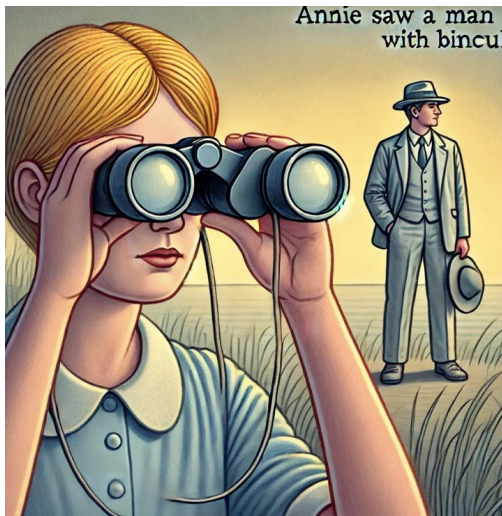


Ambiguity

In natural languages, one sentence may have multiple readings, a phenomenon known as 'ambiguity'.

1. Annie saw a man with binoculars.

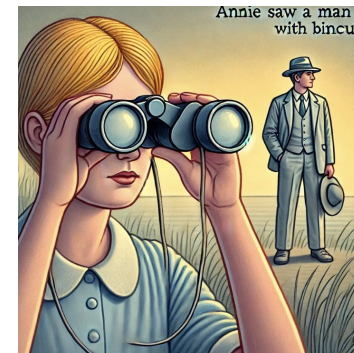


Structural ambiguity

This kind of ambiguity is due to two possible ways of analyzing the structure of the sentence.

1. Annie saw a man with binoculars.

[Annie saw a man] [with binoculars]



[Annie saw [a man with binoculars]]

What a quantificational sentence mean

2. **Every** Singaporean speaks English.



What a quantificational sentence mean

2. **Every** Singaporean speaks English.
3. **Every** Singaporean speaks **two** languages.



Singaporean citizen	languages
Sue	English, Mandarin
Nur	English, Malay
Peter	English, Mandarin
Johnson	English, Tamil
...	...

Ambiguities triggered by quantifiers

4. Every student watched a movie.



Ambiguities triggered by quantifiers

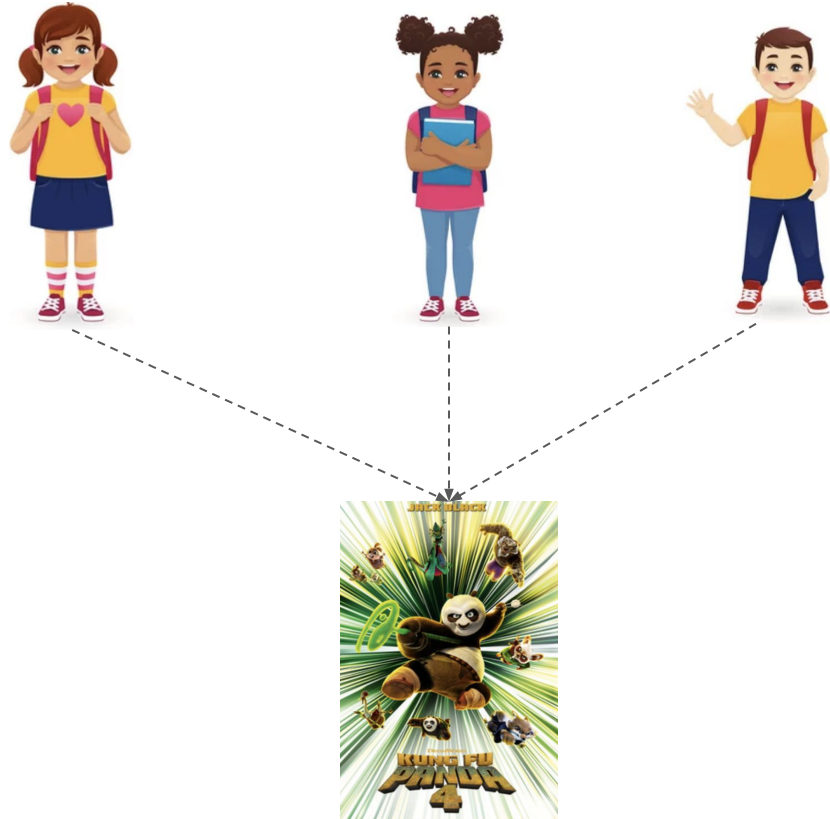
4. Every student watched a movie.

For every student, there is a movie such that the student watched it.



Ambiguities triggered by quantifiers

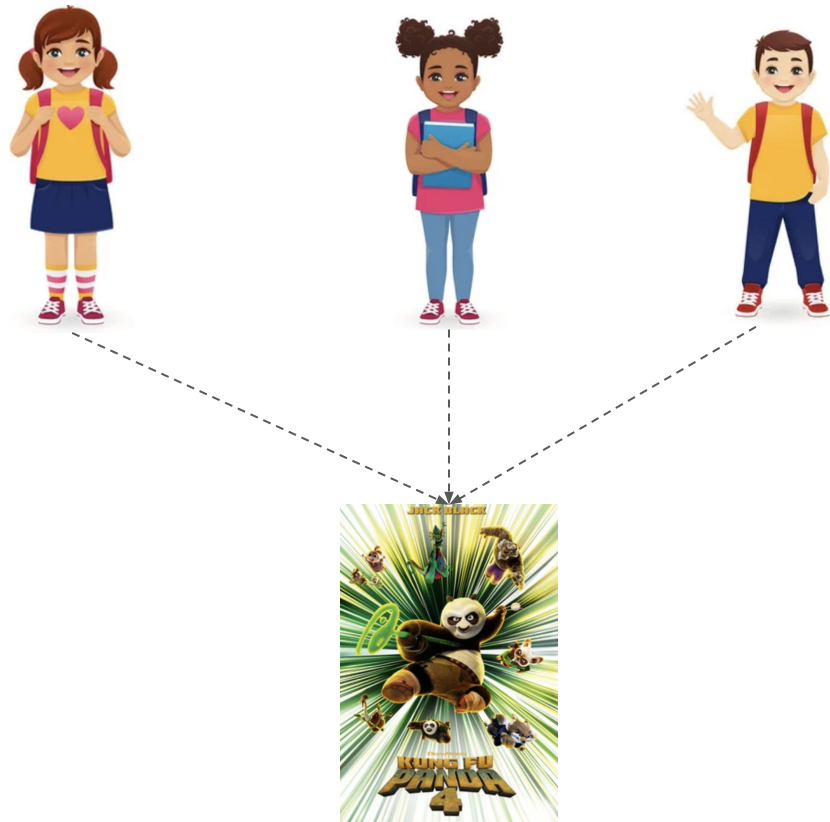
4. Every student watched a movie.



Ambiguities triggered by quantifiers

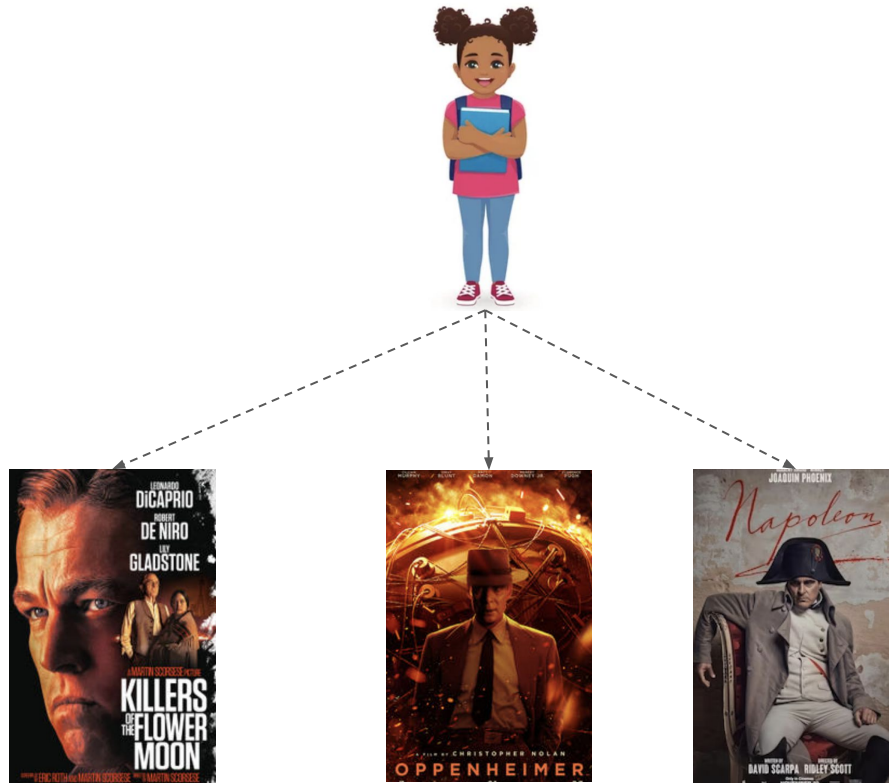
4. Every student watched a movie.

There is a movie such that every student watched it.



Multiple quantifiers

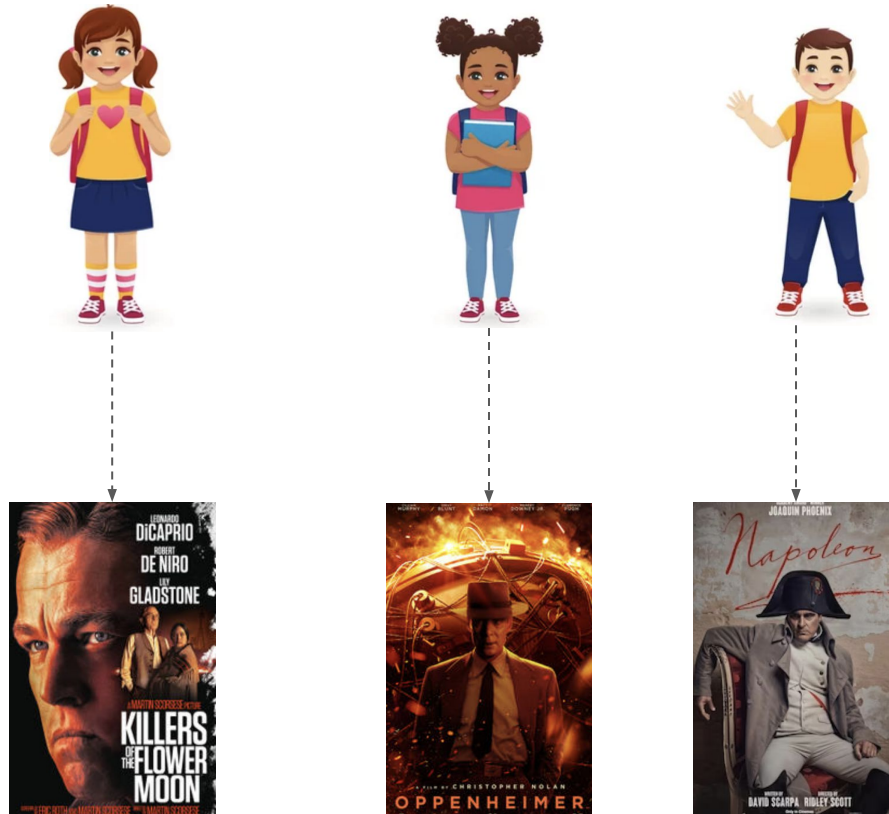
5. A student watched every movie.



There is a student who watched all movies

Multiple quantifiers

5. A student watched every movie.



For every movie, there is a student who watched it.

Multiple quantifiers

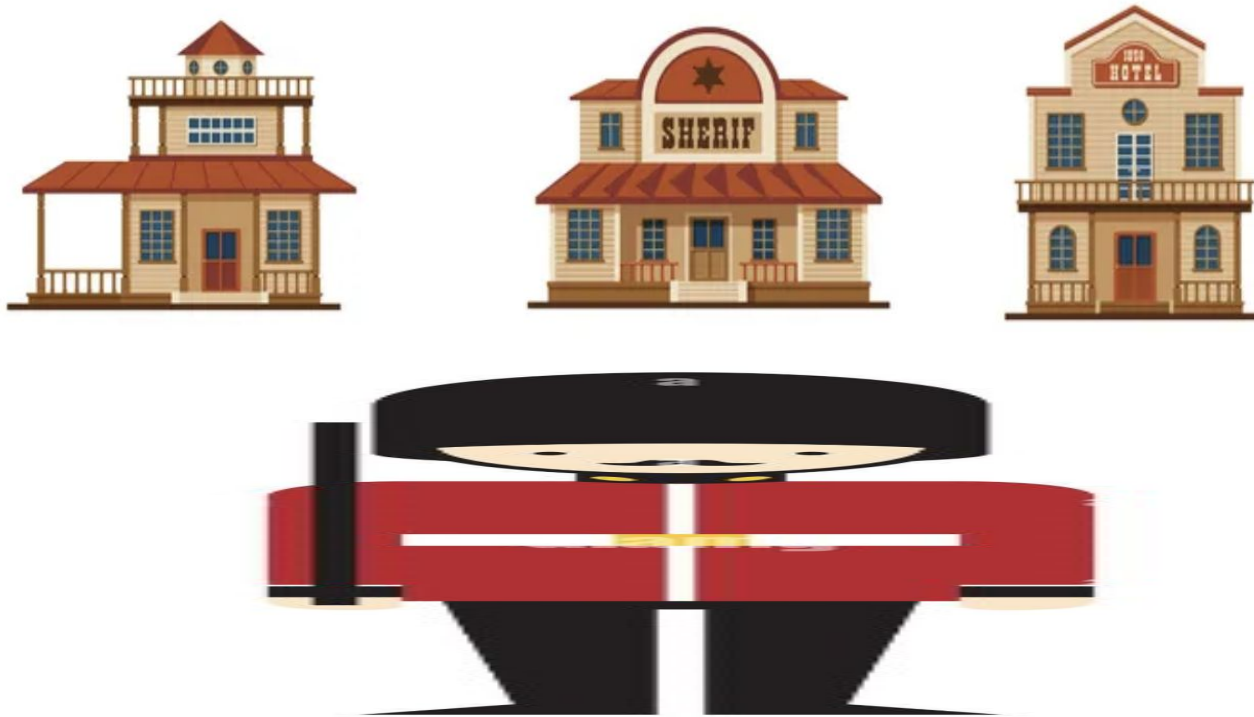
6. A guard is standing in front of every building.



For every building, there is a guard who is standing in front of it

Multiple quantifiers

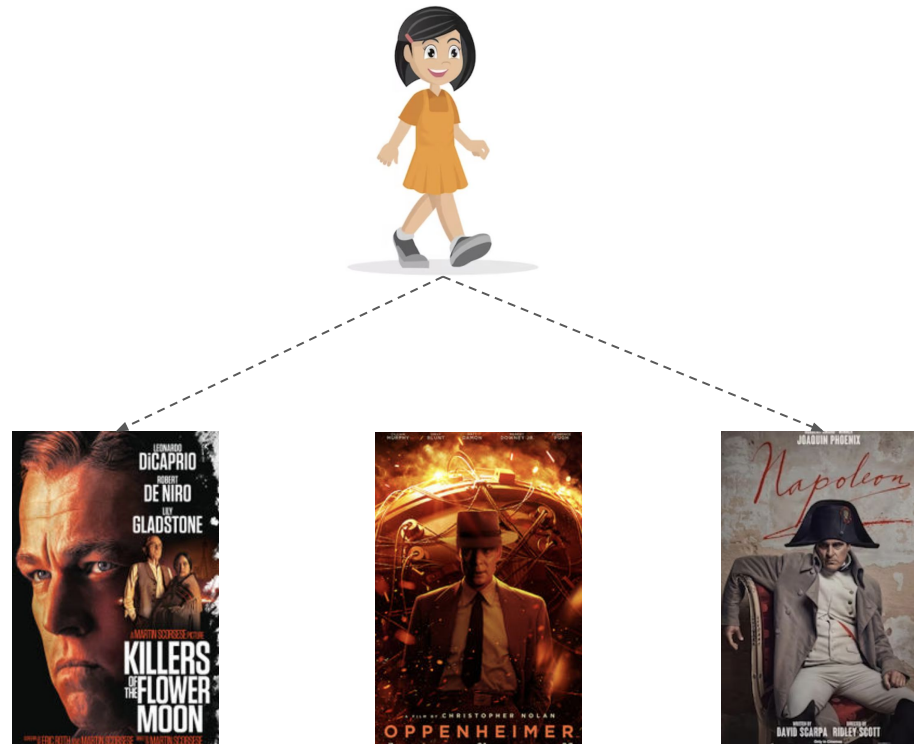
6. A guard is standing in front of every building.



There is a guard who is standing in front of all buildings.

Quantifiers and negation

7. Annie didn't watch every movie.



It is not the case that Annie watched all movies

Quantifiers and negation

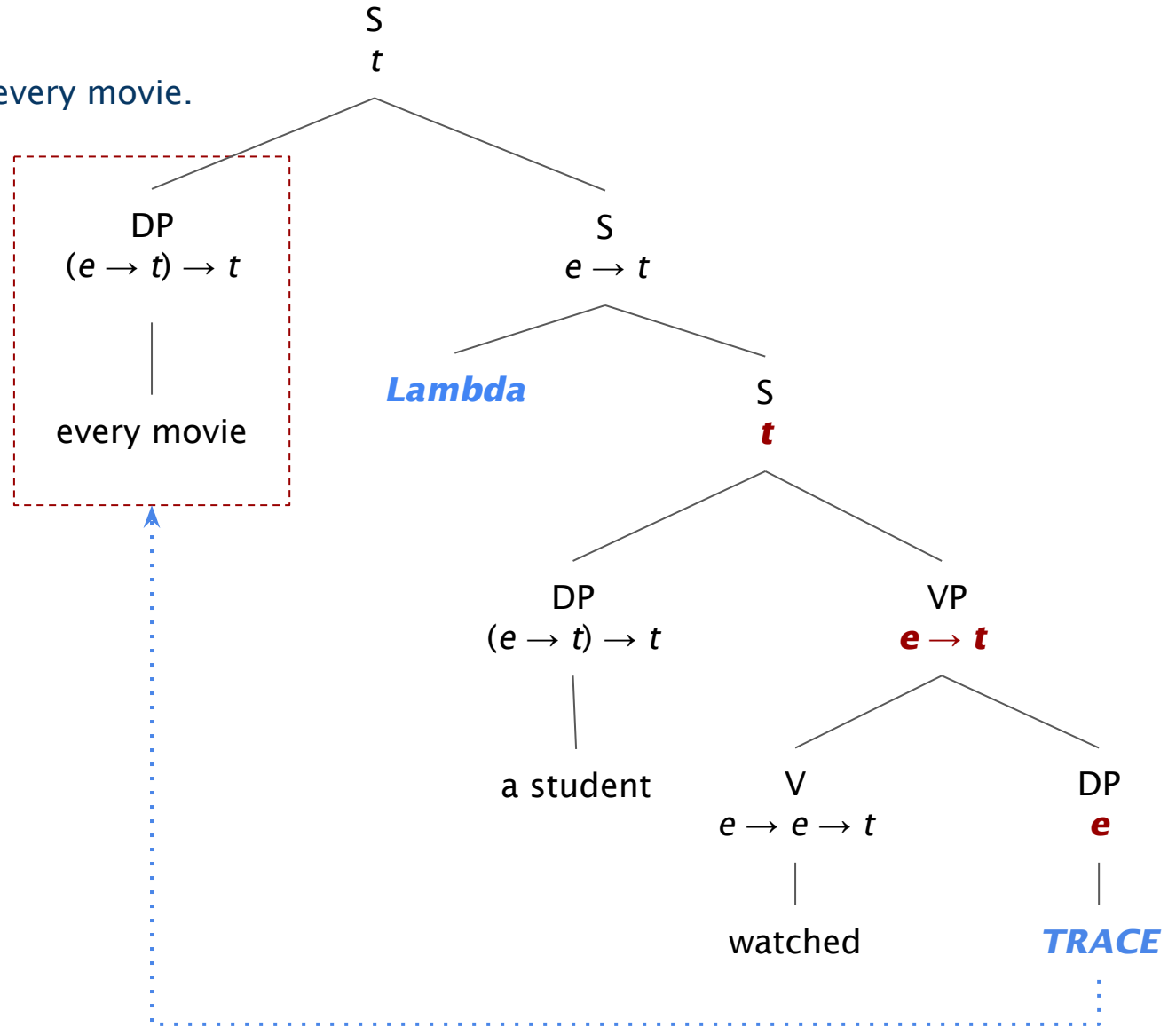
7. Annie didn't watch every movie.



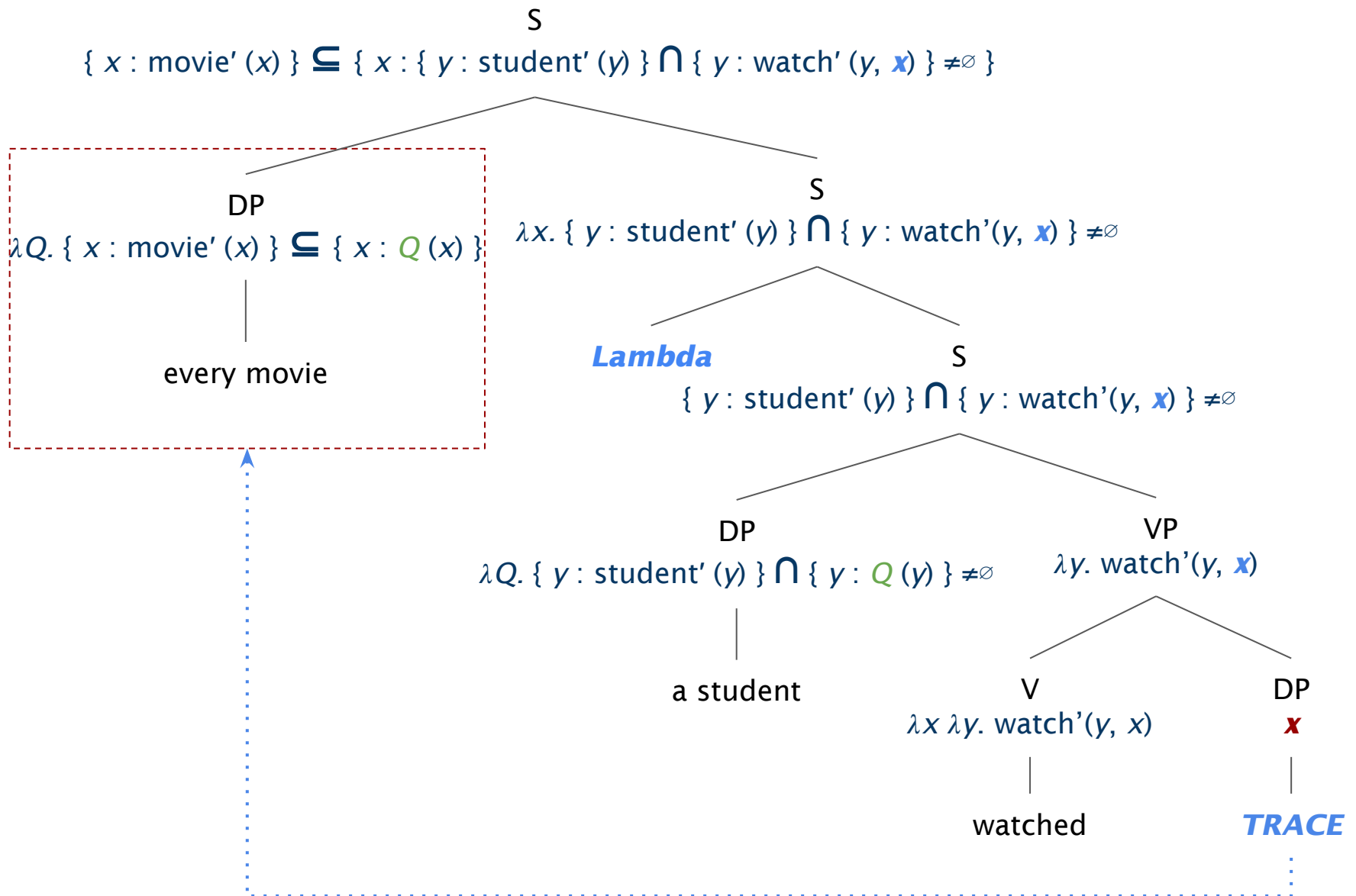
For every movie, Anne didn't watch it.

Quantifier Raising

A student watched every movie.



Quantifier Raising



Quantifier Raising

$$\{x : \text{movie}'(x)\} \subseteq \{x : \{y : \text{student}'(y)\} \cap \{y : \text{watch}'(y, x)\} \neq \emptyset\}$$



Quantifier Raising

$$\{x : \text{movie}'(x)\} \subseteq \{x : \{y : \text{student}'(y)\} \cap \{y : \text{watch}'(y, x)\} \neq \emptyset\}$$

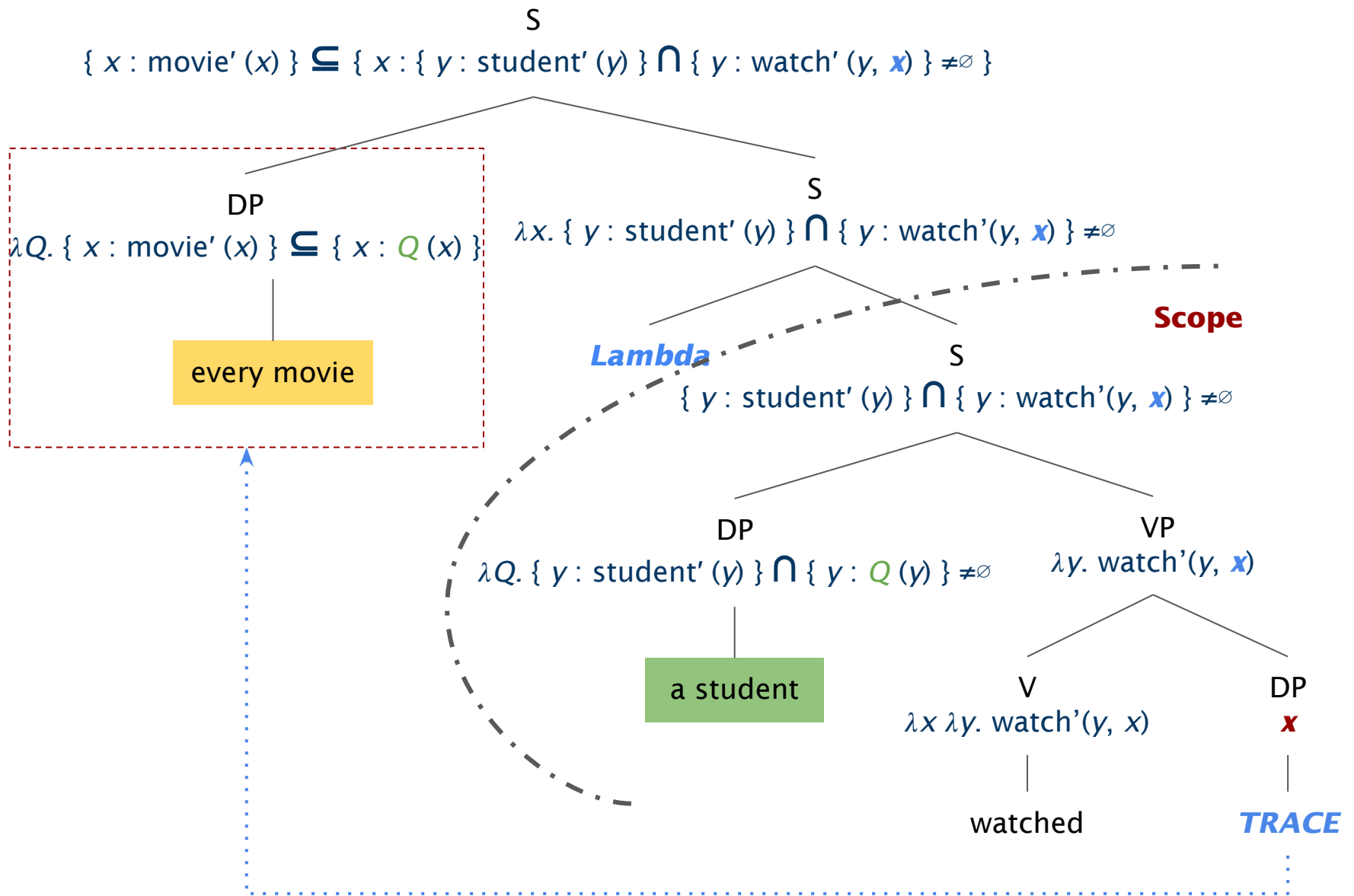


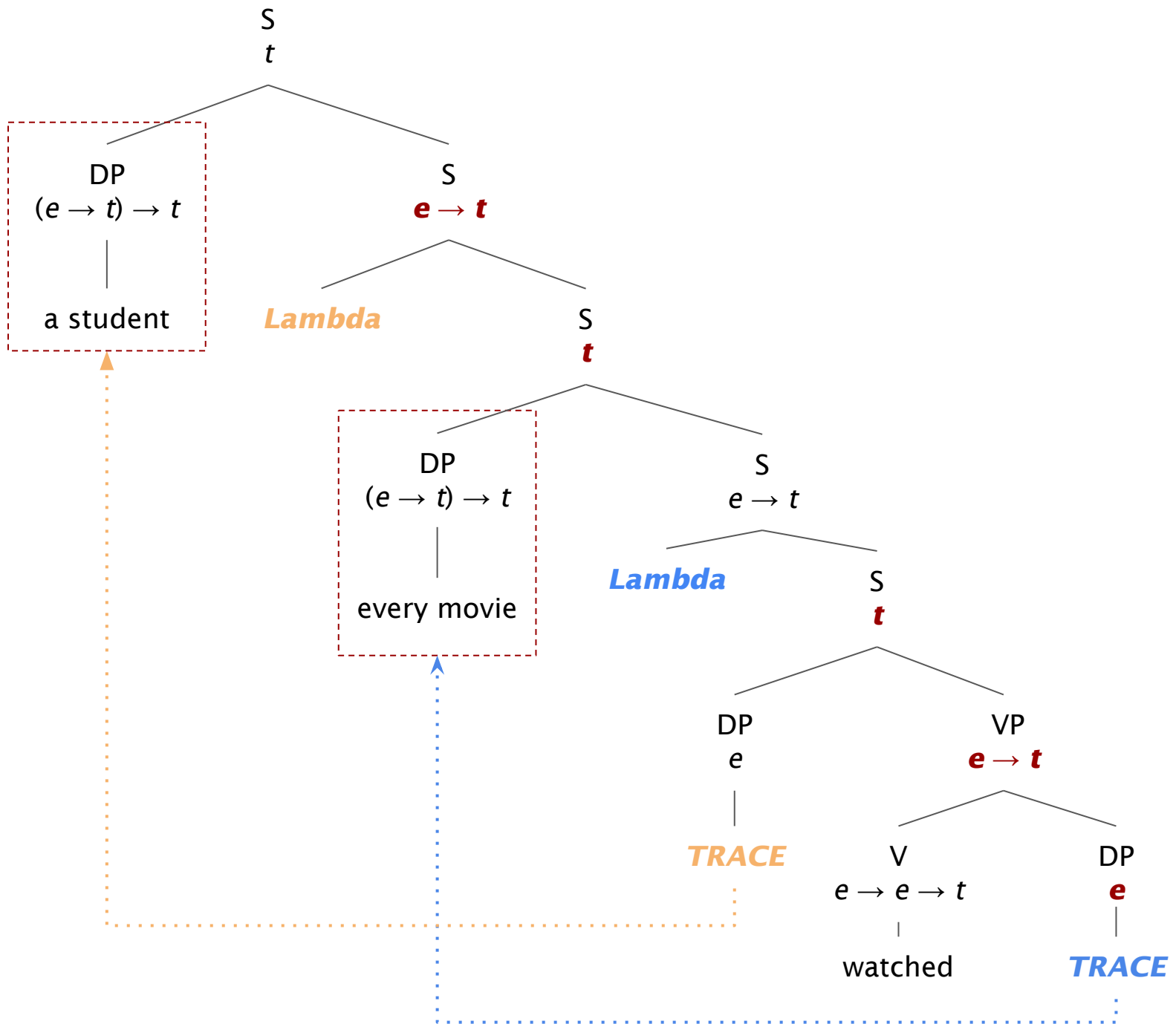
Quantifier Raising

$$\{x : \text{movie}'(x)\} \subseteq \{x : \{y : \text{student}'(y)\} \cap \{y : \text{watch}'(y, x)\} \neq \emptyset\}$$



Quantifier Raising





$$\{ y : \text{student}'(y) \} \cap \{ y : \{ x : \text{movie}'(x) \} \subseteq \{ x : \text{watch}'(\mathbf{y}, \mathbf{x}) \} \} \neq \emptyset$$

$$\lambda Q. \{ y : \text{student}'(y) \} \cap \{ y : Q(y) \} \neq \emptyset$$

a student

$$\lambda y. \{ x : \text{movie}'(x) \} \subseteq \{ x : \text{watch}'(\mathbf{y}, \mathbf{x}) \}$$

Lambda $\{ x : \text{movie}'(x) \} \subseteq \{ x : \text{watch}'(\mathbf{y}, \mathbf{x}) \}$

$$\lambda Q. \{ x : \text{movie}'(x) \} \subseteq \{ x : Q(x) \} \quad \lambda x. \text{watch}'(\mathbf{y}, \mathbf{x})$$

every movie

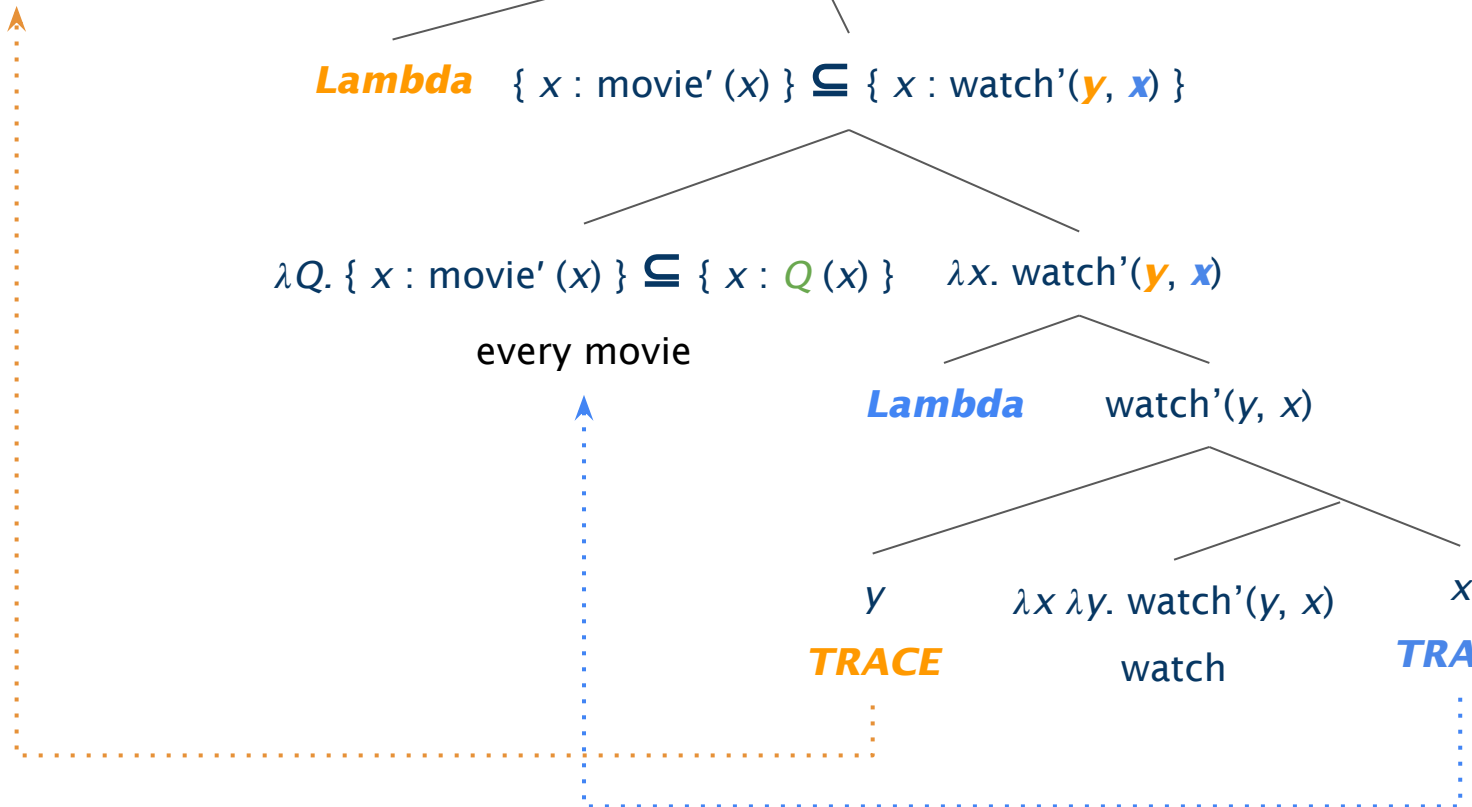
Lambda $\text{watch}'(\mathbf{y}, \mathbf{x})$

y $\lambda x \lambda y. \text{watch}'(\mathbf{y}, \mathbf{x})$ x

TRACE

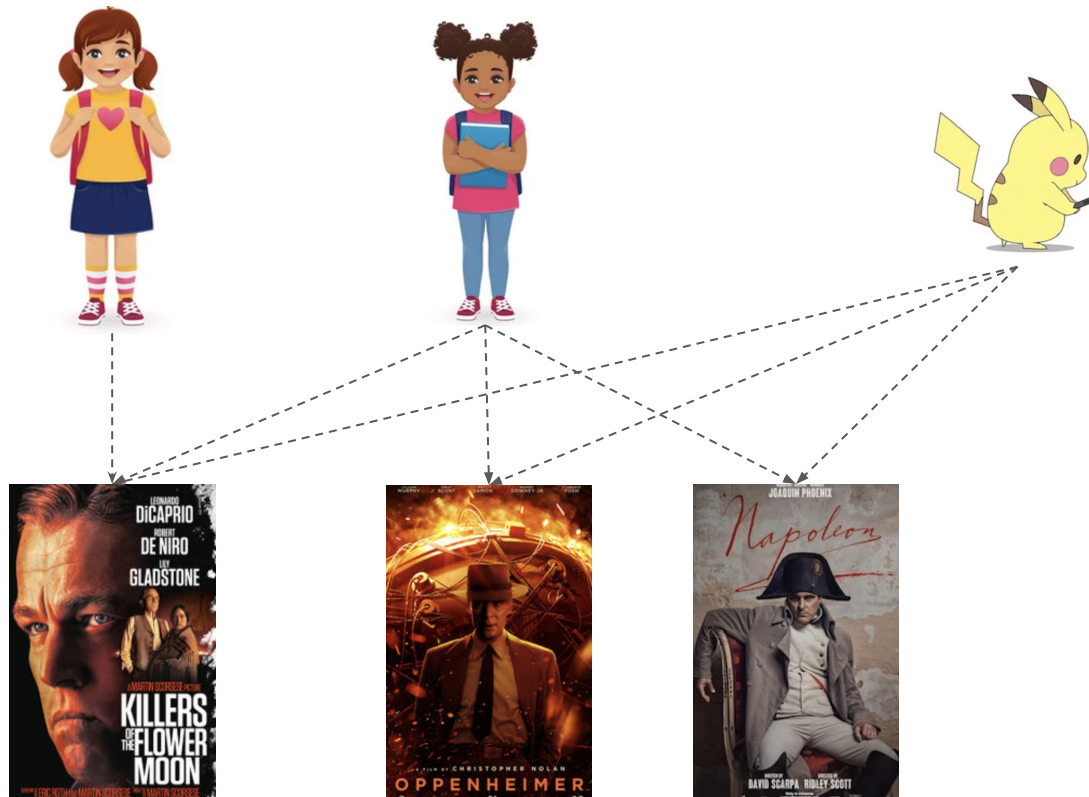
watch

TRACE



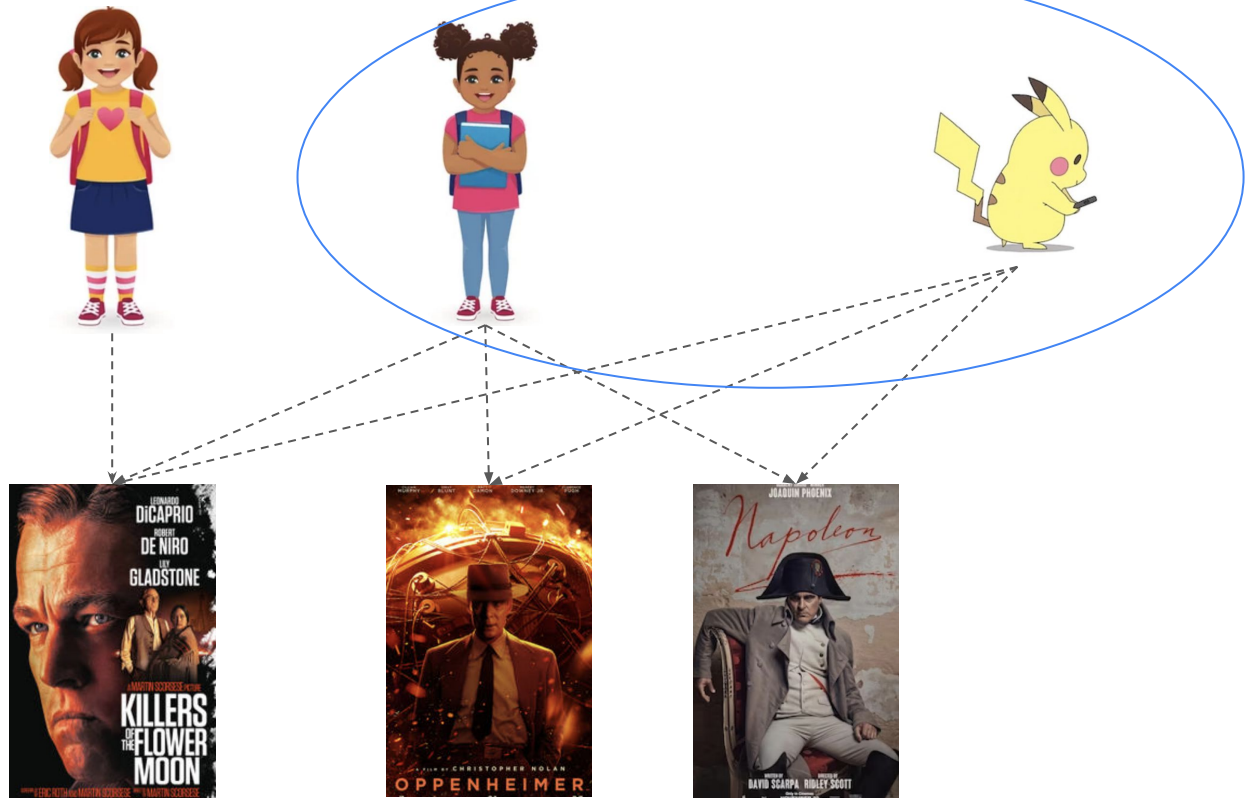
Quantifier Raising

$$\{ y : \text{student}'(y) \} \cap \{ y : \{ x : \text{movie}'(x) \} \subseteq \{ x : \text{watch}'(\mathbf{y}, x) \} \} \neq \emptyset$$



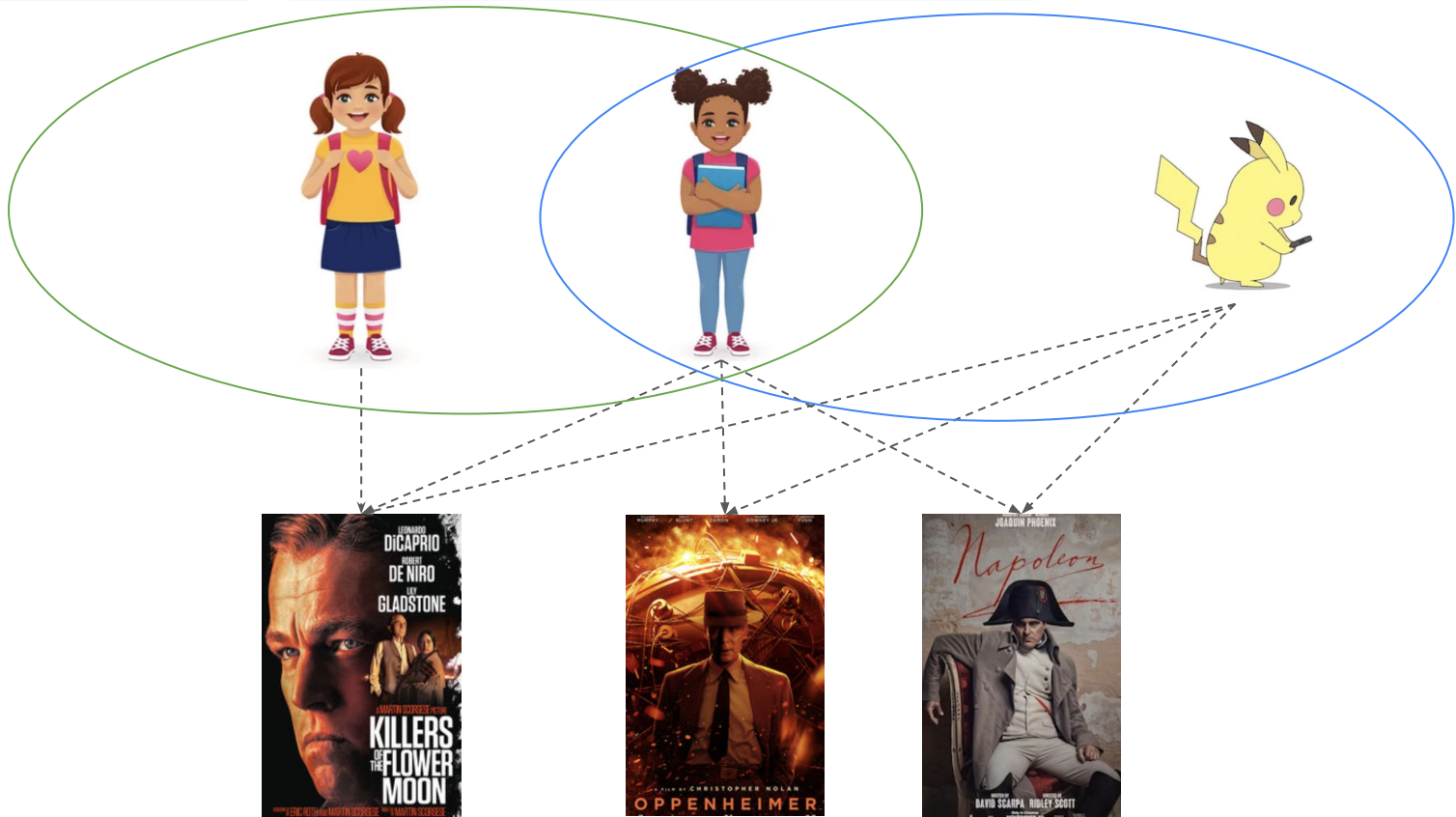
Quantifier Raising

$$\{ y : \text{student}'(y) \} \cap \{ y : \{ x : \text{movie}'(x) \} \subseteq \{ x : \text{watch}'(y, x) \} \} \neq \emptyset$$



Quantifier Raising

$$\{y : \text{student}'(y)\} \cap \{y : \{x : \text{movie}'(x)\} \subseteq \{x : \text{watch}'(y, x)\}\} \neq \emptyset$$



Quantifier Raising

$$\{ y : \text{student}'(y) \} \cap \{ y : \{ x : \text{movie}'(x) \} \subseteq \{ x : \text{watch}'(y, x) \} \} \neq \emptyset$$



Quantifier Raising

Empty Set

$$\{ y : \text{student}'(y) \} \cap \{ y : \{ x : \text{movie}'(x) \} \subseteq \{ x : \text{watch}'(y, x) \} \} \neq \emptyset$$



$$\{ y : \text{student}'(y) \} \cap \{ y : \{ x : \text{movie}'(x) \} \subseteq \{ x : \text{watch}'(\mathbf{y}, \mathbf{x}) \} \} \neq \emptyset$$

$$\lambda Q. \{ y : \text{student}'(y) \} \cap \{ y : Q(y) \} \neq \emptyset$$

a student

$$\lambda y. \{ x : \text{movie}'(x) \} \subseteq \{ x : \text{watch}'(\mathbf{y}, \mathbf{x}) \}$$

Lambda

$$\{ x : \text{movie}'(x) \} \subseteq \{ x : \text{watch}'(\mathbf{y}, \mathbf{x}) \}$$

Scope

$$\lambda Q. \{ x : \text{movie}'(x) \} \subseteq \{ x : Q(x) \}$$

every movie

$$\lambda x. \text{watch}'(\mathbf{y}, \mathbf{x})$$

Lambda

watch'(y, x)

y

$\lambda x \lambda y. \text{watch}'(y, x)$

x

TRACE

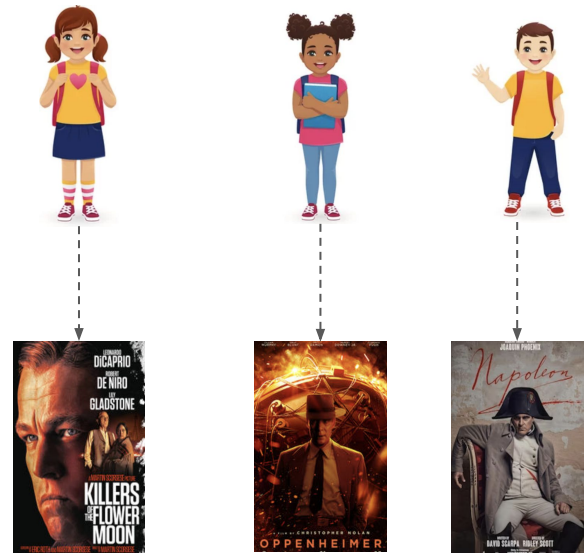
watch

TRACE

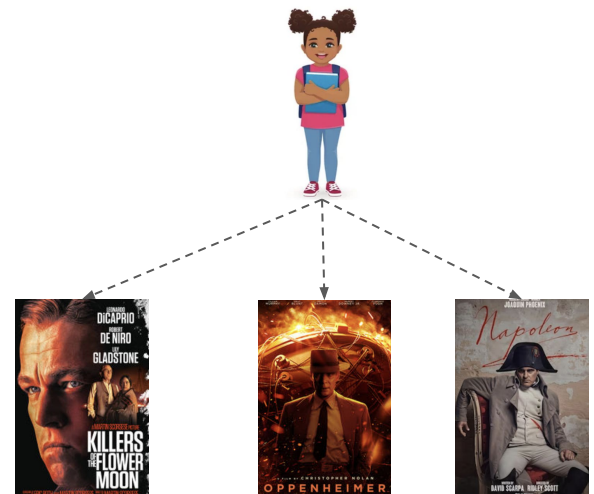
Scope ambiguity

A student watched every movie.

“a student” is inside the scope of “every movie”:



“every movie” is inside the scope of “a student”:



Scope ambiguity

Annie didn't watch every movie.

[every movie] *Lambda* [Annie didn't watch *TRACE*]

For every movie, Annie didn't watch it.



[not] [every movie] *Lambda* [Annie watch *TRACE*]

It is not the case that Annie watched every movie.



Exercise

Compute the two interpretations of the following sentence.

1. Most students read a book.
 - a. For more than half of the students, each of them read a different book.
 - b. There is a book such that more than half of the students read it.

Hungarian: A language with overt QR

8. Sok ember mindenkit szeret.

many man everyone loves

‘There are many men who loves everyone.’

[many men] **Lambda** [everyone] **Lambda** [**TRACE** loves **TRACE**]

9. Mindenkit sok ember szeret.

everyone many man loves]

‘For everyone, there are many men who loves them.’

[everyone] **Lambda** [many men] **Lambda** [**TRACE** loves **TRACE**]

Discussion

Is Mandarin a scope ambiguity language, like English, or a scope rigid language, like Hungarian?

1. 每个学生都看了一部 电影。
2. 有个学生看了每一部 电影。
3. 小王没有看每一部 电影。