COMP90048: Workshop 6

Eleanor McMurtry, University of Melbourne

Logic Programming

Intro

- Logic programming allows us to solve problems by specifying **what must be true** of the answer.

- It is related to declarative programming: strong type systems use the same form of logic.

Setting the stage

- **Atom:** similar to a value, written in lowercase: `eleanor`

Setting the stage

- **Atom:** similar to a value, written in lowercase: `eleanor`
- **Predicate:** a statement that is true or false.

```
?- staffMember(eleanor).

true.
```

Setting the stage

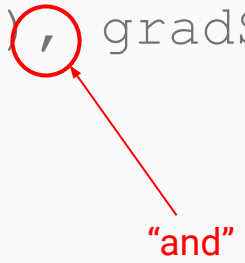
- **Atom:** similar to a value, written in lowercase: `eleanor`
- **Predicate:** a statement that is true or false.
- **Clause:** conjunction of predicates

```
?-staffMember(eleanor), gradStudent(eleanor).

true.
```

Setting the stage

- **Atom:** similar to a value, written in lowercase: `eleanor`
- **Predicate:** a statement that is true or false.
- **Clause:** conjunction of predicates

```
?-staffMember(eleanor), gradStudent(eleanor).

true.
```



Introducing variables

- **Variable:** a term of unknown value, starting with a capital letter

Introducing variables

- **Variable:** a term of unknown value, starting with a capital letter
- Prolog uses **resolution** to try to assign values to variables.

Introducing variables

- **Variable:** a term of unknown value, starting with a capital letter
- Prolog uses **resolution** to try to assign values to variables.

```
?- staffMember(Person), gradStudent(Person).

Person = eleanor;

Person = mak.
```

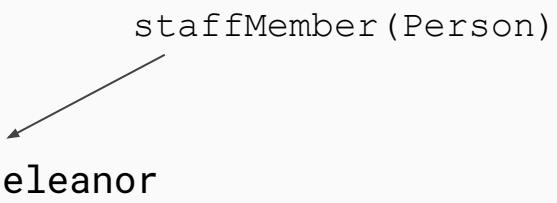
Introducing variables

- **Variable:** a term of unknown value, starting with a capital letter
- Prolog uses **resolution** to try to assign values to variables.

```
?- staffMember(Person), gradStudent(Person).

Person = eleanor ;

Person = mak.
```

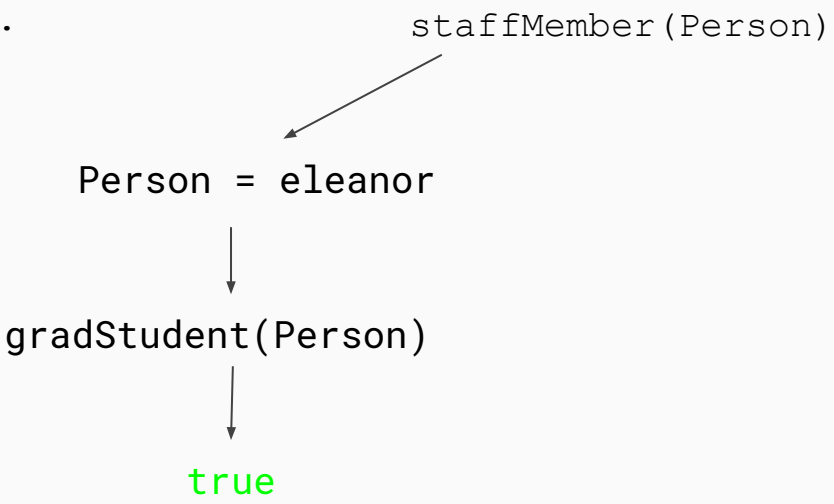
Resolution: `staffMember(Person), gradStudent(Person).`

```
staffMember(eleanor).
staffMember(mak).
staffMember(peter).
```



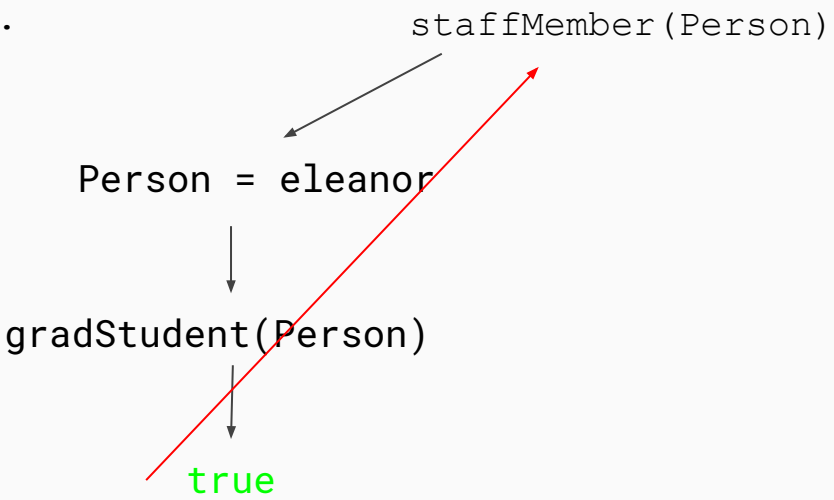
Resolution: `staffMember(Person), gradStudent(Person).`

```
staffMember(eleanor).
staffMember(mak).
staffMember(peter).
```



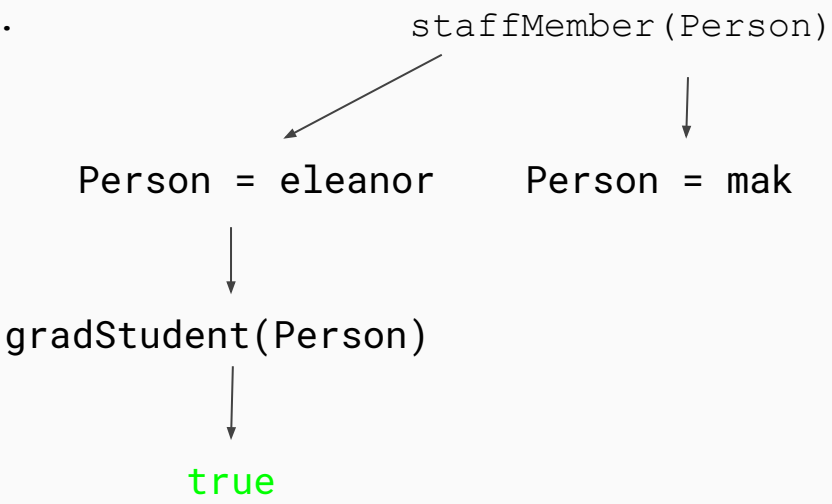
Resolution: `staffMember(Person), gradStudent(Person).`

```
staffMember(eleanor).
staffMember(mak).
staffMember(peter).
```



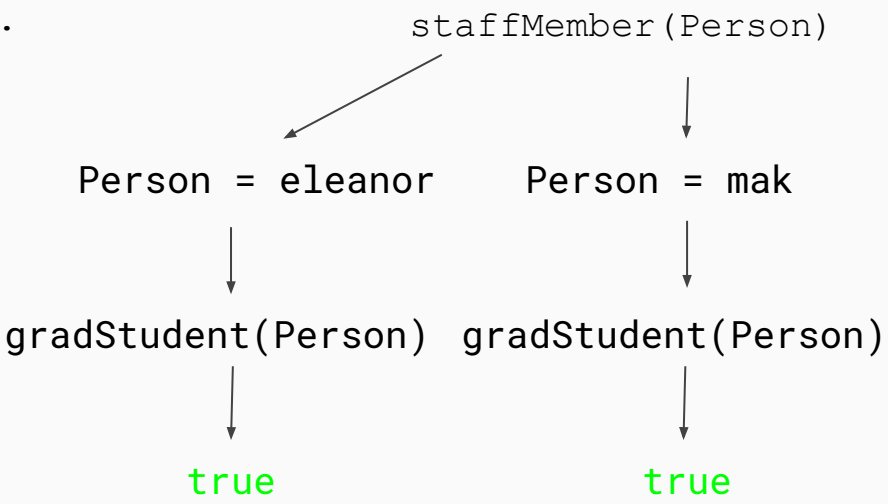
Resolution: `staffMember(Person), gradStudent(Person).`

```
staffMember(eleanor).
staffMember(mak).
staffMember(peter).
```



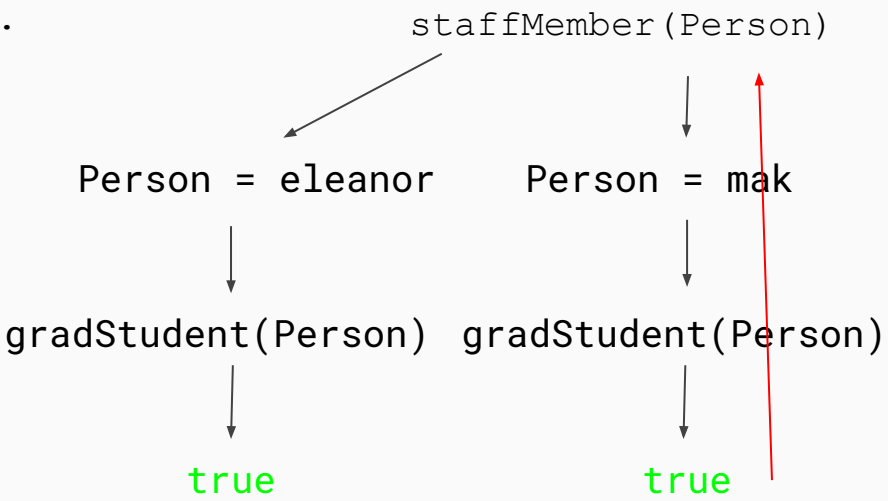
Resolution: `staffMember(Person), gradStudent(Person).`

```
staffMember(eleanor).
staffMember(mak).
staffMember(peter).
```



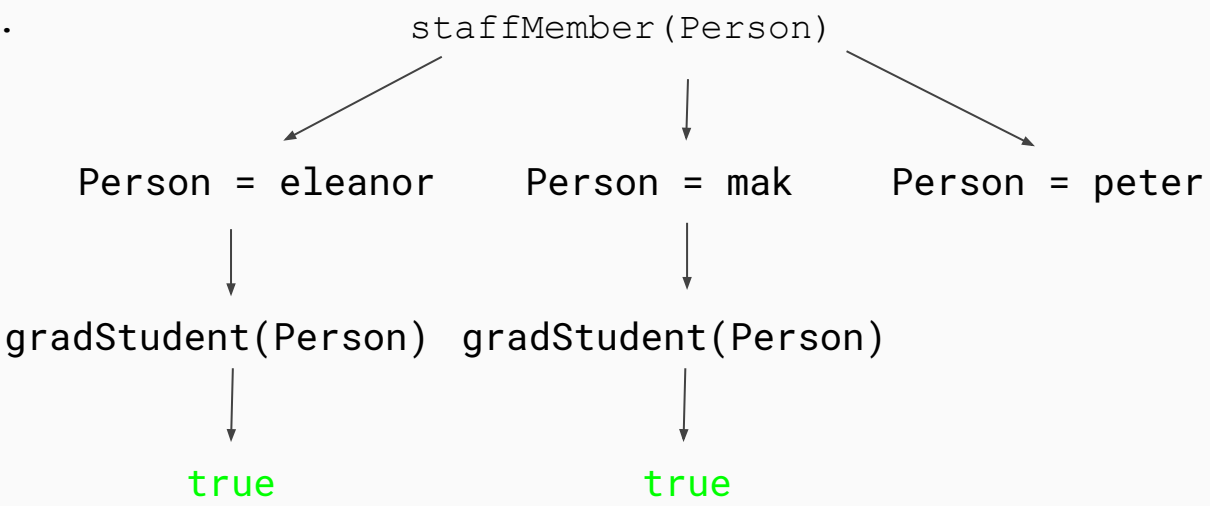
Resolution: `staffMember(Person), gradStudent(Person).`

```
staffMember(eleanor).
staffMember(mak).
staffMember(peter).
```



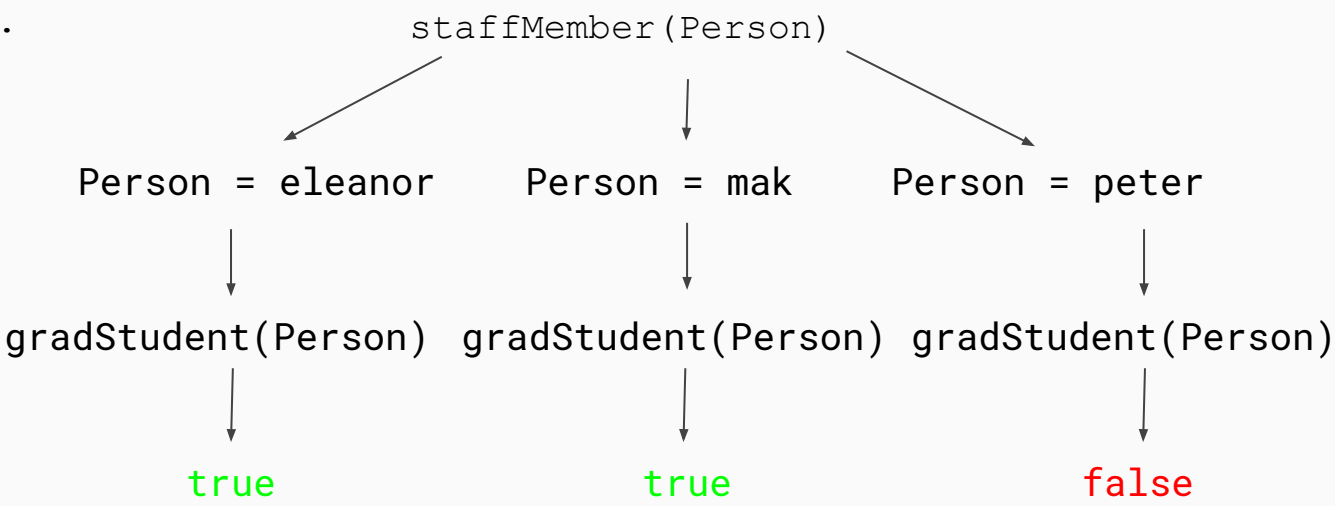
Resolution: `staffMember(Person), gradStudent(Person).`

```
staffMember(eleanor).
staffMember(mak).
staffMember(peter).
```



Resolution: `staffMember(Person), gradStudent(Person).`

```
staffMember(eleanor).
staffMember(mak).
staffMember(peter).
```



Continue with Grok Workshop 6 (Week 7).