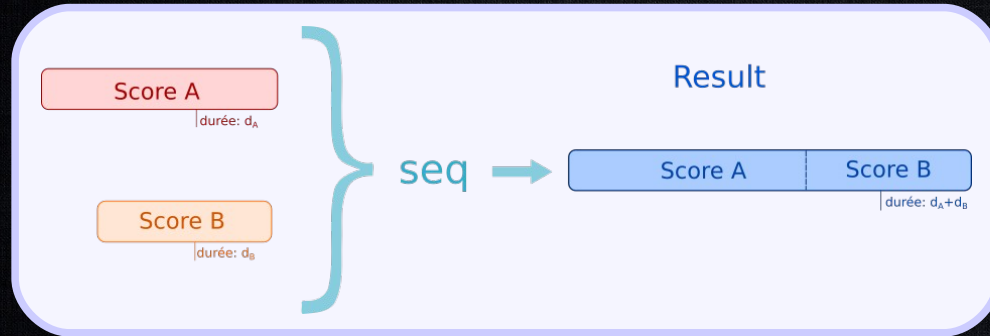# Guido Expressions

GuidoAR Operators

GuidoAR Operators

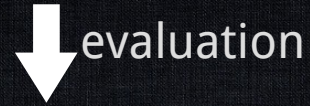- seq   *(sequence)*
- par   *(parallel)*
- head
- tail
- duration
- applyRythm

- top
- bottom
- transpose
- applyPitch

General Syntax

```
expr( operator arg arg )
```

↓ evaluation

Clean GMN String

General Syntax

expr(operator arg arg)

⬇

/ITL/scene/score set gmn expr(operator arg arg);

/ITL/scene/score set pianoroll expr(operator arg arg);

General Syntax

```
expr( operator arg arg )
```

- seq
- par
- head
- tail
- duration
- applyRythm

- top
- bottom
- transpose
- applyPitch

General Syntax

```
expr( operator arg arg )
```

- GMN String

```
/ITL/scene/score  set gmn expr(seq "[a]" "[b]") ;
```

⬇

```
/ITL/scene/score  set gmn "[ a b ]" ;
```

General Syntax

```
expr( operator arg arg )
```

- GMN String

- filepath

```
/ITL/scene/score set gmn expr(seq "[a]" score.gmn) ;
```

General Syntax

```
expr( operator arg arg )
```

- GMN String

- filepath

- existing object

```
/ITL/scene/sample  set gmn "[ {c,d,e} ]" ;
/ITL/scene/score  set gmn expr(seq "[a]" sample) ;
```

General Syntax

```
expr( operator arg arg )
```

- GMN String

- filepath

- existing object

- **Guido expression**

```
/ITL/scene/score set gmn expr(seq "[a]" (seq "[b]" "[c]") ) ;
```

General Syntax

`expr( operator arg arg )`

- GMN String

- filepath

- existing object

- Guido expression

New Data Event

```
/ITL/scene/score watch newData ;
```

- triggered when an object value change (set, write, update)

- postponed to the next event loop to handle recursion

Expression Re-Evaluation

```
...

/ITL/scene/score  set gmn expr(seq &scoreA &scoreB);

/ITL/scene/scoreA watch newData (
    /ITL/scene/score  expr gmn reexpr(seq scoreA scoreB)
    );
/ITL/scene/scoreB watch newData (
    /ITL/scene/score  expr gmn reexpr(seq scoreA scoreB)
    );
```
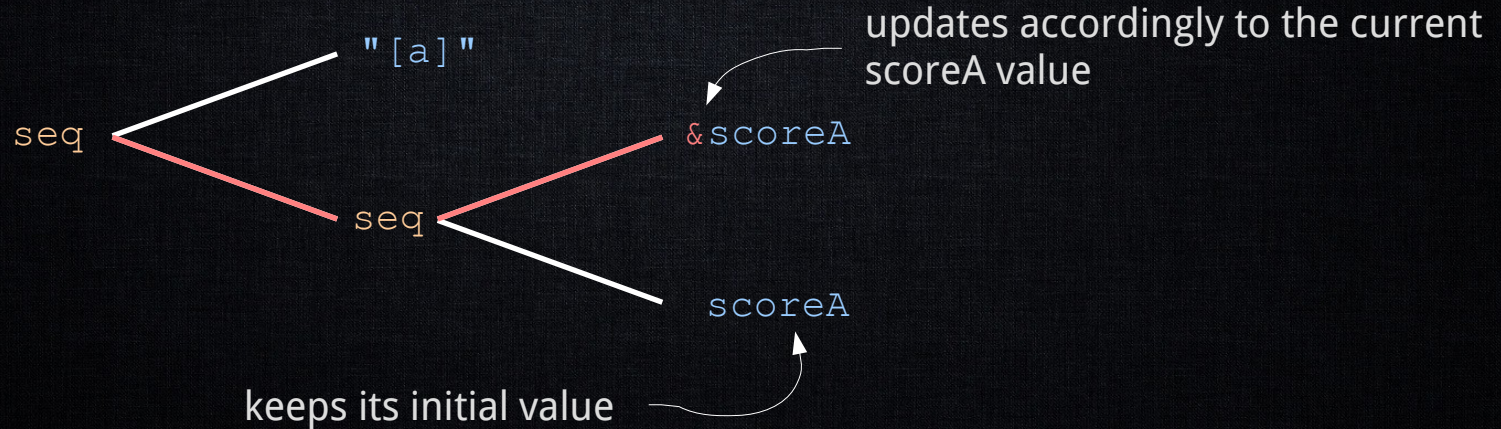
Expressions Evaluation

- reevaluation optimisation

- value from the initial state can be stored

updates accordingly to the current scoreA value

keeps its initial value

Expression composition

```
/ITL/scene/score  set gmn  expr(seq &sample &sample) ;
/ITL/scene/result  set gmn  expr(par ~score ~score) ;
```