Set Operations in the Unix Shell

Created by Peteris Krumins (peter@catonmat.net). Peter’s blog: http://www.catonmat.net – good coders code, great reuse

Set Membership

```
grep -xc 'element' set
grep -xq 'element' set
awk '$0 == "element" { s=1; exit } END { exit !s }' set
awk -v e='element' '$0 == e { s=1; exit } END { exit !s }' set
```

Set Equality

```
diff -q <(sort set1) <(sort set2)
diff -q <(sort set1 | uniq) <(sort set2 | uniq)
awk '{ if (!($0 in a)) c++; a[$0] } END{ exit !(c==NR/2) }' set1 set2
awk '{ a[$0] } END{ exit !(length(a)==NR/2) }' set1 set2
```

Subset Test

```
comm -23 <(sort subset | uniq) <(sort set | uniq) | head -1
awk 'NR==FNR { a[$0]; next } if !($0 in a) exit 1}' set subset
```

<table>
<thead>
<tr>
<th>Set Union</th>
<th>Set Intersection</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat set1 set2</td>
<td>comm -12 &lt;(sort set1) &lt;(sort set2)</td>
</tr>
<tr>
<td>awk 1 set1 set2</td>
<td>grep -xF -f set1 set2</td>
</tr>
<tr>
<td>awk set1 set2 ... setn</td>
<td>sort set1 set2</td>
</tr>
<tr>
<td>sort set1 set2</td>
<td>join &lt;(sort -n A) &lt;(sort -n B)</td>
</tr>
<tr>
<td>sort -u set1 set2</td>
<td>awk 'NR==FNR { a[$0]; next } $0 in a' set1 set2</td>
</tr>
<tr>
<td>awk '!a[$0]++'</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set Complement</th>
<th>Set Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>comm -23 &lt;(sort set1) &lt;(sort set2)</td>
<td>wc -l set</td>
</tr>
<tr>
<td>grep -vxF -f set1 set2</td>
<td>wc -l &lt; set</td>
</tr>
<tr>
<td>sort set2 set1</td>
<td>awk 'END { print NR }' set</td>
</tr>
<tr>
<td>awk 'NR==FNR { a[$0]; next } !(0 in a)'</td>
<td></td>
</tr>
</tbody>
</table>

Set Symmetric Difference

```
comm -3 <(sort set1) <(sort set2) | sed 's/\t//g'
comm -3 <(sort set1) <(sort set2) | tr -d ' ' \t'
sort set1 set2 | uniq -u
awk 'NR==FNR { a[$0]; next } !(0 in a)' set1 set2
```

```
grep -vxF -f set1 set2 | grep -vxF -f set2 set1
grep -vxF -f set1 set2; grep -vxF -f set2 set1
awk 'NR==FNR { a[$0]; next } !(0 in a)' set1 set2
```

Power Set

```
$ p() { [ $# -eq 0 ] && echo || (shift; p "$@") | while read r ; do echo -e "$1 $r\n$r"; done }$ p `cat set`
```

Set Cartesian Product

```
while read a; do while read b; do echo "$a, $b"; done < set1; done < set2
awk 'NR==FNR { a[$0]; next } for (i in a) print i, $0 }' set1 set2
```

Disjoint Set Test

```
comm -12 <(sort set1) <(sort set2)
awk '++seen[$0] == 2 { exit 1 }' set1 set2
```

Empty Set Test

```
comm -12 <(sort set) | cut -d'=' -f1
awk '{ exit 1 }' set
```

Minimum

```
head -1 <(sort set)
awk 'NR == 1 { min = $0 } $0 < min { min = $0 } END { print min }'
```

Maximum

```
tail -1 <(sort set)
awk 'NR == 1 { max = $0 } $0 > max { max = $0 } END { print max }'
```