

NAME

ydelta — calculate numeric differences for concurrent data

SYNOPSIS

ydelta [-s *regex*] -i '***interpretation*' [*inputfile*]

DESCRIPTION

The **ydelta** command outputs a single spine containing numeric data representing the numeric differences for concurrent tokens in specified input spines. For example, **ydelta** may be used to characterize the semitone distances separating pitches in a given vertical sonority or chord.

The user indicates which spines in the input stream are to be processed by specifying an interpretation via the -i option. The output interpretation is automatically assigned a new name by prepending the upper-case letter 'Y' to the given input interpretation. For example, a specified input of ****semit**s will result in an output interpretation named ****Ysemit**s.

OPTIONS

The **ydelta** command provides the following option:

-h displays a help screen summarizing the command syntax
-i *interp*** process input spines with interpretations ***interp*
-s *regex* skip over records matching *regex*; If all target spines contain the same token; token is output. If the tokens differ, the entire record is output (although all tabs are replaced by spaces).

Options are specified in the command line.

EXAMPLES

The operation of **ydelta** is best illustrated using an example. Consider the following input file called *praetorius*:

```
!! Praetorius, Es ist ein' Ros' entsprungen.
**semit    **semit    **text    **semit    **kern
!bass      !tenor     !text    !alto      !soprano
*M2/2      *M2/2      *M2/2    *M2/2      *M2/2
*F:        *F:        *        *F:        *F:
```

-7	0	Es	9	1cc
-7	0	ist	9	2cc
-7	-3	ein'	5	2cc
-2	-2	Ros'	5	2dd
-7	-3	ent-	5	2cc
-12	-5	sprun-	4	1cc
-10	-7	gen	2	1a
-14	-7	aus	2	1b-
-7	-3	ei-	0	2a
-8	0	ner	.	1g
-10	(-2	Wur-	2	.
.	-3)	.	.	1f
.	.	.	-3	.
-12	.	zel	0	.
.	-5	.	.	2e
-19	-3	zart,	0	1f
=	=	=	=	=
*_	*_	*_	*_	*_

Given the following command line:

```
ydelta -i '**semitis' praetorius > chords
```

the above input file would result in the following output:

```
!! Praetorius, Es ist ein' Ros' entsprungen.
**Ysemitis
!
*
*
[-7] 7 16
[-7] 7 16
[-7] 4 12
[-2] 0 7
[-7] 4 12
[-12] 7 16
[-10] 3 12
[-14] 7 16
[-7] 4 7
[-8] 8
[-10] 8 12
[-3]
[-3]
[-12] 12
```

```
[-5]
[-19] 16 19
.
*-
```

Only those input spines labelled `**semit`s have been processed; both `**text` and `**kern` have been ignored. The first value (given in square brackets) represents the lowest numerical value found in the processed spine(s) for the current data record. Successive numerical values indicate the differences between the lowest value and the remaining numeric values in the other spines. For example, the first input record contains the `**semit`s values -7, 0, and 9. The lowest of these values is -7 — which is placed in square brackets. The next lowest value (0) is 7 units above the lowest value, whereas the value 9 is 16 units above the lowest value. These other values are sorted and printed in ascending order in the output token. Notice that the output is typically in the form of Humdrum multiple stops. If only one numerical value is present in the processed input, a single value (in square brackets) will be output. If no numerical values are present, a null token (.) is output (see for example, the barlines).

PORTABILITY

DOS 2.0 and up, with the MKS Toolkit. OS/2 with the MKS Toolkit. UNIX systems supporting the *Korn* shell or *Bourne* shell command interpreters, and revised *awk* (1985).

SEE ALSO

hint (4), **regexp** (4), **regexp** (6), **xdelta** (4)