

## ProLog

### Basic structures

- names
  - valid characters in names include most any on a keyboard; a name can even contain a space, if the entire name is enclosed in quotation marks.
  - names *beginning* with an upper-case letter or an underscore are reserved for variables
  - *atoms* (or constants) include words beginning with a lower-case letters used in isolation.
  - *predicates* (procedures) include words beginning with a lower-case letters and requiring (as indicated by parentheses) inputs.
- **Conditionals:** if  $p$  then  $q$  is coded as  $q : -p$ .
- **Conjunctions:**  $p \wedge q$  is coded as  $p, q$ .

Together, the code

$$r : -p, q$$

means  $p \wedge q \rightarrow r$ .

- **Disjunctions:**  $p \vee q$  is coded as  $p; q$ .
- **Negation:**  $\neg p$  is coded as  $\backslash +p$ , meaning the non-negated statement is not provable.

### Other important items:

- *Program* files, often referred to as “knowledge bases”, can be created separately using a text editor.
- Commands in prolog end with a period (.) character.
- Sample rules:

```
husband(luke, mara) :- wife(mara, luke).  
    expresses wife(mara, luke)  $\rightarrow$  husband(luke, mara)  
    modus ponens  
ownsLightSaber(X) :- jedi(X).  
    expresses  $\forall X(\text{jedi}(X) \rightarrow \text{ownsLightSaber}(X))$   
husband(X, Y) :- wife(Y, X).  
    expresses  $\forall X \forall Y(\text{wife}(Y, X) \rightarrow \text{husband}(X, Y))$ 
```

- Sample queries:

```
jedi(dooku).  
    evaluates as True since this is a fact in the knowledge base.  
master(sidious, X).  
    expresses “Is  $\exists X$  for which  $\text{master}(\text{sidious}, X)$  is true?” All instances are listed.
```

Practice

1. Write queries for
  - (a) whether luke is a child of leia.
  - (b) all children of leia.
  - (c) all sons of leia.
  - (d) all uncles of jacen.
  - (e) all grandchildren of anakin
  - (f) all names of "force-sensitive" characters (whether sith or jedi)
  - (g) all names of characters who are both sith and jedi
2. Write rules for
  - (a) mother( $X$ ), so that the mother of  $X$  is sought/found
  - (b) nephew( $X$ )
  - (c) isForceSensitive( $X$ )
  - (d) isForceSensitive( $X$ )
  - (e) grandfather( $X$ )
3. Add information to the knowledge base so that there is a person named owen who appears in response to the query `uncle(luke)`.