Definite Clause Grammars (DCGs) in Prolog

What's a DCG?

- Basically, a nicer way to write context-free grammars
- Difference lists:
  - Represent info as two lists instead of one
    - [input list], [output list]
  - If it can chomp through the input list without leaving anything behind (i.e. leaving the output list empty), then it's a valid sentence
- Queries:
  - Is "the unicorn licked the dragon" a valid sentence?

```
?- s([the,unicorn,licked,the,dragon],[]).
```

  - Generate all the valid sentences in the grammar

```
?- s(X,[]).
```

My example: music notation

DCG for validating measures in various time signatures

What’s it do?

- Given an input list of note types, it can check whether or not it would constitute a valid measure in a given time signature
- Currently supports whole notes, half notes and dotted half notes, quarter notes and dotted quarter notes, eighth notes and dotted eighth notes, sixteenth notes, and the corresponding rests

Examples

- Time signature: 4/4
- Notation:
  - `[sr,s,s,s,s,s,s,s,dq,er]`
- Queries:
  - `m4_4([sr,s,s,s,s,s,s,s,dq,er],[]).`
  - `m4_4(X,[]).`

Examples

- Time signature: 3/4
- Notation:
  - `[qr,q,q]`
- Queries:
  - `m3_4([qr,q,q],[]).`
  - `m3_4(X,[]).`